NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATION

BSc THESIS

# Motion Hollow: A WebVR experience for annotating movement qualities on dance content

**Lori S. Kougioumtzian**

**Supervisors:** **Dr Maria Roussou,** Assistant Professor
**Dr Katerina El Raheb**

ATHENS
OCTOBER 2020

**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

# Motion Hollow: Μια WebVR εμπειρία για επισημείωση κινητικών ποιοτήτων σε περιεχόμενο χορού

**Λόρι Σ. Κουγιουμτζιάν**

**Επιβλέπουσες:** **Δρ. Μαρία Ρούσσου,** Επίκουρη Καθηγήτρια
**Δρ. Κατερίνα Ελ Ράχεμπ**

**ΑΘΗΝΑ**
**ΟΚΤΩΒΡΗΣ 2020**

**BSc THESIS**

Motion Hollow: A WebVR experience for annotating movement qualities on dance content

**Lori S. Kougioumtzian**

**S.N.:** 1115201400076

**SUPERVISORS:** **Dr Maria Roussou,** Assistant Professor
**Dr Katerina El Raheb**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

# Motion Hollow: Μια WebVR εμπειρία για επισημείωση κινητικών ποιοτήτων σε περιεχόμενο χορού

**Λόρι Σ. Κουγιουμτζιάν**

**Α.Μ.:** 1115201400076

**ΕΠΙΒΛΕΠΟΥΣΕΣ:** **Δρ. Μαρία Ρούσσου,** Επίκουρη Καθηγήτρια
**Δρ. Κατερίνα Ελ Ράχεμπ**

# ABSTRACT

There are a lot of conceptual frameworks derived from the extensive research conducted in the field of movement analysis that are being used in the process of the identification and annotation of qualities in movement. However, these frameworks are not easily comprehensible by people with minimal or no scientific context related to the field, especially when used raw without being molded and expressed in a more user-friendly manner.

In the interest of tackling this problem, this thesis covers the process of developing a web-based crowd-sourcing experience where a set of simple questions are used as a theoretical framework for the annotation of movement qualities in dance moves performed by 3D animated models. This thesis also investigates ways of making the crowdsourcing experience more appealing, in order to encourage the full engagement of the users. Therefore, the MEAN Stack model was chosen along with A-Frame, a web framework for building 3D/AR/VR experiences, for the development of the application.

After developing a first prototype for the experience, a formative evaluation was carried out. 8 people participated in the evaluation suggesting improvements regarding the usability and the content of the experience. Moreover, the participants' feedback led to the formation of ideas regarding possible future extensions. In conclusion, setting all issues aside, all the participants found the experience very interesting and pleasant. They were able to complete the whole experience without losing interest, a fact which is not frequently the case when it comes to crowdsourcing activities and renders this thesis successful in its goal.

**SUBJECT AREA**: Web Applications

**KEYWORDS**: crowdsourcing, annotation on movement, human-computer interaction, webVR, 3D animation, Laban Movement Analysis

# ΠΕΡΙΛΗΨΗ

Η εκτενής έρευνα που πραγματοποιείται στον τομέα της ανάλυσης κίνησης έχει οδηγήσει στη δημιουργία πολλών θεωρητικών πλαισίων και κανονισμών για την αναγνώριση και επισημείωση κινητικών ποιοτήτων που παρατηρούνται στην ανθρώπινη κίνηση. Παρά το γεγονός αυτό, τα πλαίσια αυτά είναι δυσνόητα για ανθρώπους με ελάχιστο ή μηδενικό επιστημονικό υπόβαθρο σχετικό με τον τομέα. Για αυτό τον λόγο, η διαδικασία επισημείωσης κινητικών ποιοτήτων από απλούς χρήστες παρουσιάζει μέχρι στιγμής πολλές δυσκολίες.

Η πτυχιακή αυτή στοχεύει στην επίλυση αυτού του προβλήματος, αναπτύσσοντας μια διαδικτυακή εφαρμογή επισημείωσης των κινητικών ποιοτήτων που παρατηρούνται στην κίνηση τρισδιάστατων ανθρωπόμορφων μοντέλων που εκτελούν χορευτικές κινήσεις. Η εφαρμογή παραθέτει στον χρήστη ένα σύνολο προτάσεων που περιγράφουν με απλό τρόπο τις διάφορες ποιότητες και ζητά από τον χρήστη να επιλέγει ποιές θεωρεί ορθές για κάθε χορευτική κίνηση που συναντά. Η πτυχιακή αυτή ερευνά επίσης τους διάφορους τρόπους που μπορεί η εφαρμογή να γίνει πιο ελκυστική για τον χρήστη, έτσι ώστε να τον παρακινεί να ολοκληρώσει την εμπειρία. Γι'αυτό το λόγο, για την ανάπτυξη της εφαρμογής επιλέχθηκε ο συνδυασμός του MEAN Stack μοντέλου και του A-Frame, μιας τεχνολογίας που χρησιμοποιείται συχνά για την ανάπτυξη 3D/VR/AR διαδικτυακών εμπειριών.

Μετά την ολοκλήρωση της ανάπτυξης του πρωτοτύπου, πραγματοποιήθηκε διαμορφωτική αξιολόγηση με 8 χρήστες. Κατά τη διάρκεια της αξιολόγησης, προτάθηκαν εναλλακτικές για την βελτίωση της ευχρηστίας και του περιεχομένου της εμπειρίας, δίνοντας χώρο και στην ανάδειξη πιθανών μελλοντικών επεκτάσεων. Ωστόσο, παρά τα όποια ζητήματα, οι συμμετέχοντες δήλωσαν πως βρήκαν την εμπειρία ενδιαφέρουσα και ευχάριστη. Ολοκλήρωσαν την εμπειρία πληθοπορισμού δίχως να χάσουν το ενδιαφέρον τους, κάτι το οποίο δεν συμβαίνει συχνά σε παρόμοιες δραστηριότητες και καθιστά αυτή τη πτυχιακή επιτυχημένη στον στόχο της.

**ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ**: Διαδικτυακές Εφαρμογές

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ**: πληθοπορισμός, επισημείωση κίνησης, αλληλεπίδραση ανθρώπου-υπολογιστή, webVR, 3D animation, Laban Movement Analysis

# ACKNOWLEDGMENTS

I would first like to thank my supervisor, assistant professor Dr. Maria Roussou for her valuable assistance in the realization of this thesis and for helping me choose to pursue this subject, as she was the one who made me understand that working on a subject you love makes you strive to make it the best way possible. I would also like to thank her for all the helpful comments, reflections and suggestions regarding improvements or alterations, which she gave me during the evaluation process of the application.

I would also like to thank my advisor, Dr. Katerina El Raheb for providing her expertise in the movement analysis field, for introducing me to the field of movement qualities annotation, for all the help and support she offered me throughout the design and development of the application and for helping me realize that art can be expressed through technology, and when it does, beautiful things can be created.

All Motion Capture Animation data used for animating the 3D models were taken from the WhoLoDancE Motion Capture Library. WhoLoDancE (Whole-body Interaction Learning for Dance Education) is a Research and Innovation Action funded under the European Union’s Horizon 2020 programme (2016-2018) under Grant Agreement No 688865.

# TABLE OF CONTENTS

# LIST OF FIGURES

# PREFACE

This thesis was conducted as my Bachelor’s thesis at the Department of Informatics and Telecommunications. The design and development of the prototype took 6 months to complete and the writing part took one month. The illustrations used in the front-end of the application were all hand drawn on my digitizer.

Many thanks to my sister Maral for composing, performing and recording a mini piano lullaby for me to use as background music during gameplay in the application.

# 1. INTRODUCTION

## 1.1 Scope of the thesis

The main topic of this thesis is the design and implementation of a web-based crowdsourcing gamified experience which will be used to collect data regarding the way users (also referred to as players in this thesis) perceive the various qualities present in movement. There are a lot of conceptual frameworks derived from the extensive research [1][3][5][8] conducted in the field of movement analysis that are being used in the process of the identification and annotation of qualities in movement. However, these frameworks are not easily comprehensible by people with minimal or no scientific context related to the field, especially when used raw without being molded and expressed in a more user-friendly manner. In the interest of tackling this problem, this thesis covers the process of designing and developing a web-based crowdsourcing experience in the form of a game, where a set of simple questions are used for the annotation of movement qualities in dance moves performed by 3D animated models.

## 1.2 Contributions

This thesis is divided in two main parts, each with its respective purpose:

- The first part presents and examines different approaches of Laban Movement Analysis theory seen in Arts and digital games. The aim of this section is to give the reader an introduction to Laban Movement Analysis - the movement analysis theory used in this thesis - and to present the thought process according to which all analyzed data is factored into creating a set of simple and easily comprehensible questions based on the theory. The created set of clauses, along with a more elaborate conceptual framework, will be used in the annotation process of the game. Finally, this part also presents applications with a similar purpose to the experience we want to design (Related work).
- The second part of the thesis demonstrates the process of designing and developing a web-based crowdsourcing application using A-Frame [26], a web framework for building 3D/AR/VR experiences. The goal of this part is to walk the reader through all the stages that led to the developed experience, starting from the definition of the targeted user groups all the way to the final product.

## 1.3 Structure of the Thesis

The structure of the thesis involves two main sections:

- The first section (Chapter 2) provides an introduction to the theoretical background of the thesis which mainly consists of Laban Movement Analysis theory. Specifically, this section includes a presentation of LMA’s Eight Efforts using information gathered from various fields where the theory is used in practice, such as Dance, Animation, Theatre and Video Games. Since the application to be developed aims in the collection of data regarding the qualities present in movement, Laban’s Eight Efforts are then used to create a set of eight clauses that describe each Effort in a simple manner. This set of questions will be given to the user in the game during the annotation process. This section will also provide a presentation of previously developed applications or written papers that have investigated the problem of quality annotation (Related work).

- The second section (Chapter 3) provides the presentation of the design of the data-collection web-based experience. Specifically, it provides a section (Section 3.2) for explaining the storyline of the game and introducing the reader to the conceptual framework created to further examine beyond the main question of the thesis by taking into account factors and elements associated with the game's environment and how that could affect the user's perception during the annotating process. The next sections consist of the methodology and design process of the application's user interface. Specifically, the steps taken towards the design of the final product are outlined as follows: Definition of the targeted user groups (Section 3.3), Creation of possible personas (Section 3.4), Hierarchical Task Analysis and possible user scenarios (Section 3.5), Requirement Definition and Specifications Export (Section 3.6), Sitemap (Section 3.7), Low fidelity Wireframes (Section 3.8), Game Elements and Design (Section 3.9), Definition of the "Look and Feel" of the experience (Section 3.10).
- The third section goes through the steps of implementing the application (Chapter 4). It provides a section presenting the decisions made regarding the technologies used for the application development (Section 4.2), a section describing the animation process of the models and a section for the actual development of the game (Section 4.3 and Section 4.4 respectively). The final section provides an insight to the feedback received by the evaluation process of the prototype (Section 4.5) and some possible future extensions for the application (Section 4.6). Last but not least, the final section presents the conclusions derived from the evaluation process (Section 4.7).

# 2. BACKGROUND AND RELATED WORK

## 2.1 Introduction

This part of the thesis contains an introduction to Laban's Movement Analysis theory and a presentation of the various ways this theory is approached in different art fields. The fields examined are the following:

- Laban Movement Analysis in the Animation industry
- Laban Movement Analysis in Theatre
- Laban Movement Analysis in the Digital Games industry

The analysis is followed by the presentation of the conceptual framework created specifically for the game developed for this thesis, which is used during the annotation process of the game.

## 2.2 Background framework

### 2.2.1 Laban Movement Analysis

Laban Movement Analysis (LMA) is an expert-based method used to observe and analyze movement. It was developed by Rudolf Laban, a Hungarian dance artist and theorist. Laban, along with his collaborators, was able to identify and compartmentalize all movement patterns and thus create a rich and complex movement language, the LMA. LMA is widely used in various fields of research, particularly in fields studying movement expressivity and computation [7][8]. Human Computer Interaction (HCI) is one of these fields since research on whole body movement analysis is considered to be one of HCI's research subjects.

Laban originally categorized his studies in two basic categories: "Eukinetics", which is the name for his Effort studies, and "Choreutics" which is basically his Spatial Harmony theory. However, one of Laban's students, Irmgard Bartenieff, elaborated these categories in four, Body, Effort, Shape and Space. Thus, a new system called BESS was developed.

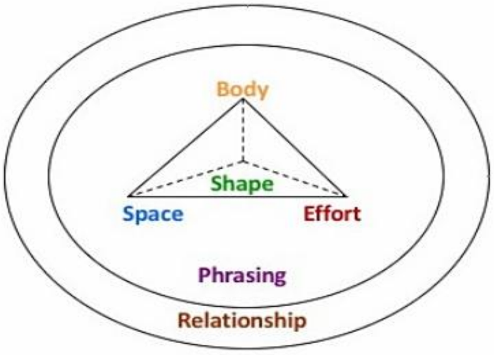


**Figure 1: LMA and BF BESS System**

In the BESS system, every movement can be described related to the categories presented in Figure 1, through experience, sensing and observation. The “Body” category focuses on using one’s body as a tool, observing each part of which can make it possible for the observer to describe movement. In the “Effort” category, Laban uses the motion factors of “Weight”, “Time”, “Space” and “Flow” to describe movement sensation. Each factor has opposite polarities that reveal the subtleties of movement [27]. The four motion factors are presented below [22]:

- **Weight**
  This factor has two elements, Heavy (or Strong) and Light. The Heavy element can be used to describe movement that is resistant to the moving object’s or body’s weight. On the other hand, The Light element can be used to express the lack of resistance to weight, a more “relaxed” attitude towards it.
- **Space (or Direction)**
  This factor has two elements, Direct and Indirect (or Flexible). Movement that happens along a straight line or shows resistance to the usage of their surrounding space, or kinesphere, can be characterized as Direct. The Indirect element can be used to describe a multi-directional movement.
- **Time**
  This factor has two elements, Sustained and Sudden (or Quick). The Sustained element is used to tag a movement with slow speed, while the Sudden element is used for quicker movements.
- **Flow**
  This factor has two elements as well, Bound and Free. Movement can be considered Free if it is entirely unrestricted and hard to stop all at once. On the other hand, the Bound element can be used to describe controlled and obstructed movement.

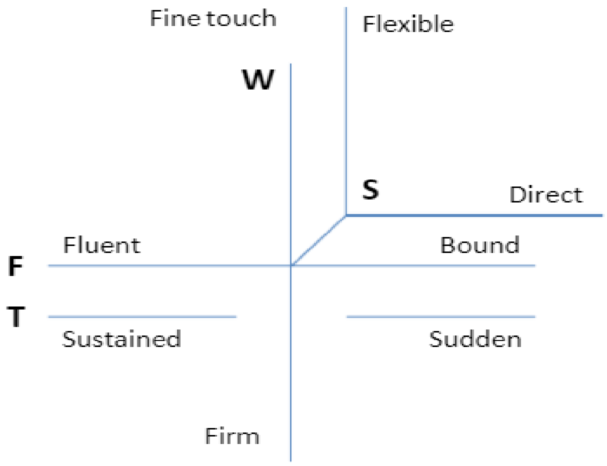


**Figure 2: Laban Effort Graph for describing quality of effort**

Combining these four factors we can generate multiple expressions describing the dynamic qualities of movement. These expressions are what Laban called Efforts (Figure 2).

The next category of the BESS System is “Space”, which mainly consists of Laban’s Space Harmony Theory, which focuses on analyzing the relationship between people and their environmental surroundings. Last but not least, the BESS System has a category called “Shape”, which explores how the human body changes in relation to itself and the space it’s moving in [24].

In the next chapters, this thesis will be mainly focusing on the Effort category.

### 2.2.2 Laban Movement Analysis – The 8 Basic Efforts

The human body has the capacity to produce countless movements according to one’s personality, characteristics and mannerisms. However, Laban decided to pursue a different approach. After studying the movement of the human body extensively, he was able to reduce the set of countless movements down to eight basic versatile efforts that all people do every day to some extent. “Effort” is defined as the inner attitude towards a motion factor [27], and is applied through the eight basic effort actions presented below:

- **Punch**
- **Slash**
- **Dab**
- **Flick**
- **Press**
- **Wring**
- **Glide**
- **Float**

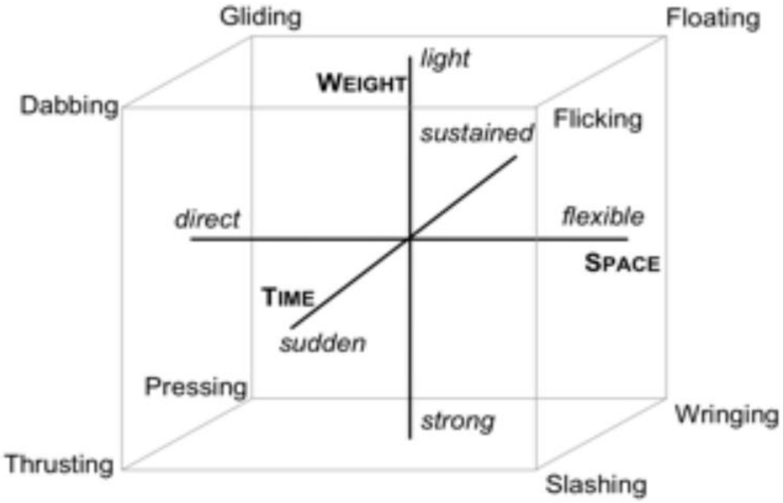
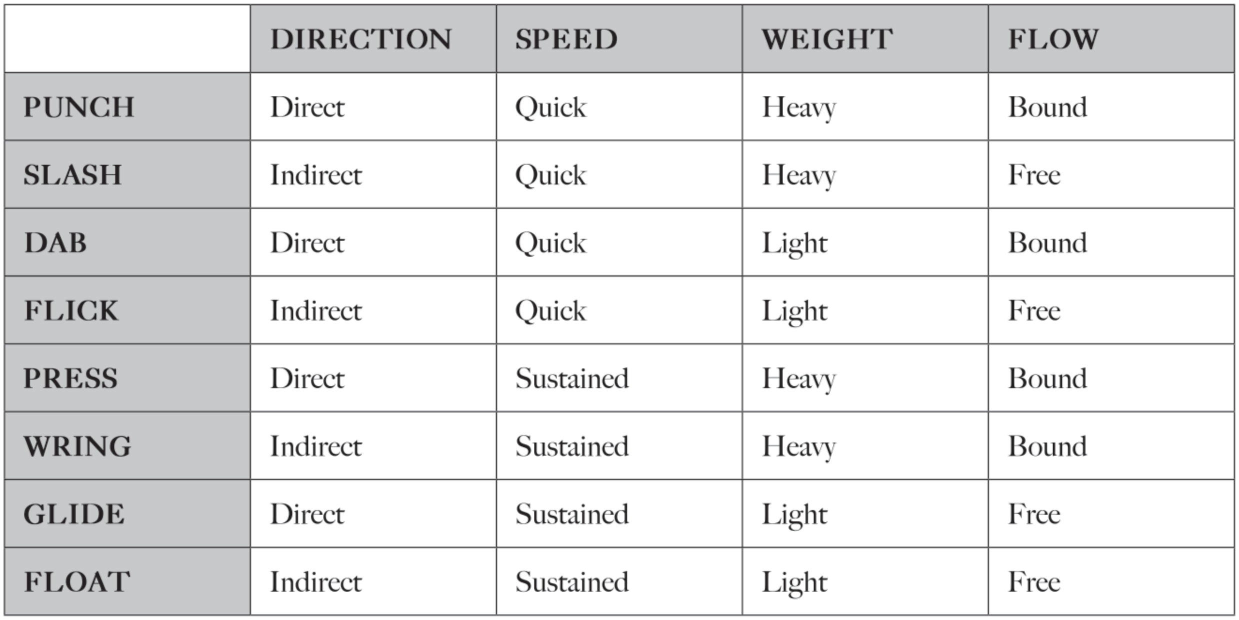


**Figure 3: Rudolf Laban’s dynamosphere describing his notion of basic motion efforts**

Figure 3 shows how each of these Efforts is expressed with a unique combination of the four motion factors - Weight, Flow, Space (or Direction) and Time (or Speed). The Eight Basic Efforts and their motion factor combinations are presented in Figure 4.



| | DIRECTION | SPEED | WEIGHT | FLOW |
|---|---|---|---|---|
| PUNCH | Direct | Quick | Heavy | Bound |
| SLASH | Indirect | Quick | Heavy | Free |
| DAB | Direct | Quick | Light | Bound |
| FLICK | Indirect | Quick | Light | Free |
| PRESS | Direct | Sustained | Heavy | Bound |
| WRING | Indirect | Sustained | Heavy | Bound |
| GLIDE | Direct | Sustained | Light | Free |
| FLOAT | Indirect | Sustained | Light | Free |

**Figure 4: Laban's Eight Efforts**

The table in Figure 4 presents Laban’s Eight basic Efforts, as referenced from a Drama Educational source [10]. Since it is used by actors, who are by definition living things, the “Sudden” keyword of the Speed motion factor has been replaced by the “Quick” keyword which makes more sense for human movement. Moreover, we can see that the Space factor is named Direction and the “Flexible” keyword has been replaced with the “Indirect” keyword, since people usually move following a direction towards a destination in the real world.

However scientifically accurate and elaborate the table above might be, it is very hard for people of little or no scientific knowledge on the field to understand what this mappings actually mean. We will try to explain each of the eight efforts using examples of movement from our everyday lives. The following descriptions use only three out of the four factors: Time, Space and Weight.

**Punching** is a violent and direct move, the most common form of which is the clenched fist thrusting at a target during a fight or a stand-off. There is no time-related indulgence in this action, it happens very quickly and suddenly. Punching does not surrender to space flexibility, its path stays direct from the beginning to the end.

**Slashing** is a movement commonly seen in combat or sports that require the use of some sort of racquet or handheld equipment, like tennis. The movement can be associated with bringing a sword quickly down-front to cut a targeted entity in half. Another example of slashing would be the movement of a tennis racquet while the player is serving the ball. Slashing is a heavy effort, happens quickly, it is flexible in terms of space usage.

**Dabbing** is a light movement that is very common in everyday activities. People dab computer keyboards while typing, painters dab a canvas with a brush while painting. Dabbing is direct, not moving flexibly around space. It is a quick and light movement.

**Flicking** is a movement seen quite frequently. People might flick fluff off of their clothes, a swarm of bees away from their body, or hair out of their sight. Flicking is quick and brief as a movement, it is quite flexible in terms of space usage usually with no initial specific direction and it is light.

**Pressing** is an action performed by people on a daily basis. We press door-bells, buttons on any sort of equipment, the keys of a piano in order to play a tune or a stamp on an envelope to be mailed. Pressing as an effort is direct, time-resistant and heavy, meaning that the action of pressing requires a certain amount of weight to be forced.

**Wringing** makes us think of a wet towel or wash cloth that has to be wrung out until all the liquid is squeezed out of it. It is closely associated with the action of twisting. As an effort, wringing is time-resistant, but flexible when it comes to space and direction. It is a heavy action, meaning that it requires a certain amount of weight for it to be initiated.

**Gliding** is commonly seen in activities such as ice-skating or roller-blading, or maybe even coming down a snow-covered mountain on a sleigh. As an effort, wringing is a time-resistant action, meaning that it doesn’t happen suddenly. It is also light, since it does not require too much weight in order to be initiated. In terms of space, wringing is direct.

**Floating** can be associated with activities such as flying or swimming. As a basic effort, floating is time-resistant. It is indirect and flexible in terms of space and direction, and it is light in terms of weight, which is only fair since floating suggest buoyancy and weightlessness [22].

However, since we want to design an experience that incorporates many game elements, it would be very helpful to consult a source which explains the Efforts under the scope of the digital gaming world. In the gaming world there are no limits to the way characters and avatars might move, considering that the natural laws of physics could be easily omitted in an imaginary cosmos. Characters can be forced to move in a certain direction because “they are cursed”, or “there’s a spell cast on them”. They can be able to roll through the room as if there is nothing stopping them “because they are wearing a magic flying cape” or “they drank the ghost potion and now they can pass through everything”. When it comes to movement in games, there are no limitations.

According to gaming blogs [11], Laban’s Eight Basic Efforts are frequently used in role-playing video games as basic actions performed by the players’ characters or they can be used as descriptions for the characters’ personalities themselves. The definitions are explained below:

**Press –** Presser characters are great people – An emperor, a general, calm, intense, ambitious person who is unafraid of grinding anything in their path to dust to get what they want.

**Thrust -** Thruster characters are intense people. When they are the “good guys” of the story, they’re often proud and capable and exceedingly restrained.

**Slash -** Slasher characters are arrogant, audacious and notorious characters.

**Wring -** Wringer characters are often scheming villains, twisted and evil, looking out only for themselves.

**Glide -** Gliders are often socially adept, dangerous characters who get you to do things you didn’t intend to do and yet somehow have you respecting them for it.

**Float -** Floaters are kind friends, fools skipping off cliffs that can easily become collateral damage, and sometimes the annoying partner the player can’t restrict.

**Flick -** Flickers are often annoying, unpredictable creatures who must be convinced or connived into commitment. If they are loyal, they make inconsequential allies.

**Dab -** Dabbers are devastating social creatures. They’re political powerhouses, and ruthless manipulators.

These basic eight efforts, along with Laban’s Space Harmony Theory presented in the following section, will be used in Chapter 3 to create a conceptual framework for the annotation process of the developed experience.

### 2.2.3 Laban Movement Analysis – Space Harmony Theory

In the late 1920s, the term “choreutics” was introduced through Rudolf Laban’s publication of “Choreographie”. Choreutics, or space theory, deals with the principle of form which is built on directional elements. Later on, the term “choreutics” was replaced by the term “space harmony”, a term generally associated with the relationship between spatial structure and the dynamic content of movement [28]. To understand the space theory better, we are also going to define some related terminology.

Laban defined as Kinesphere “the sphere around the body whose periphery can be reached by easily extended limbs without stepping away from that place which is the point of support when standing on one foot” [29]. In other words, Kinisphere is the space surrounding our bodies which is within the reach of the limbs without having to modify our initial center. When doing big movements using the limbs, somebody can reach a wider area around them, called the Far Reach Kinesphere. If somebody moves within near reach of themselves, they are moving in the Near Reach Kinesphere. The zone in-between is the Mid Reach Kinesphere. Kinesphere is the area explored before moving to the general space, which is the actual room we are in.

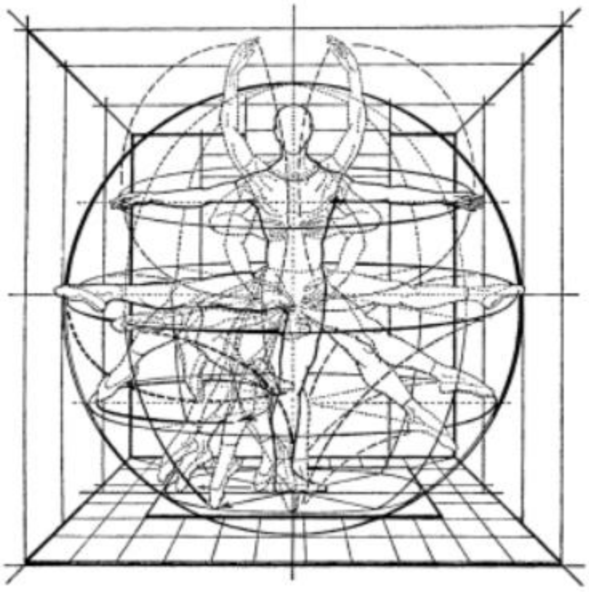


**Figure 5: Kinesphere and Reach Areas**

According to Laban, there are three types of movers, whether they are dancers, actors, martial-artists or they are involved with other activities that require movement. The first type consists of the movers who prefer moving in the high level, such as leaping, jumping or springing off the ground as much as possible. Next, there is the type of movers who enjoy moving in the central level, moving in a more sensuous and intuitive manner. Last but not least, there is the type of movers who like to move in the deep level, meaning that they prefer more earth-bound movements that use lower-body strength [22].

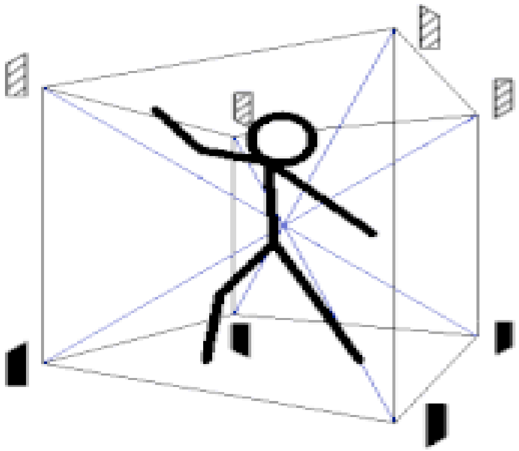


**Figure 6: Laban's moving levels**

## 2.3 Related work

The following sections consist of the related work, papers and applications that were taken to consideration in the application's design process. Multiple annotation experiences were examined [17][30][31][32], two of which are presented in detail below, one with a synchronous multi-player approach and one with an asynchronous single-player approach. The applications presented are the following:

- TagATune: A Game for Music and Sound Annotation
- MajorMiner: A Web-Based Game for Collecting Music Metadata

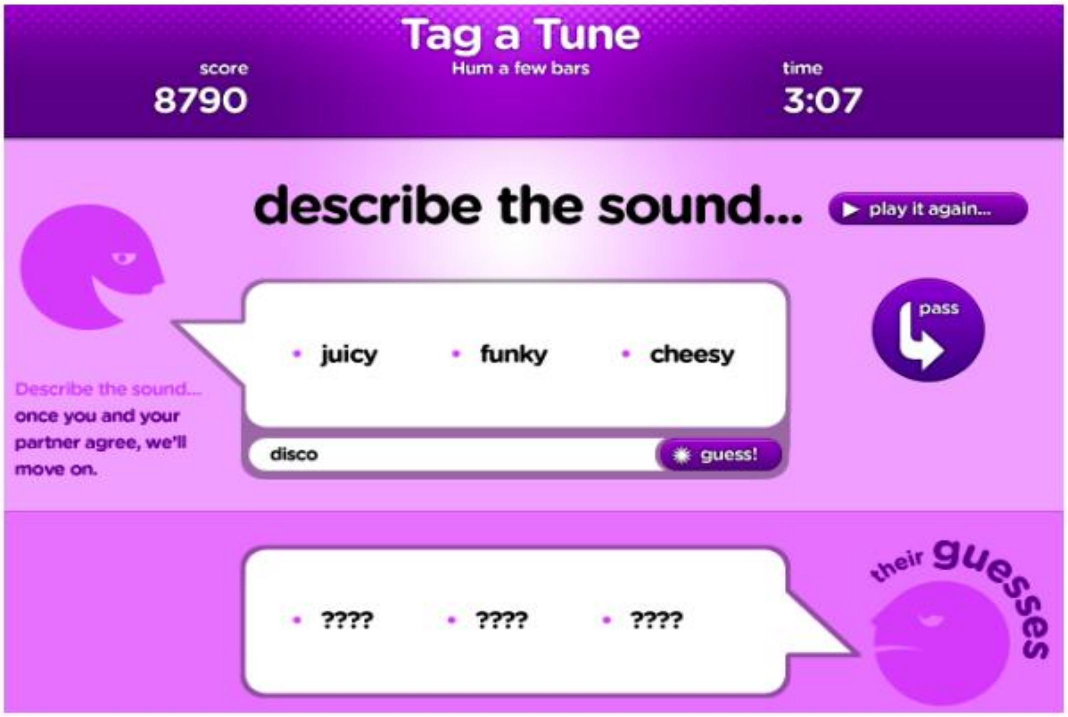
### 2.3.1 TagATune: A Game for Music and Sound Annotation

TagATune [17] is an audio-based online game for annotations of audio files. The game's goal is to extract meaningful descriptions of sounds and music from users, without having to deal with the high cost of obtaining annotation data manually. Since TagATune's goal is very similar to the goal of this thesis, it was worth examining and incorporating it into this section.

In crowdsourcing applications such as TagATune, the assignment of descriptive keywords to media files by users is mainly subjective, since the characterizations are based on the player's opinion on the stimuli they are being exposed to. TagATune is able to produce a set of labels by taking the ones that have been confirmed and validated by the most players, thus making the process of the annotation low-cost or even free. The players would have to answer questions such as whether a tune feels cold or warm, scary or pleasant, bright or dull, happy or sad.

In regards to the application's proposed gameplay, players are immersed in an audio environment that consists of not only music, but a variety of sound clips too. The sound clips are provided by FreeSound.org. The players are paired with partners randomly and anonymously from a pool of available players instead of playing against a database or against multiple players concurrently, both of which are approaches followed by other similar applications. The players paired with each other have to collaborate to tag a given tune with a particular label.

Specifically, they have to guess what their partner is thinking and are given three minutes in total to come up with agreed descriptions for as many sounds as possible. In each round, a sound is randomly selected from the database and presented to the partners. A description becomes an official tag that can be used for searching sounds that have been associated with it when it is agreed upon by a certain number of people. That number will depend on game statistics.



Describe the sound... once you and your partner agree, we'll move on.

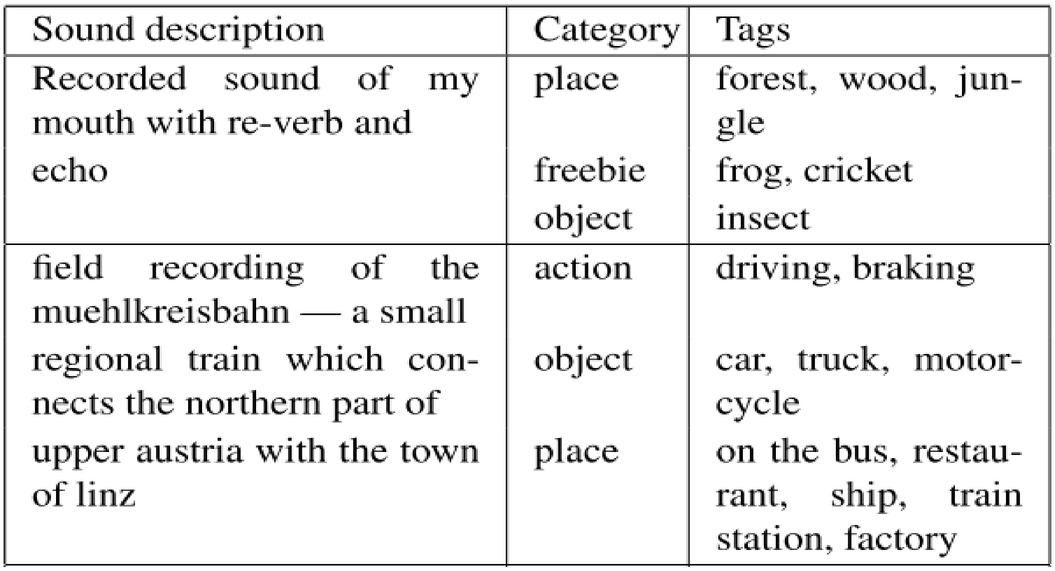
**Figure 7: TagATune User Interface**

Figure 7 is a screenshot of the game's annotation round, in which the two players are asked to choose the category word for the tune. Each word is attached with a score. The harder it is to describe the given tune, the higher the score associated with it. Afterwards, the comparison round presents a sound which the players have to compare to other sounds that fall into the same category, using one of the following types of questions:

1. Preference questions: *"which of the two tunes do you prefer?"*
2. Similarity questions: *"which of the two tunes is more similar to this third tune?"*
3. Perception questions: *"which of the two tunes is more X"*, X being a description provided by previous players

All of these are questions that require the player to make a decision in order to tag the tune they are currently playing for. So regardless of the type of question the user is required to answer during gameplay, we could categorize TagATune's tagging technique as decision making.

The table in Figure 8 presents some tags that were collected by the prototype. The second column states the author's chosen descriptions and the last column presents the descriptions produced by players.



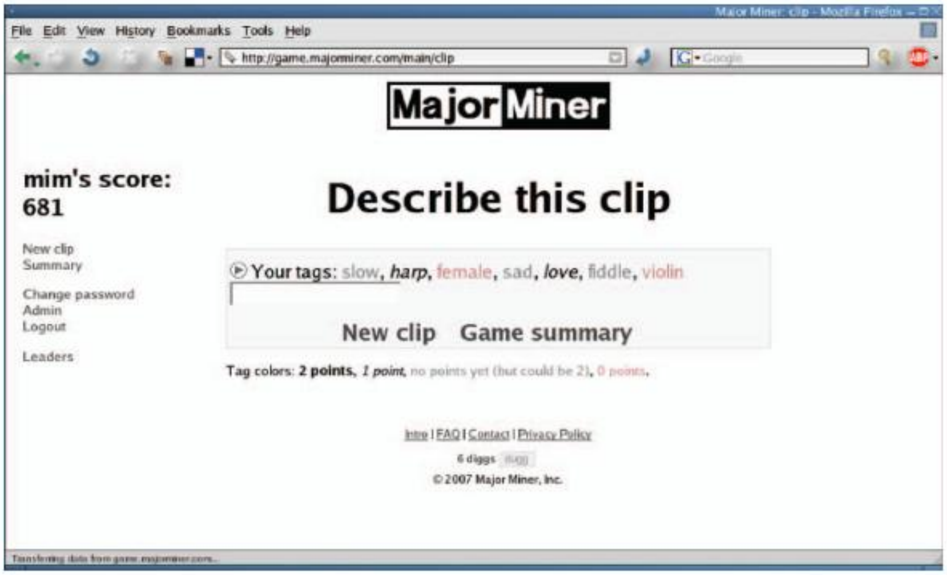
| Sound description | Category | Tags |
|---|---|---|
| Recorded sound of my mouth with re-verb and echo | place | forest, wood, jungle |
| | freebie | frog, cricket |
| | object | insect |
| field recording of the muehlkreisbahn — a small regional train which connects the northern part of upper austria with the town of linz | action | driving, braking |
| | object | car, truck, motorcycle |
| | place | on the bus, restaurant, ship, train station, factory |

**Figure 8: Descriptions of four different sounds: author versus players**

### 2.3.2 MajorMiner: A Web-Based Game for Collecting Music Metadata

MajorMiner [30] is a web-based game that aims to make the process of collecting descriptions of musical excerpts user-friendly, fun and objective. Participants are given sound clips with duration of 10 seconds each, and they are asked to tag the clips with certain descriptions. They score points when their descriptions match the descriptions other players have given for the same tune before. Unlike TagAtune or other annotation experiences [17][30][31][32], MajorMiner is designed with a single-player mode approach, since the user's chosen tags and scoring are registered to an offline database.

When it comes to gameplay, MajorMiner is simple. In the beginning the player asks for a sound clip. The clip could either be chosen among sound clips that other users have tagged before, or be completely new. The player listens to the clip and types in the tags they feel describe the clip. If any of the descriptions they used have been previously used by one player only, the current player scores one point and the players who originally used the tags score 2 points each respectively. If the tags have already been used by two or more users, the current player gets zero points. Description tags used for the first time don't add to the score, but they can potentially score the player two points if used by other players in the future. Figure 9 presents a screenshot of the application's gameplay.


Major Miner: clip - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://game.majorminer.com/main/clip
Google
MajorMiner
New clip
Summary
Change password
Admin
Logout
Leaders
New clip
Game summary
Intro | FAQ | Contact | Privacy Policy
6 diggs digg
Transferring data from game.majorminer.com...

mim's score:
681

Describe this clip

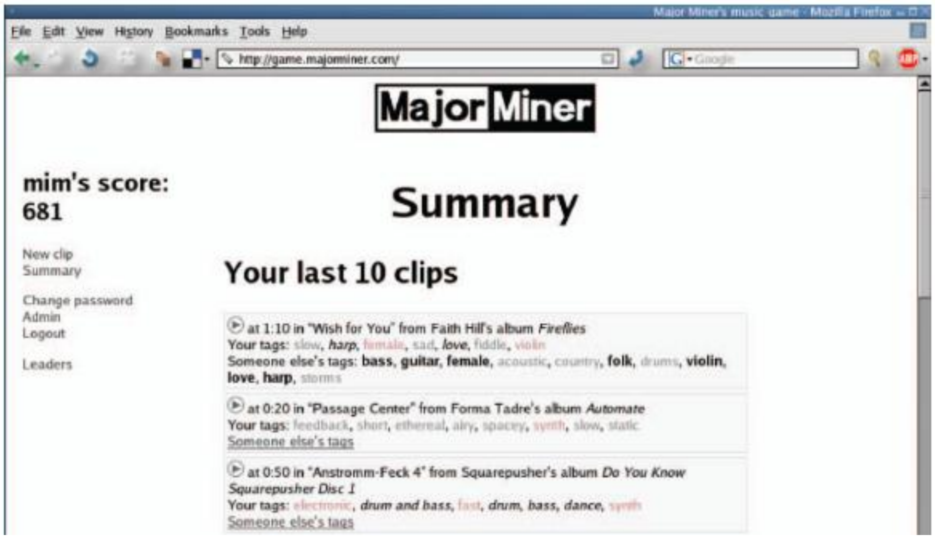
Your tags: slow, *harp*, female, sad, *love*, fiddle, violin

Tag colors: **2 points**, *1 point*, no points yet (but could be 2), 0 points.

© 2007 Major Miner, Inc.

**Figure 9: MajorMiner basic gameplay**

After the player has finished tagging the clips, they can go to their game summary, which shows clips the player has recently interacted with. It also lists the artist, album, and track names of the clips and gives the player the opportunity to see one other player's tags for the same clips. The next time the player logs in, the system informs them whether any of their descriptive tags have been used by other players while they were logged out and thus updates the player's score accordingly. Figure 10 presents a screenshot of the summary page.


Major Miner's music game - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://game.majorminer.com/
Google
MajorMiner
New clip
Summary
Change password
Admin
Logout
Leaders

mim's score:
681

Summary

Your last 10 clips

at 1:10 in "Wish for You" from Faith Hill's album *Fireflies*
Your tags: slow, *harp*, female, sad, *love*, fiddle, violin
Someone else's tags: **bass**, **guitar**, **female**, acoustic, country, **folk**, drums, **violin**, **love**, **harp**, storms

at 0:20 in "Passage Center" from Forma Tadre's album *Automate*
Your tags: feedback, short, ethereal, airy, spacey, synth, slow, static
Someone else's tags

at 0:50 in "Anstromm-Feck 4" from Squarepusher's album *Do You Know Squarepusher Disc 1*
Your tags: electronic, *drum and bass*, fast, *drum*, *bass*, *dance*, synth
Someone else's tags

**Figure 10: MajorMiner player's game Summary**

MajorMiner's rules and scoring were designed to encourage the players to tag the clips as carefully as possible. The duration of 10 seconds was chosen to make the process of the annotation and comparison more objective. The data gathered by the application is used to measure the degree to which binary classifiers could be trained to predict the most popular description tags.

### 2.3.3 Conclusions

In general, no matter the different approaches followed, both TagATune and MajorMiner applications have interesting points. TagATune supports synchronous gameplay which requires players working in pairs. On the other hand, MajorMiner is designed for asynchronous gameplay but provides the players with access to other people's tagged descriptions for the clips that they have both interacted with. While MajorMiner's task is to agree on tags, TagATune's task is to decide whether or not the currently playing tune is the same for both of the collaborating players. MajorMiner requires the player to enter the descriptive tags with free-text input, while TagATune uses free-text input along with questions that prompt the user to make decisions based on their perception. TagATune has a more "playful" user interface, while MajorMiner has a simpler, more minimalistic design.

Motion Hollow might not be an experience for annotations on sound clips, but many of the elements found in the aforementioned applications could be easily applied to an experience regarding annotations on dance content. So, in the process of designing the application, we considered using elements found in either of these applications. The Motion Hollow prototype was designed to initially support only a single-player mode, as does MajorMiner. In Motion Hollow the player's task is to tag dance content by making a decision based on a standard set of descriptive clauses, a layout that is quite similar to the TagATune's layout. Moreover, we decided to include several game elements to the application's user interface to make the experience more engaging.

These are just a few of the decisions made regarding the design of our application's prototype. The complete design and development process is presented in the following chapters.

# 3. DESIGNING A PROTOTYPE

## 3.1 Introduction

This chapter presents the methodology followed in order to design the application's prototype. The design process consists of the steps outlined below:

- Definition of the storyline and the conceptual framework
- Definition of the target user groups
- Definition of possible personas
- Possible user scenarios and Hierarchical Task Analysis
- Requirements definition and Specifications export
- Sitemap
- Low fidelity wireframes
- Game elements and design
- General “look and feel”

## 3.2 Storyline and Conceptual Framework

### 3.2.1 The storyline of the experience

The story takes place in a small mystical town called Motion Hollow. The town consists of four districts in total, each corresponding to one of the basic elements of nature: Water, Earth, Fire and Wind. Motion Hollow is ruled by the Empresses of Nature, each of whom reigns over one of the districts. Every empress possesses the power of the natural element associated with the district they rule over. The empresses have a sister called Celestine. She is the Muse of Light, and every morning she dances around the districts to light up the whole town. Just like white light is a combination of all colors in the color spectrum, Celestine's power derives from the combination of all the natural elements, a fact which justifies her role as the light barrier.

The experience begins with Motion Hollow immersed in darkness. Celestine has gone missing and so has the light. Her sisters, the Empresses, are extremely worried and ask for the player's help to find Celestine. Being completely panicked, they scatter around the town; some of them maybe even leave their districts and travel to others to search.

The player listens to the narration of the story and then is given a map of Motion Hollow with links to all the districts. The player's mission is to visit all the districts, and meet the empresses. At every district the empress who is present performs three different dance moves, which are some of Celestine's favorite moves, so as to help the player identify her movement if they see her. The player is provided with eight description clauses at every dance move, one for every one of Laban's 8 basic Efforts, and is required to mark true the ones they believe match the movement qualities in the move performed.

After completing the annotation process, the player is asked to answer whether or not they think that the empress they met with is the ruler of the particular district or not. This question investigates the secondary research topic of this thesis, which is whether the environmental surroundings affect the way people interpret movement qualities.

The player's mission is complete when they have travelled to every district and have answered all the questions required.

### 3.2.2 The Conceptual Framework

After deciding the storyline of the experience, it is time to define the conceptual framework which will be used during the movement quality annotation process. As mentioned in the chapters above, this thesis has two research goals:

- To gather data from users regarding the movement qualities present in dance content.
- To gather data from users regarding the degree to which the environmental surroundings affect the way people perceive the movement qualities in dance content.

This section comprises two parts, one for covering each of the aforementioned goals, using information from the theoretical background analyzed in Chapter 2.

For the movement qualities annotation process, the proposed conceptual framework is based on Laban's Eight Basic Efforts. The framework consists of one descriptive clause derived from each Effort, making it a set of eight descriptions in total. So consulting the table presented in Figure 4, the following set of clauses is created:

1. She moves her arms as if she is **punching**, aiming invisible objects around her.
2. She moves her arms as if she is **slashing** obstacles around her with an invisible sword.
3. She moves around like a **gliding** fairy.
4. She stands still like a **floating** fairy, as if there's nothing weighing her down.
5. It's as if she is **pressing** her body or limbs against invisible walls blocking her way.
6. She moves as if she wants to **dab** a locked target and knock them out in silence.
7. She moves like she is **flicking** stuff away from her body.
8. She moves her body twisting and turning as if she is a wet washcloth being **wrung**.

Each district incorporates the performance of three dance moves, each of which prompts the user with the set of descriptions presented above. The user is required to mark as true the descriptions that match the movement qualities they can identify in the current dance move.

However, we are not only interested in collected data about the movement qualities. We are also interested in understanding the degree to which the environmental surroundings affect the user's judgment in the process of the annotation. Initially, instead of creating four different environments, we thought we should choose four different avatars to perform the moves, thus making the comparisons avatar-centric. However, the WhoLoDance project has previously explored movement visualizations with different avatars and the way users interact with them in more than one case [2][4], so we opted for the environment-centric approach. Ambient sounds or intense visual elements can possibly focus the user’s attention to certain qualities of the movement and keep them from noticing others.

In order to investigate this hypothesis, we first had to select four sets of three dance moves, a set for each district, from the WhoLoDance [43] Motion Capture database [46]. The dance moves that constitute each set are chosen to have qualities that are also present in the movement of the natural element representing the corresponding district. To explain our choices, we tried to define each natural element’s characteristics using Laban’s Space Harmony theory, which was examined thoroughly in Chapter 2. However, it is interesting to note that the definitions presented below might not be compatible with all the forms of the natural elements. Different definitions might be appropriate for different visualizations. For example, this experience visualizes the water element as an environment of a calm and peaceful ocean or lake, which is not always necessarily the case. Water can be aggressive and fast, when it refers to a river or a waterfall or even as rough sea during a storm.

Before moving to the elements, it would be helpful to be reminded of some basic terms associated with Laban’s Space Harmony Theory:

- The term **“Kinesphere”** refers to the personal space, the space around us within reaching possibilities of the limbs without changing one's place**.**
- The term **“Levels”** refers to 3 different spatial levels people can move in**,** and the three different types of movers derived from those Levels**.** There are the ones who enjoy moving in the High Level, such as leaping and springing off the ground. There are those who enjoy moving in the Central Level, their bodies leading with more sensuous movement. And finally, there are those who enjoy moving in the Deep Level, who prefer more earth-bound movements.

For a more detailed presentation of the Space Harmony Theory you can revisit Chapter 2. The definition of the elemental movement types is presented below:

### The “Fire” element

This element often represents the qualities of “passion”, “power” and “intensity”. Fire expands very easily in terms of space, without following a particular direction. So the value of Laban’s **Space** motion factor can be considered to be Indirect. However there can be cases when fire moves in a very direct manner, when for example it feeds on a trail of gasoline.

When it comes to Laban’s Levels, the movement associated with the fire element could possibly be classified as a type of movement that covers the Deep Low level, just like a fire is based on a surface, but is also capable of penetrating all levels upwards, just like a fire can reach the sky and be visible from a great distance. It is quite obvious that when it comes to Laban’s **Weight** motion factor, the fire movement classifies as Light, a fact also confirmed by the ability of a fire to expand just with a tiny blow of wind.

The **Flow** motion factor can be considered to be Free, since fire is not an element to be restricted by any kinds of obstacles in its way. When it comes to the **Speed** motion factor, the movement associated with fire could be classified as Sudden. A fire might be static when not provoked by anything, but if there’s a wind blowing or the fire is ignited in any way, it can suddenly revive and expand very quickly.

Taking all of this information into consideration, the dance moves chosen from the database for the “Fire” movement are the following:

- Alegrias – Flamenco
- Multiple pirouettes in a circle – Ballet
- Pentozalis – Greek folk

### For the “Earth” movement

This element often represents the “ground-space**”** where creatures move. The earth element, can be strongly linked to the image of a person, object or creature moving only within their Kinesphere, since it is the element which represents the ground, a substance that stays static, and does not move like water, earth or fire would. So in a sense, the value of Laban’s **Space** motion factor can be considered to be Indirect, in a very abstract way.

When it comes to Laban’s movement Levels, the earth element is strongly associated with the Deep Low level, since it refers to the types of movement that are literally bound to the ground, such as crawling or plié. So in a sense, the value of Laban’s **Weight** motion factor for the earth element can be considered to be Heavy.

When it comes to deciding on Laban’s motion factor of **Flow** for the earth element, the decision is not as obvious as it was for the factors of Space and Weight, and could vary in different circumstances. Earth elements such as boulders and rocks are not considered to move fluidly, on the contrary, if forced to move and disengage from the ground, their movement is quite forced. On the other hand, earth elements like mud and soil can move fluidly. However, mud’s fluid movement is due to the fact that mud is made of soil and water**,** and soil’s fluid movement (or sand’s fluid movement) can be caused due to currents of wind, which lift the specs in the air.

The **Speed** motion factor varies as well. When heavy weight is used from high levels accelerating downwards it can produce high speeds. But considering that the earth element is associated with Deep-Low level movement (see above), the Weight factor would not be able to inflict major changes to the dynamic, and the speeds would supposedly remain low.

Considering all of the above, the dance moves chosen from the database for the “Earth” movement are the following:

- Directionality with a cube – Contemporary
- Weight on spot – Contemporary
- Zagorisios – Greek folk

### For the “Water” movement

In this section, we will be defining the movement associated with water. In the Motion Hollow experience water will be visualized in the form of a calm ocean or lake. However, the following definition will be examining all forms water can take, such as aggressive streams or calm lakes.

The water element often represents the qualities of “abundance” and “infinity”. It is considered to be a symbol for life. Water can expand easily in terms of space, as long as there is an open way for its stream to follow. So the value of Laban’s **Space** motion factor for the movement representing water can be considered to be Indirect, Flexible.

Water in the form of a calm ocean stays static, slightly moving parallel to the ground it is covering. So when it comes to Laban’s Levels, the movement associated with the water element certainly classifies as a type of movement that covers the Deep Low level, but it could also possibly reach the Mid Level depending on the extent to which the water level rises. It is quite obvious that when it comes to Laban’s **Weight** motion factor, the water movement classifies as Heavy, since water as an element is resistant to changes made to its position.

The **Flow** motion factor can be considered to be Free, since water is not an element to be restricted by any kinds of obstacles in its way. It can surpass everything because of its liquid form. When it comes to the **Speed** motion factor, the movement associated with water could in some cases be classified as Sustained, at least when comparing movement to the movement of a calm ocean, not agitated by a strong storm or a violent battleship. However, in the image of a rough stormy sea, or an aggressive waterfall, water could definitely classify as **Sudden** (or Quick).

So, the dance moves chosen from the database for the “Water” movement are the following:

- Directionality with a maximum step – Contemporary
- Weight shifting with slides, from side to side – Contemporary
- Rond de jambe – Ballet

### For the “Wind” movement

This element often represents the qualities of “ethereal” or “breezy”. Wind travels very easily through space. As a movement, wind usually follows a certain direction, as opposed to the air which floats statically in the space above our heads. So the value of Laban’s **Space** motion factor can be considered to be Direct.

When it comes to Laban’s Levels, the movement associated with the wind element would consist of moves such as jumping or springing off the ground, reaching for a level higher than the head. As a result, it can be classified as a type of movement that covers the High Level. It is quite obvious that when it comes to Laban’s **Weight** motion factor, the wind movement classifies as Light, a fact justified by the fact that wind consists of air particles only, and it does not have a tangible body.

The **Flow** motion factor's value is Free, since wind is not an element to be restricted by any kinds of obstacles in its way, just like fire. When it comes to the **Speed** motion factor, the movement associated with wind could be classified as Sudden, or in other words Quick, since wind is by definition the accelerated movement of air particles in the atmosphere. But in some cases, wind could be considered to be of Sustained speed, in the case of a light breeze coming and going.

We can observe that the Wind movement and the Fire Movement have many similarities. In fact, the only section they show any difference is the motion factor of Space.

Taking all of this information into consideration, the dance moves chosen from the database for the "Fire" movement are the following:

- Center jump – Ballet
- Center pirouette – Ballet
- Weight travelling in Space – Contemporary

Now that all the natural elements have been classified as certain types of movement, we have to distribute them among the districts of Motion Hollow. In order to test the hypothesis regarding the influence of the environmental surroundings, we decided to position all of the sets of "elemental" moves to a district bound to a different element. The pairings created are presented below:

- The "Fire" moves are presented in the Water District.
- The "Earth" moves are presented in the Wind District
- The "Water" moves are presented in the Earth District
- The "Wind" moves are presented in the Fire District

In the story, these switches are translated as the Empresses travelling to districts other than their own, but performing dance moves of their own element. This way, the user is required to annotate the movements with the qualities they identify, as objectively as possible. After the user has finished with all the moves of the district, they are asked a question like the following:

> *"Do you think that this empress is the empress of this District?"*

This question examines how much the user's answers are affected by the fact that the environment surrounding the empress might represent different qualities than the ones identified in the moves performed by her. It is important to note that there is no right or wrong answer neither in the annotation process nor at this question. The purpose of this study is to simply collect the data.

After answering this question, the user is returned back to the map.

### 3.3 Definition of the Target User Groups

Considering that the application collects data regarding movement qualities, it is very important for the integrity of the research that people with different kinds of relationships with movement are equally represented. This is why the targeted user groups should include people starting from “couch potatoes”, all the way to experienced dancers and athletes. It is not considered vital for the participants to have any kind of expertise in the field, although the participation of some experts would be highly appreciated. By experts we mean dance and athletics professionals like teachers, trainers and choreographers. Professionals with a high level of experience in the movement analysis field would also be desirable participants, however hard they may be to pursue.

For the definition of the targeted user groups, we created a generalized model that evaluates only the level of experience and the field which the user is experienced in (should it exist). Additionally, data such as the age and the sex of the participant will also be collected for statistical analysis purposes. That being said, a few possible primary target user groups are presented below:

**I. Dancers – (super-group)**

a. Beginners (1-3 years of experience)
b. Intermediate (3-6 years of experience)
c. Advanced (7 years and above)

**II. Athletes – (super-group)**

a. Beginners (1-3 years of experience)
b. Intermediate (3-6 years of experience)
c. Advanced (7 years and above)

**III. Martial Artists – (super-group)**

a. Beginners (1-3 years of experience)
b. Intermediate (3-6 years of experience)
c. Advanced (7 years and above)

**IV. Interested - Inexperienced Users**

These participants don’t have any type of particular experience in any field, but they are very interested in the concept of the project and what it represents and want to actively contribute to the results of the study. It’s an important user group, since we are interested to see how people from different backgrounds interpret qualities and produce comparative results.

It is important to note, that the levels defined as sub-levels in every super-group don’t represent the level of the user’s professionalism in the field, but the level of anticipation they are expected to have when it comes to movement, due to their respective years of experience.

When it comes to the age range of the participants, 15 was chosen as the minimum, because for most people 15 is considered to be the age that a young dancer or athlete starts to appreciate all the parameters affecting their movement, and the different energies and qualities that their body can produce depending on the circumstances. As for a maximum age, we didn’t choose one. For dancers or athletes, experience is not a trait that wears off over time. For inexperienced users, nothing changes over age. The only requirement when it comes to age is that all participants, even the most senior ones,  need to be familiar with modern day web technologies and can successfully navigate through a simple web browser application. The application testing is open for both female and male participants.

### 3.4 Creating Possible Personas

This section present some personas that could possibly represent the different user groups defined in the previous section. It’s crucial to create personas that are as realistic as possible to be able to define realistic user scenarios later on and eventually design a prototype that caters to targeted user groups. Otherwise, if the user doesn’t feel like the application is made for someone like them, they might end up not identifying and thus dropping the project. Some possible personas for the Motion Hollow experience are presented below:

**Marilia Papadopoulou, 17 years old**
Super Group I: Dancer
User Group B: Intermediate level of experience, non-professional

- High school senior, currently studying for finals. She wants to become a veterinarian
- She is half-Greek and half-Portuguese (by her mother’s side), but she was born and raised in Athens
- She lives with her parents and her brother in Argyroupolis, Attica
- She and her brother fight a lot because his is very protective of her, but she always manages to do what she wants in the end
- She used to be a ballet dancer. She started out when she was 10 years old and stopped at 16 due to heavy studying schedule
- She loves dancing and wishes to explore different styles after she gets into a University
- She has a boyfriend who is a freshman in university, and plays semi-professional basketball
- She has a black Labrador, Grover. She loves him to death.
- Her best friend’s name is Anthony. They’ve known each other since kindergarten

**Jean Terry, 24 years old**
Super Group I: Dancer
User Group C: Advanced level of experience, non-professional

- Sophomore in New York City School of Performing Arts
- He works part-time as a performer in a Theatrical Café in Manhattan
- He is from Marseille in France, but he moved to NYU when he graduated High School to pursue his dream of becoming a professional Ballet dancer

- He lives in a small apartment in Brooklyn with 2 roommates, one of whom is trying to make it as an actor and the other one is an aspiring writer
- He misses his family, but due to low income he can't afford to visit home the past 5 years
- He is currently not involved with anybody, since his last girlfriend broke up with him and moved to New Orleans
- He loves to walk up to the last dock of the Hudson riverside and watch the sun rise in the early mornings

**George Ioannou, 47 years old**
Super Group I: Dancer
User Group C: Advanced level of experience – professional

- He works as a ballroom dance teacher for over 22 years, but he has recently discovered that other genres might interest him as well
- He participates in Latin-Ballroom competitions abroad with his dance partner Anastasia, and has won third place twice at the Advanced tournament in Stockholm
- He lives in a one-bedroom apartment in Ampelokipous, Attica
- His family is originally from the island of Crete, and that's why he feels very connected to the Cretan traditional dances such as "Pentozalis", "Balos" etc., even though he doesn't teach this genre
- He is not married, nor has children
- He has a cat name Hoku which he forgets to feed sometimes
- He drives a Vespa scooter to move around the city

**Sophie Jameson, 39 years old**
Super Group IV: Inexperienced user

- She works as a salesperson at a clothing store downtown London
- She lives with her husband Adam and their two daughters, Anna and Brittany, in a semi-detached house in Tooting, London
- She regularly goes to the gym in order to maintain a healthy physique. After her second pregnancy she dealt with postpartum depression and gained some weight, so now that she's better she tries to keep in shape.
- She has enrolled both her daughters in ballet classes, because it was her childhood dream to take ballet classes, but unfortunately her parents couldn't afford to register her
- The idea of dancing seems very interesting and intriguing to her, even though she has no experience. She loves how someone can communicate their emotions to others using just their body movements
- On her day off she enjoys going to Latin bars in Brixton with her girlfriends and their husbands, and blow off some steam

**Konstantinos Georgiou, 28 years old**
Super Group I: Dancer
User Group B: Intermediate level
**and**
Super Group III: Martial Artist
User Group C: Advanced level of experience, professional

- He is a professional Judo instructor, who trains children of ages 6-17
- He lives with his life partner Athanasia and their two dogs in a two-bedroom apartment in Neos Kosmos, Attica
- When he doesn't have an actual class to teach at his gym, he voluntarily teaches self-defense courses for women at the community center of his neighborhood
- His favorite movie has always been Karate Kid, because it's what got him interested in the martial arts
- He has always been fascinated with how the strength and focus of the human brain, if used correctly, can make the body generate actions that require a lot more effort than someone could ever imagine of being capable of
- His family is originally from Epirus, so in order to connect with his roots he has been taking traditional Greek folk dance classes with his partner, for the past 5 years.

The personas described above are all quite realistic and relatable, imaginary people with whom the user can easily identify. The descriptions are a lot more detailed than we essentially need them to be, but the additional information is useful in order to paint a better picture of what the personas would be like as real people. There can be personas that fall into more than just one category, which is desirable since usually a user is never only part of one user group. People can have complex personalities with lots of interests and abilities, so it would be inaccurate to portray the personas as one-dimensional characters. It is worth noticing that every single one of the personas that could ever be constructed always belongs to the generalized target group of the **interested user,** no matter what other categories they might fall into.

### 3.5 User Scenarios and Hierarchical Task Analysis

In this case, the same Hierarchical Task Analysis (HTA) applies to all the possible users (personas), since the application has only one single goal for the user: the completion of the experience by the user, through which the collection of the necessary data will be succeeded. When it comes to the user scenarios, in our case defining one is enough, considering the purpose of someone participating in the experience is to contribute to the study. Additionally, the steps they will have to take in order to complete the experience are very specific.

In order to better illustrate our Hierarchical Task Analysis, we created a user scenario. Consider our third persona, George the dance teacher. We will create a possible scenario which will explain why George might be using our application, always keeping in mind that in our case such scenarios might be created around very specific circumstances, when, for example, the possible user is informed of our study and participates voluntarily. As a result, we can end up having slightly different scenarios for each persona, depending on their background, but all of them result in the exact same HTA. George's scenario is presented below:

"*George sat on his Vespa and drove to his dance class. He had 3 consecutive classes to teach that day, with the last one being Tango for adults. When the tango class came to an end, one of his students approached him. He told him that he was currently working on a web application in the context of a study conducted to determine the different qualities of movement using audio-visual stimuli and asked him if he would like to participate in the study by playing the game. George liked the idea and agreed to do so, and he asked for the website's link. As soon as he got home, he turned on his computer and opened up a browser. He navigated to the website by pressing on the link his student had given him. He then proceeded to the game by pressing 'Start'. He listened to the story and he navigated to the map of Motion Hollow. He listened to the instructions and then clicked on the Fire District. He was teleported to the Fire District where he met an Empress, a 3D dummy-doll model, who performed three dance moves. George observed each dance move and annotated the qualities he identified by marking true the clauses matching those qualities. He was then required to answer the question of whether he thought the empress was the ruler of the Fire District. He clicked 'No' and was navigated back to the map. He then proceeded with following the exact same steps for the rest of the districts. After completing all the districts and answering all the required questions, he arrived at the finish line of the game, where he was given the ending of the story, and he was thanked for his participation by a note. He then proceeded to complete a profiling form with his personal information and data regarding his experience. Satisfied with the experience, he closed the browser window and turned off his PC."*

George's scenario above is quite detailed, but in order to be able to define a precise set of technical requirements for the application it is essential to define a set of discrete steps George and any other user has to follow:

**Main Goal: Complete the experience to help his student with his research**
**Sub-goals:**

1. He opens a browser window on his pc
2. He types the website's URL in the browser's bar
3. He is welcomed by a screen with an introduction to the story of the experience, a brief explanation of the purpose of the study and a "START" button
4. He presses the "START" button and the experience starts
5. He is navigated to a screen with a digital book that narrates the story of the experience and gives context to the user, and a "NEXT" button which opens the book when pressed
6. He presses the "NEXT" and listens to the first chapter of the story. He is prompted with two buttons, a next "NEXT" button and a "BACK" button which flip the pages of the book forward and backwards respectively

**7.** He presses the “NEXT” and listens to the second chapter of the story. He is prompted with two buttons, a next “NEXT” button and a “BACK” button which flip the pages of the book forward and backwards respectively

**8.** He presses the “NEXT” and listens to the third and last chapter of the story. He is prompted with two buttons, a next “NEXT” button and a “BACK” button which flip the pages of the book forward and backwards respectively

**9.** He presses the “NEXT” and the book closes and George knows now what his mission is. He is prompted with a “GO” button.

**10.** He presses the “GO” button and he is navigated to a screen with a map of the town where he listens to some brief instructions of how to proceed. The map consists of links navigating to all four districts

**11.** He clicks on the Fire district.

**12.** He is navigated to the environment of Fire District, where the 3d model of an Empress is rendered.

**13.** The empress model performs a dance move and George is prompted with a set of 8 descriptions, all having the option to mark them as true or false. He marks as true the ones he thinks match the qualities he identifies by clicking the “YES” button for those. He marks as false the qualities he thinks don’t match the dance content by clicking the “NO” button. He leaves the ones he is not certain about unanswered.
**This step happens 3 times because the empress performs 3 dance moves at every district.**

**14.** After finishing the annotation process of the Fire district, George is prompted with the question “Do you think the empress is the ruler of the fire district? Along with two buttons, a “YES” button and a “NO”. After taking everything into consideration, George decides she is not, and presses the “NO” button.

**15.** He is navigated back to the map screen, where he is required to choose his next destination.

**Watch how steps 11-15 will be repeated by George 3 more times, one for the Wind District, one for the Water District and one for the Earth District.**

**16.** After visiting all the districts, George’s mission is complete. He is navigated to the final screen, where he is told the final chapter of the story, the ending.

**17.** After listening to the story’s finale, George is navigated to a screen with a form he is required to fill in with some necessary information.

  **a.** He locates the input field marked with the “Email” tag
  **b.** He fills in his email address
  **c.** He locates the input field marked with the “First Name” tag
  **d.** He fills in his first name
  **e.** He locates the input field marked with the “Last Name” tag
  **f.** He fills in his last name
  **g.** He locates the dropdown button marked with the “Age” label
  **h.** He chooses ‘47’ from the list
  **i.** He locates the checkbox feature marked with the “Gender” label
  **j.** He checks the ‘Male’ option
  **k.** He locates the button-group with the following options: ‘Dancer’, ‘Athlete’, ‘Neither'

  - **l.** He checks the ‘Dancer’ option
  - **m.** He locates the dropdown list marked with the label “Level of Experience”
  - **n.** He chooses “Advanced” from the list
  - **o.** He locates the input field marked with the label “Elaborate on your experience”
  - **p.** He fills in the textbox with his experience on the field of Dancing
  - **q.** He locates the button labeled “Submit feedback”
  - **r.** He presses the button

**18.** He closes the tab

**19.** He exits the browser

It is important to number the tasks as precisely as possible from early on in order to proceed to the requirement definition and specification export with a clear vision of the goal. When it comes to step 17 regarding the profiling and all of its sub steps (steps a-r), there is still flexibility. Changes might be applied to the questions the user is required to answer in the prototyping process.

### 3.6 Requirements Definition and Specifications Export

This section presents a set of requirements which must be covered in the application design for the user to be able to achieve their goal, based on the scenario and the HTA described in the previous section. A possible set of such requirements is the following:

- The purpose of the study must be explained clearly to the user, so as the user is acclimated with the project’s goal and understands why they are participating
- The user’s task should be easily comprehensible at any point during gameplay
- Since the purpose of this experience is to contribute to research, the experience shouldn’t be challenging for the user to complete. The user should have access to elaborate instructions and hints at all times during gameplay.
- Since the purpose of this experience is to contribute to research, the experience must be pleasant and fun, for the user to want to complete the whole thing.
- Given the age range and different possible technical backgrounds of our targeted user groups, the application’s gameplay should be simple and user-friendly. It shouldn’t take too much time for the user to familiarize themselves with the controls.
- Game mechanisms and controls should make use of the most commonly used ones by the average user, such as mouse-based navigation.

Based on the requirement set presented above we decided on the following specifications for the application:

- A text component explaining the purpose of the study and the goals of the experience, along with links to pages such as the WhoLoDance project website and the Wikipedia page for Laban Movement Analysis, should the user want to gain more context of the study. These components will be rendered on the first screen of the experience, the “welcoming page”.
- Audio narrations and instructions will be provided in order to explain the task to the user before they have to start the mission. A “Help” button will be accessible

at all times during gameplay, which will trigger an audio narration of detailed instructions. The instructions will also be rendered as a text component on the screen when the “Help” button is clicked.

- In order to make the experience pleasant and fun for the user to complete it, an engaging storyline will be narrated to the user in the form of a digital storytelling book. The user will be able to flip the pages of the book forward and back and listen to three chapters of the story as many times as they would like. Along with each chapter, illustrations will be rendered on the screen. Additionally, the whole experience will be visualized through beautiful graphics, which could potentially increase the interest of the user and get them to focus.
- The experience will be designed to utilize click-based navigation. This means that the user will be able to navigate themselves through the entire experience by using only their mouse.
- The navigation among the different screens and stages of the experience will be implemented with buttons the user will have to press. These buttons will be positioned in obvious and easily accessible positions.

### 3.7 Site map

A site map is a list of pages of a website. Site maps usually depict the connections among the website’s pages as well. This section presents a sitemap of the application to be developed. It consists of all the screens a user will navigate through while participating in the experience, starting from the starting/welcoming screen to the page including the profiling form. The sitemap in our case has a linear structure and is presented in Figure 11.

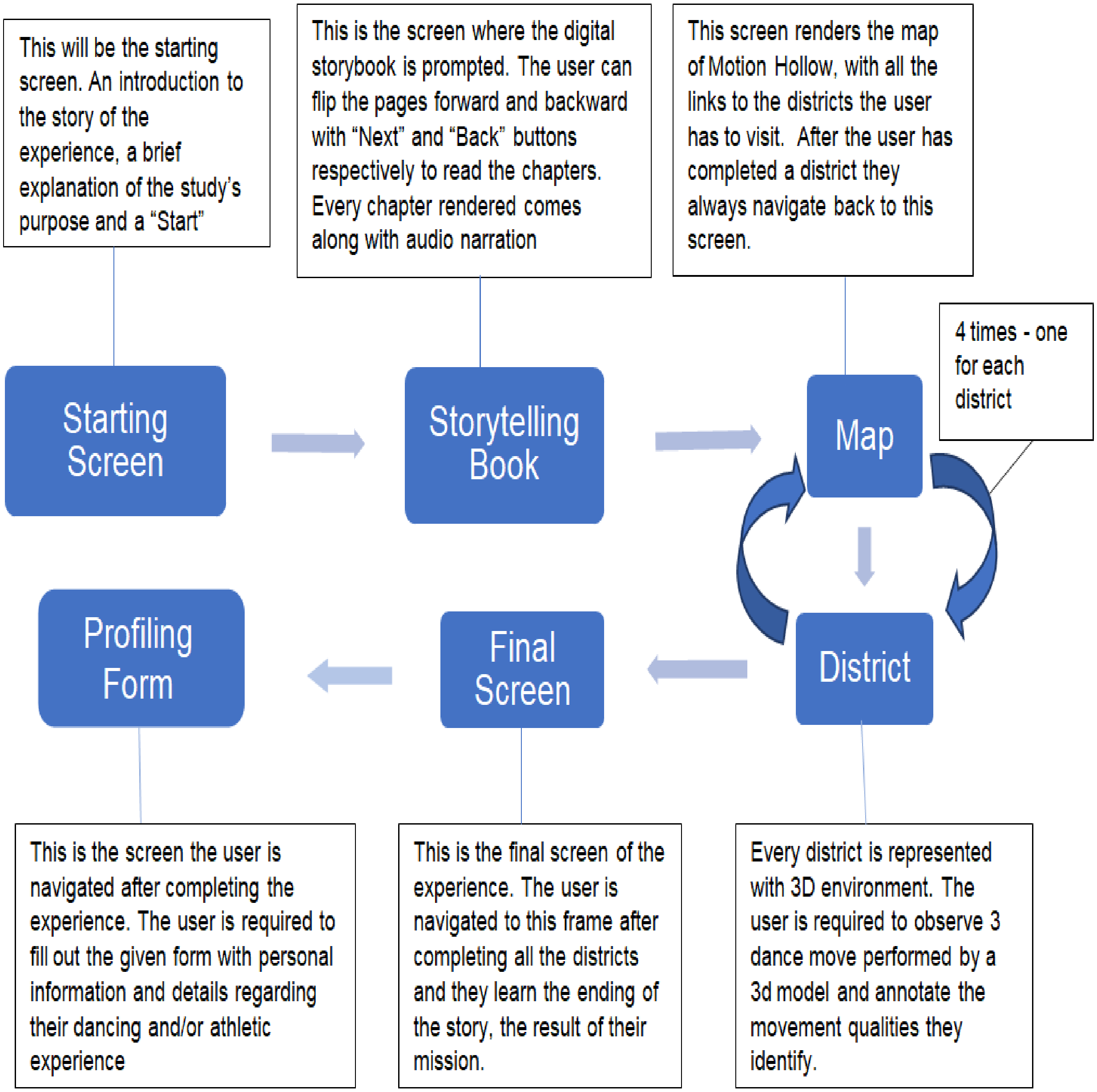
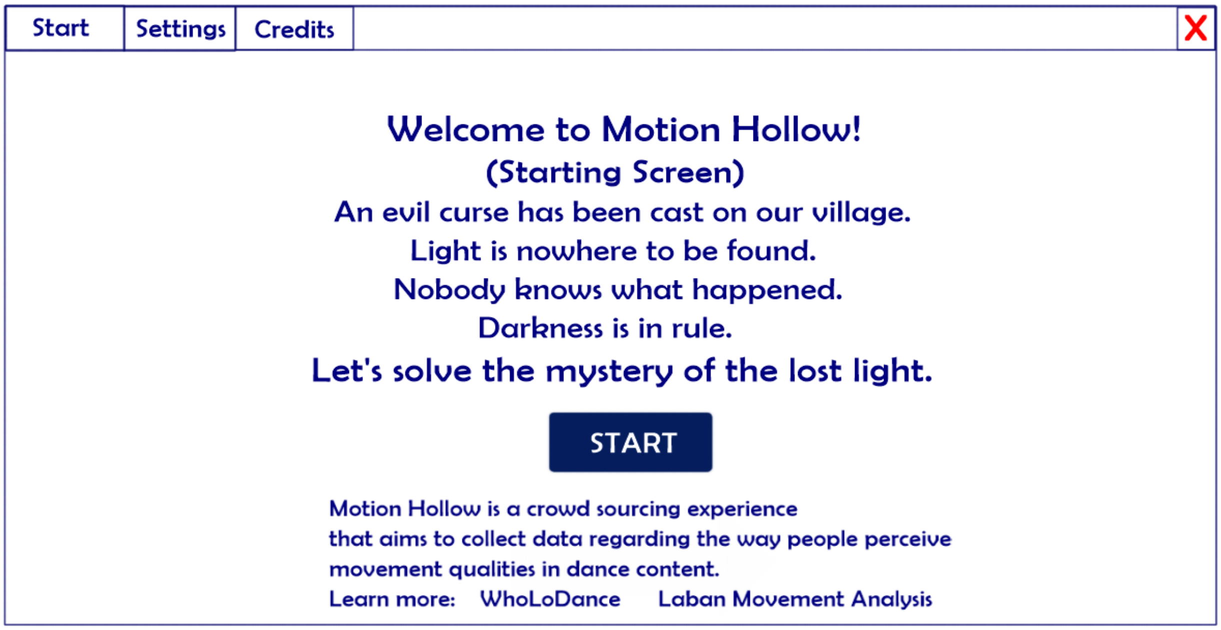

This will be the starting screen. An introduction to the story of the experience, a brief explanation of the study's purpose and a "Start"
This is the screen where the digital storybook is prompted. The user can flip the pages forward and backward with "Next" and "Back" buttons respectively to read the chapters. Every chapter rendered comes along with audio narration
This screen renders the map of Motion Hollow, with all the links to the districts the user has to visit. After the user has completed a district they always navigate back to this screen.
4 times - one for each district
Starting Screen
Storytelling Book
Map
Profiling Form
Final Screen
District
This is the screen the user is navigated after completing the experience. The user is required to fill out the given form with personal information and details regarding their dancing and/or athletic experience
This is the final screen of the experience. The user is navigated to this frame after completing all the districts and they learn the ending of the story, the result of their mission.
Every district is represented with 3D environment. The user is required to observe 3 dance move performed by a 3d model and annotate the movement qualities they identify.

**Figure 11: Sitemap of Motion Hollow application**

### 3.8 Low Fidelity Wireframes

Low fidelity wireframes are used to help map out the shell of the interface, its screens and basic information architecture. This section consists of wireframes for all the screens the user essentially navigates through respectively. Figures 12 to 19 present the low fidelity wireframes of the designed application.

## 1. Starting Screen

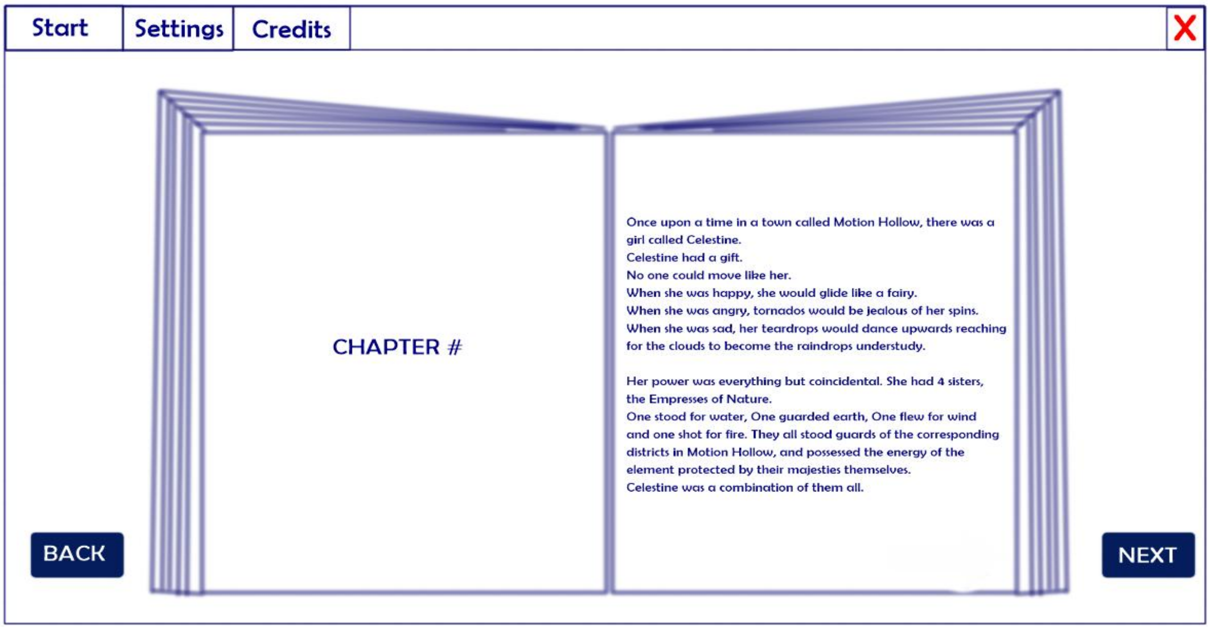


Welcome to Motion Hollow!
(Starting Screen)
An evil curse has been cast on our village.
Light is nowhere to be found.
Nobody knows what happened.
Darkness is in rule.
Let's solve the mystery of the lost light.

Motion Hollow is a crowd sourcing experience
that aims to collect data regarding the way people perceive
movement qualities in dance content.
Learn more: WhoLoDance Laban Movement Analysis

**Figure 12: Low Fidelity wireframes - Starting screen**

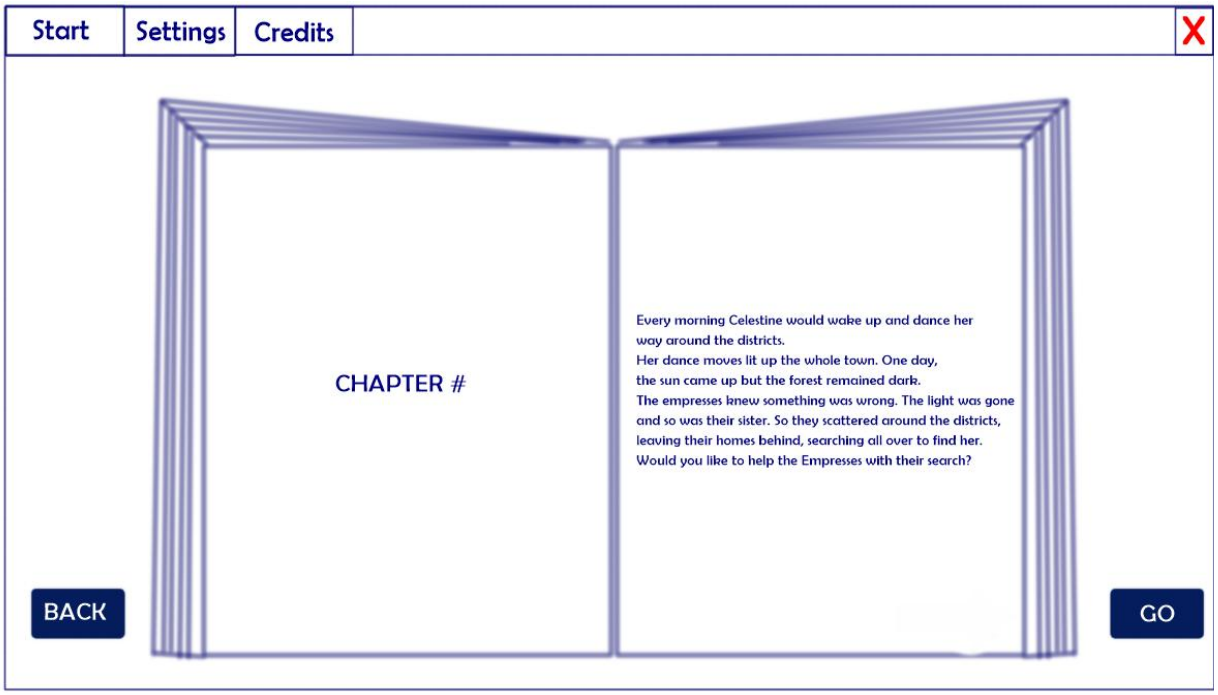
## 2. Storytelling Book



Once upon a time in a town called Motion Hollow, there was a
girl called Celestine.
Celestine had a gift.
No one could move like her.
When she was happy, she would glide like a fairy.
When she was angry, tornados would be jealous of her spins.
When she was sad, her teardrops would dance upwards reaching
for the clouds to become the raindrops understudy.

Her power was everything but coincidental. She had 4 sisters,
the Empresses of Nature.
One stood for water, One guarded earth, One flew for wind
and one shot for fire. They all stood guards of the corresponding
districts in Motion Hollow, and possessed the energy of the
element protected by their majesties themselves.
Celestine was a combination of them all.

**Figure 13: Low Fidelity wireframes - Storytelling book, in-between chapter**



Every morning Celestine would wake up and dance her
way around the districts.
Her dance moves lit up the whole town. One day,
the sun came up but the forest remained dark.
The empresses knew something was wrong. The light was gone
and so was their sister. So they scattered around the districts,
leaving their homes behind, searching all over to find her.
Would you like to help the Empresses with their search?

**Figure 14: Low Fidelity wireframes - Storytelling book, last chapter**

## 3. Motion Hollow Map

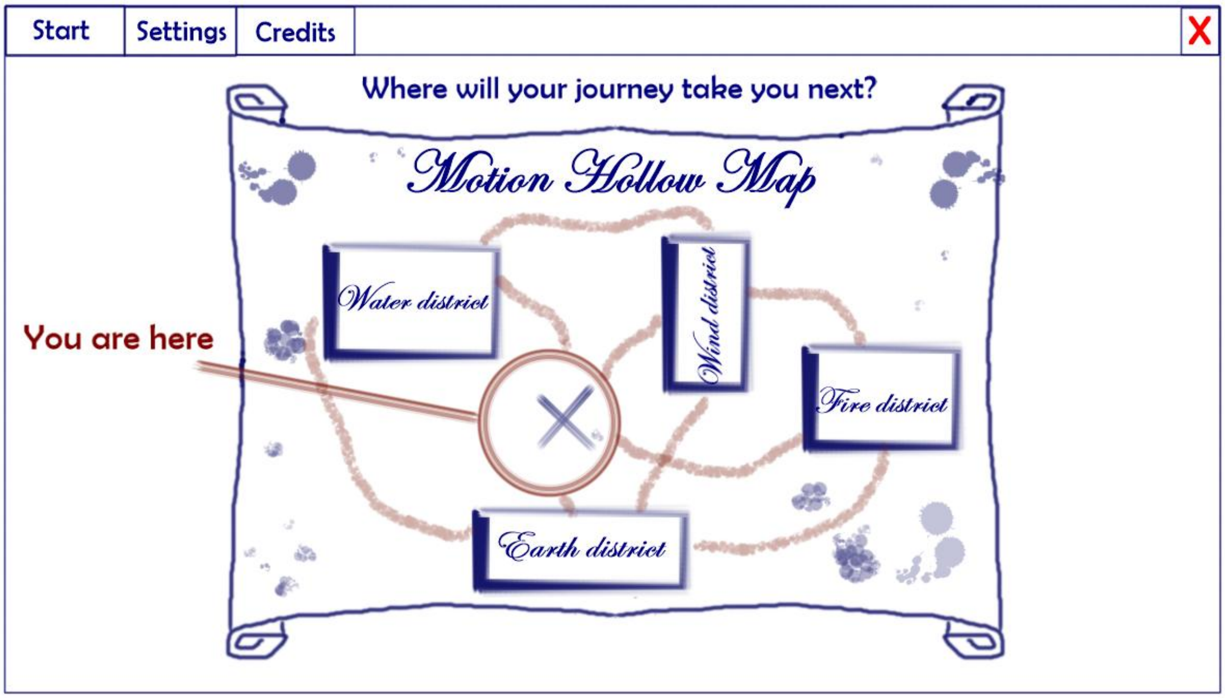
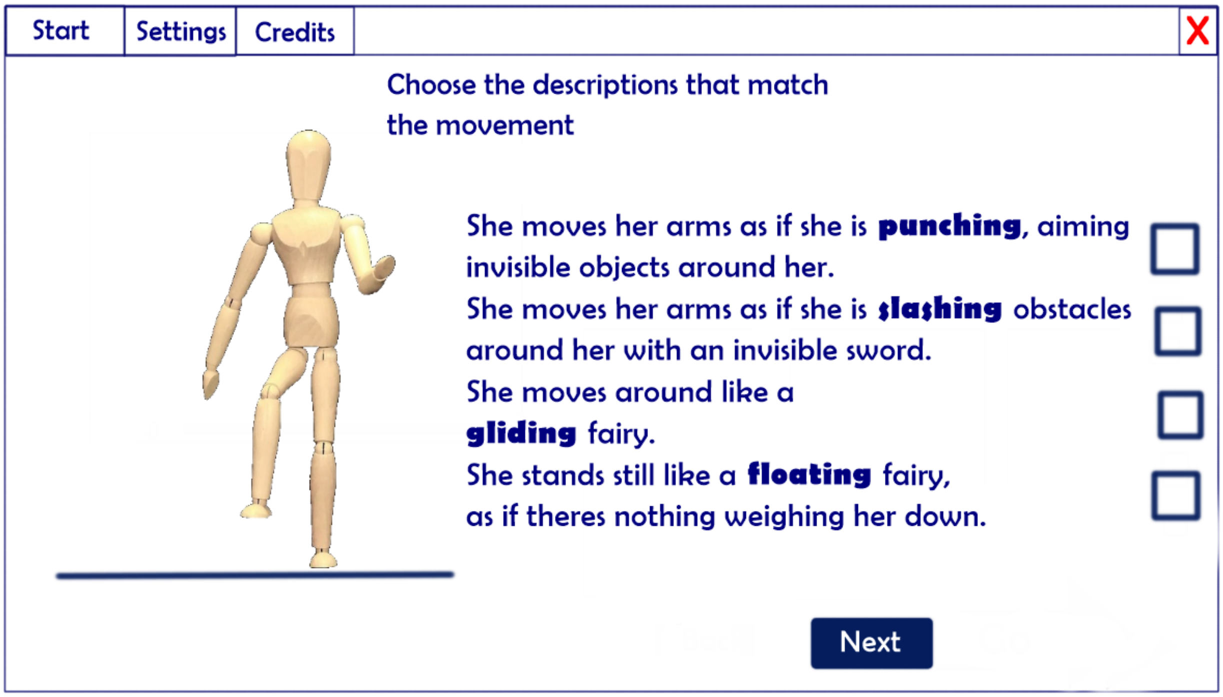


**Figure 15: Low Fidelity wireframes - Motion Hollow map**

## 4. District – Descriptions matching Laban's Efforts 1-4



Choose the descriptions that match the movement

She moves her arms as if she is **punching**, aiming invisible objects around her. □

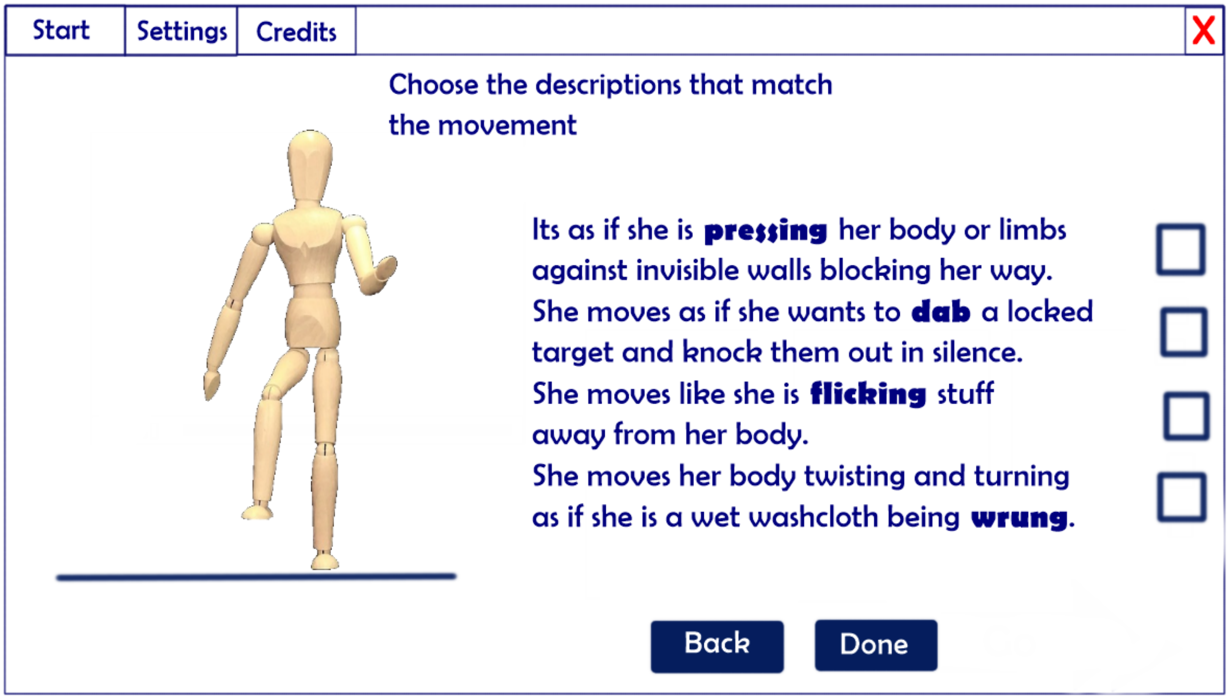
She moves her arms as if she is **slashing** obstacles around her with an invisible sword. □

She moves around like a **gliding** fairy. □

She stands still like a **floating** fairy, as if theres nothing weighing her down. □

**Figure 16: Low Fidelity wireframes - Annotation process, Efforts 1 to 4**

## 5. District – Descriptions matching Laban's Efforts 5-8



Choose the descriptions that match the movement

Its as if she is **pressing** her body or limbs against invisible walls blocking her way. □

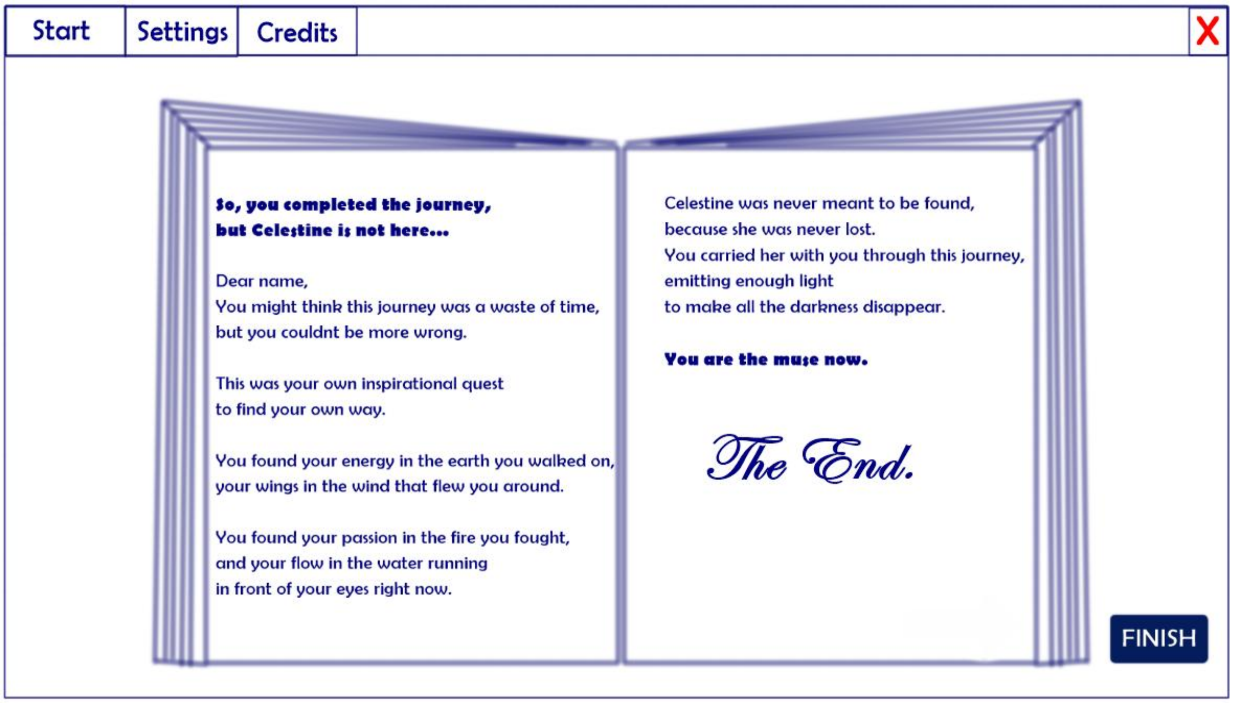
She moves as if she wants to **dab** a locked target and knock them out in silence. □

She moves like she is **flicking** stuff away from her body. □

She moves her body twisting and turning as if she is a wet washcloth being **wrung**. □

**Figure 17: Low Fidelity wireframes - Annotation process, Efforts 5 to 8**

## 6. Story Finale



**So, you completed the journey,**
**but Celestine is not here...**

Dear name,
You might think this journey was a waste of time,
but you couldnt be more wrong.

This was your own inspirational quest
to find your own way.

You found your energy in the earth you walked on,
your wings in the wind that flew you around.

You found your passion in the fire you fought,
and your flow in the water running
in front of your eyes right now.

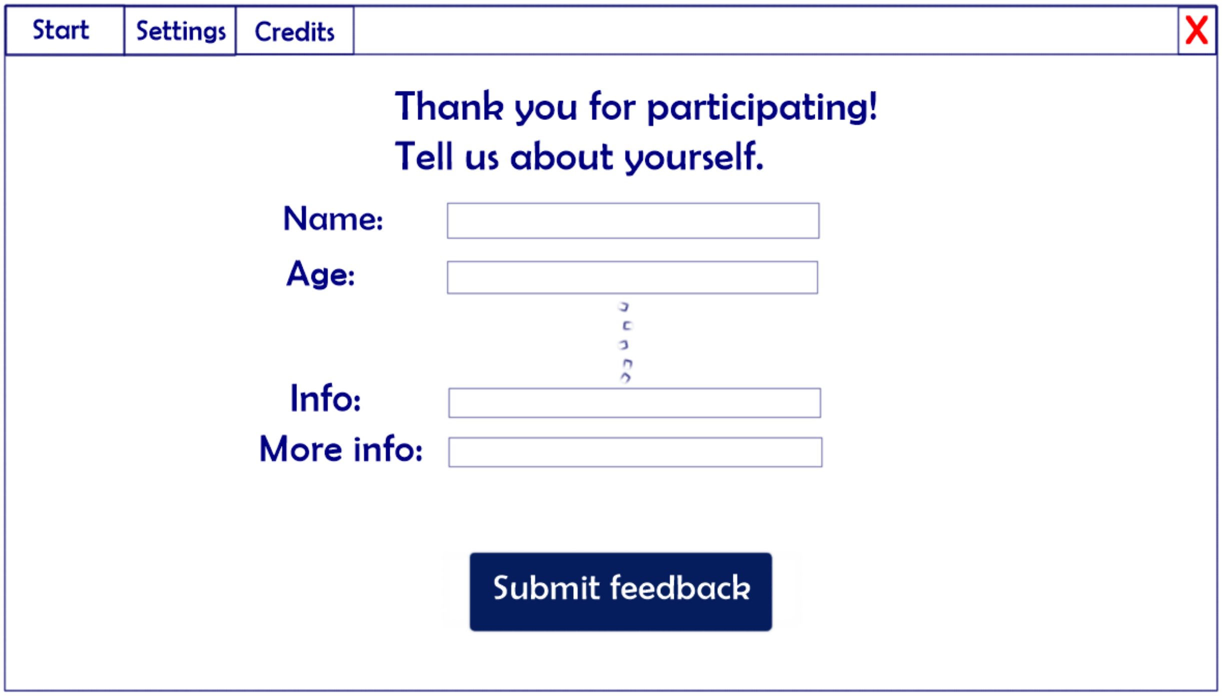
Celestine was never meant to be found,
because she was never lost.
You carried her with you through this journey,
emitting enough light
to make all the darkness disappear.

**You are the muse now.**

*The End.*

**Figure 18: Low Fidelity wireframes - Story ending after mission is complete**

## 7. Profiling Form



Thank you for participating!
Tell us about yourself.

Name:

Age:

Info:

More info:

**Figure 19: Low Fidelity wireframes - Profiling Form**

### 3.9 Adding Game Elements

Companies nowadays are “gamifying” or creating games around crowdsourcing problems, in order to motivate participants to engage in the experience and do their best for the crowdsourcing task. In general, adding game elements such as point scoring, competition with others, rules of play, to an application to other areas of activity, typically as an online marketing technique is claimed to encourage engagement with a product or service, while simultaneously enhancing user experience [34]. As any other crowd sourcing experience, the Motion Hollow experience could benefit from certain game elements. There are 12 game elements that are vastly used in Game Design in order to make the player more engaged to the game [33]:

- Conflict
- Collaboration
- Competition
- Strategy
- Chance
- Aesthetics
- Theme
- Story
- Resources
- Time
- Rewards/Scoring
- Levels

Experiences including all of the above might exist (maybe online role-playing games), however when it comes to our application only few of those could actually be applied. For example, elements like Collaboration or Competition could not be of any use in this case, since the experience is designed to support only single-player mode, at least for now.

One element that is already incorporated in the scenario of the experience quite clearly is the element of Conflict, which represents a challenge for the player to overcome. The task of the user is to search the town of Motion Hollow and search for Celestine, the Muse of Light. Every district the user has to visit and go through the annotation process, presents a challenge that needs to be resolved.

Elements like Time and Levels could actually make the experience more engaging and motivate the user to continue to play. However, in this case using the element of Time as it is traditionally used in the form of a countdown for example, would possibly make the users rush their answers during the annotation process in order to finish as quickly as possible. Since annotation relies by definition on observation, there is no need to complicate the process by adding a timer. The element of Levels is quite interesting, but is not compatible with our concept if used in the context of “the higher the level the harder the challenge”, since the process of annotating cannot be classified as hard or easy. The level of comprehensibility in annotation is mainly subjective. There is no right or wrong answer, the experience’s sole purpose is to extract information about the way the user understands the movement qualities when exposed to certain content. According to studies, engagement and motivation are conceptually distinct phenomena and the one influences the other. Engagement captures a player state that is characterized by deep cognitive and emotional absorption that is over once the playing session ends. Intrinsic motivation presents an inner drive that urges the user to get into and continue coming back to the activity [15][33]. For this reason the use of Rewards and/or a Scoring system in general could prove very helpful. It is always easier to commit to the scenario of a game when it is lined out in a series of goals and achievements. This element could be added in our experience in the form of Completion Badges and a Progress Bar. For example, when the user completes the annotation in the Fire District and returns to the map, they are awarded with a completion badge for the Fire Element and the progress bar increases, showing how much the user has left to reach their mission's goal.

Finally, the last and maybe most important game elements that should be incorporated in the Motion Hollow experience are the elements of Theme, Story and Aesthetics. Those 3 elements actually complement each other. The Aesthetics and the Theme should be revolving around the same concept. The Shapes, the colors and the general "Look and Feel" of the application will be thoroughly presented in the next sections.

In conclusion, the game elements that will probably be included in the application are the element of Conflict, the element of Story and the elements of Theme and Aesthetics. Finally, the elements of Rewards and Scoring will be used in a very abstract way, by implementing a system of Completion Badges and enabling a Progress Bar.

### 3.10 Defining the "Look and Feel" of the Application

The term "Look and Feel" is the initial design of a website or an application, which presents a visual of how the entire product will look before it is developed. It entails the following elements [35]:

The "look" is defined by the following components:

- Color palette
- Imagery
- Layout
- Font styles and choices
- Overall styling & spacing

The "feel" is determined by these characteristics:

- Components like dropdown menus, buttons, forms, portfolios, galleries
- Media
- Responsiveness

When it comes to the "Look" of the application, the proposed concept is creating an environment with a "fantasy" or "galaxy" theme. The elements are further analyzed below [35]:

- **Color palette**: Different variations of Blue, Pink, Purple, and Yellow.
- **Imagery**: A background that resembles a midnight-blue galaxy sky, with objects floating around. The buttons have a retro feel to them.

- **Layout**: A step-by-step completion experience, with an introduction to the story, the mission/task of the user and the story ending.
- **Font styles**: “dejavu”

When it comes to the “Feel” of the application, the components are defined below:

- **Components**: Navigation through the experience will be click-based navigation, mainly using buttons. A profiling form including dropdown-menus and text-field inputs will be provided when the experience is completed.
- **Media:** Button sounds and object interaction sounds will be used. Ambient environmental sounds will be used to enhance every district’s experience and the experience overall will be adorned with a background lullaby-like tune.

These elements help create a certain “mood” for the general “Look and Feel” of the experience. However, the aforementioned “mood” would be impossible to understand without seeing the different elements involved in some sort of assortment. In order to visualize the general idea, a **Mood Board** was created. Mood boards provide a mechanism for UX and UI designers to respond to perceptions about the problem as it emerges and the ideas as they develop. The construction of mood boards potentially stimulates the interpretation of more short-lasting elements such as color, texture, form, image and status [36]. Figure 20 presents the mood board which brings together the color palette, the font styles and elements similar to the ones that have to be created in order to capture the desirable “Look and Feel” for the Motion Hollow experience.

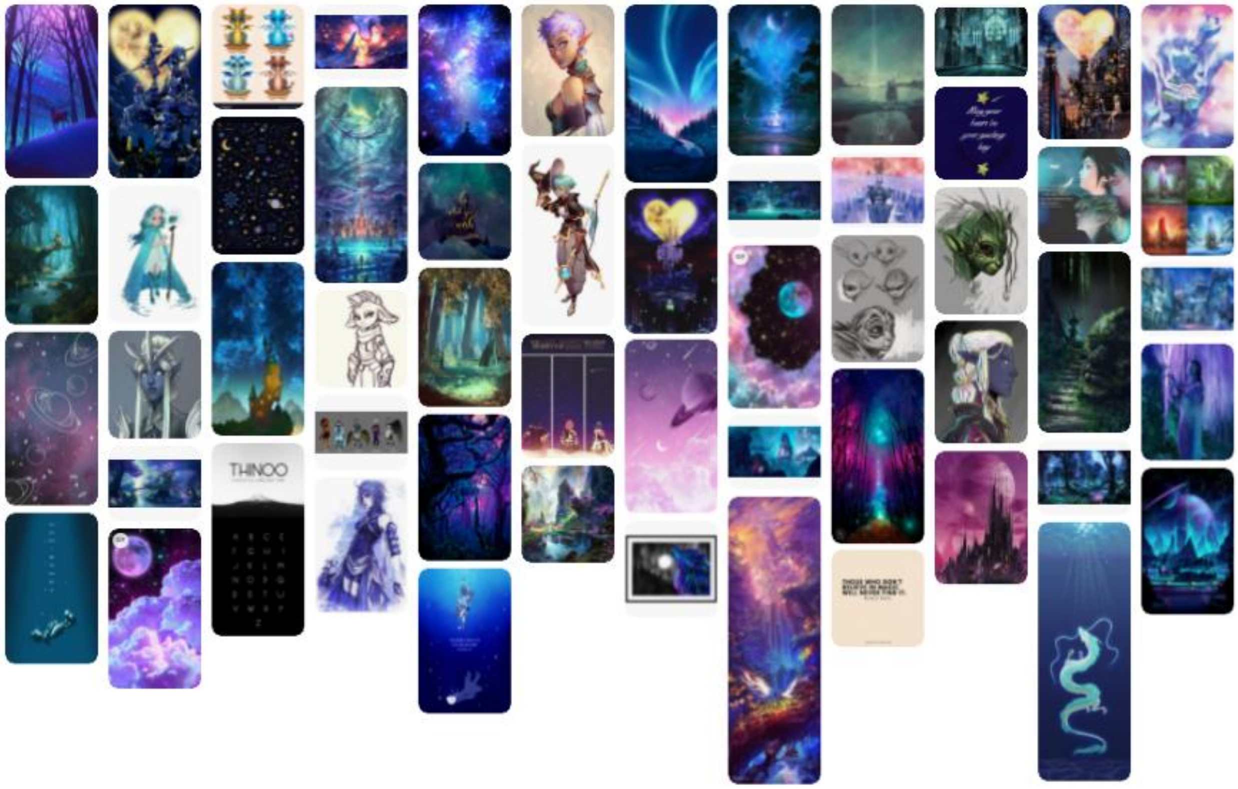


**Figure 20: Mood Board visualization for the Motion Hollow experience**

# 4. DEVELOPING THE PROTOTYPE

## 4.1 Introduction

This chapter consists of the sections associated with the development of the application we designed in the previous chapter. It consists of the following chapters:

- Development frameworks and technologies
- Animating the 3d models with Motion Capture data
- Developing the application
- Evaluation and Feedback
- Future Extensions
- Conclusions

## 4.2 Choosing the Development Framework

For the development of the application core, a MEAN stack model was chosen. MEAN is an open-source, free JavaScript software stack for building dynamic websites and web applications. The MEAN stack is set of four technologies in total, each being represented by one letter in the name of the stack. “M” stands for MongoDB [39], a NoSQL database program which uses JSON-like documents with schema. “E” stands Express.js [38], a modular web application framework package for Node.js. “A” stands for Angular [37], an application design framework and a development platform for creating efficient and sophisticated single-page apps. Last but not least, the “N” in the name stands Node.js [40], an open-source, cross-platform, back-end, JavaScript runtime environment that executes JavaScript code outside a web browser.

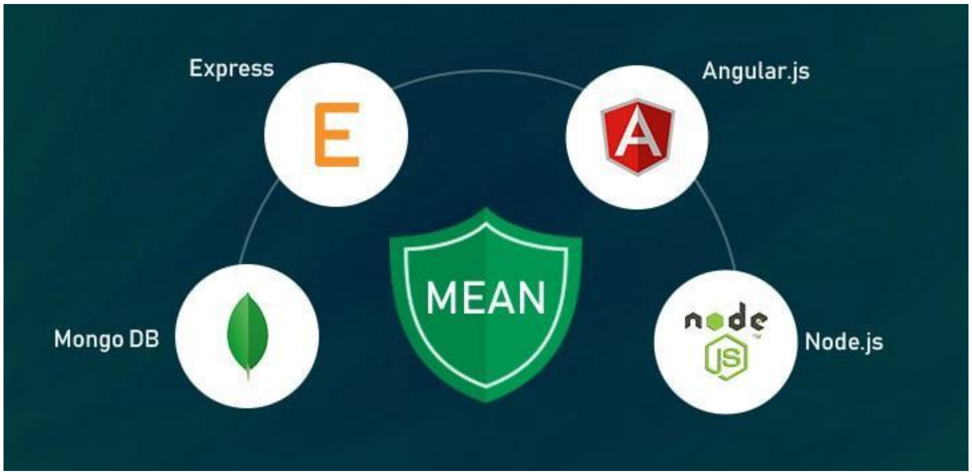


**Figure 21: The MEAN stack model**

After some consideration, we decided that the experience would possibly become a lot more interesting if it were to be developed in a WebVR environment. Aesthetically speaking, using a WebVR framework makes it possible for us to use 3-dimensional shapes and other components for the front-end development part, thus creating a more beautiful and lively UI. Additionally, since the Motion Hollow experience is designed to prompt the user with dance content, the 3-dimensional representation of space gained by the use of a WebVR environment would allow us to use 3d animated models for the performance of the dance moves instead of video, thus making performances more realistic. For all the aforementioned reasons, we decided to use A-Frame alongside the MEAN stack model. A-Frame [26] is a web framework for building virtual reality (VR), augmented reality (AR) and 3D experiences. A-Frame is based on top of HTML but its core is a powerful entity-component framework that provides a declarative, extensible, and composable structure to the three.js library [44] - a cross-browser JavaScript library and application programming interface used to create and display animated 3D computer graphics in a web browser using WebGL.

### 4.3 Animating the 3D model with motion capture data

As mentioned in Section 3.2.2, the moves chosen from the WhoLoDance Motion Capture library are the following:

1. **For the "Fire" movement**
    - Alegrias – Flamenco
    - Multiple pirouettes in a circle – Ballet
    - Pentozalis – Greek folk

2. **For the "Earth" movement**
    - Directionality with a cube – Contemporary
    - Weight on spot – Contemporary
    - Zagorisios – Greek folk

3. **For the "Water" movement**
    - Directionality with a maximum step – Contemporary
    - Weight shifting with slides, from side to side – Contemporary
    - Rond de jambe – Ballet

4. **For the "Wind" movement**
    - Center jump – Ballet
    - Center pirouette – Ballet
    - Weight travelling in Space – Contemporary

In order to make the annotation process as objective as possible, it was essential to choose a 3D model that does not affect the performance and does not take the user's focus away from the dance move itself during the annotation process. Thus, the model chosen is a white colored dummy-doll model. The white color was chosen to agree with all kinds of lighting the model could be exposed to in the Elemental District environments.

When it comes to the animation of the model with the motion capture data, Autodesk Maya [41] was used. Maya is 3D computer animation software with powerful modeling, rendering, simulation, texturing, and animation tools for artists, modelers, and animators. The process of the animation consists of 8 steps. Figures 22 to 29 provide screenshots of each step of the process.

1. Import the model into the Maya project.

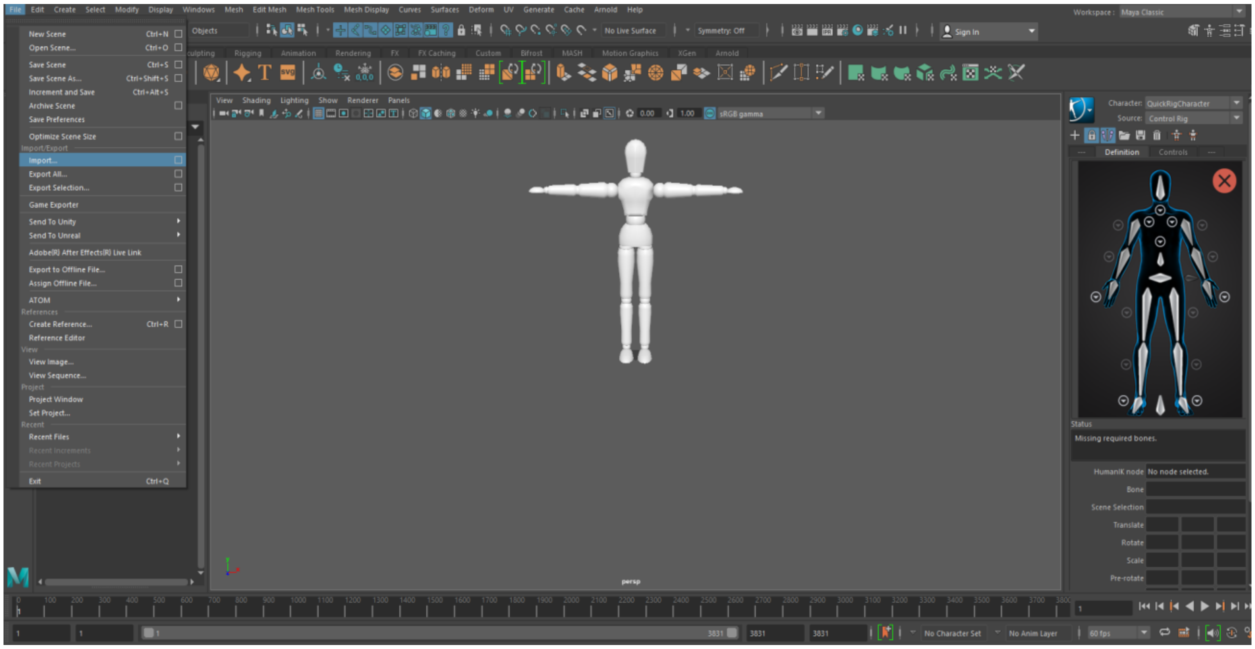


**Figure 22: Importing the 3d model into the project**

2. Generate a control rig for the model, by manually adjusting the joint and bone positions of the skeleton according to the model's anatomy.

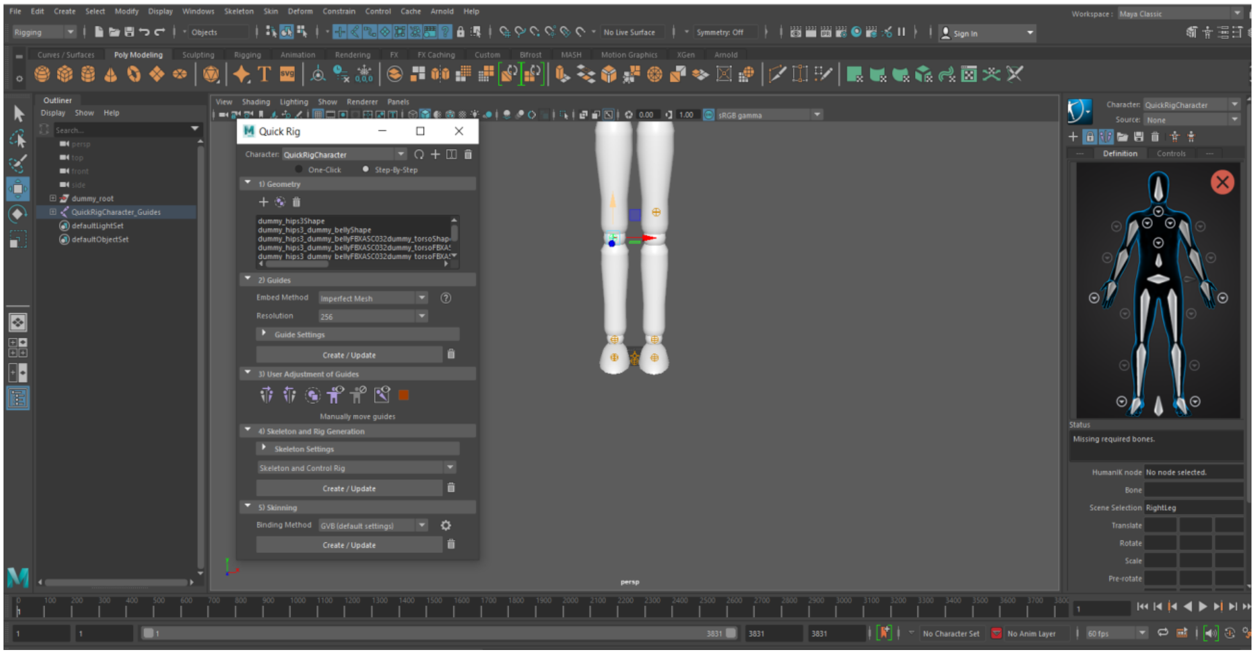
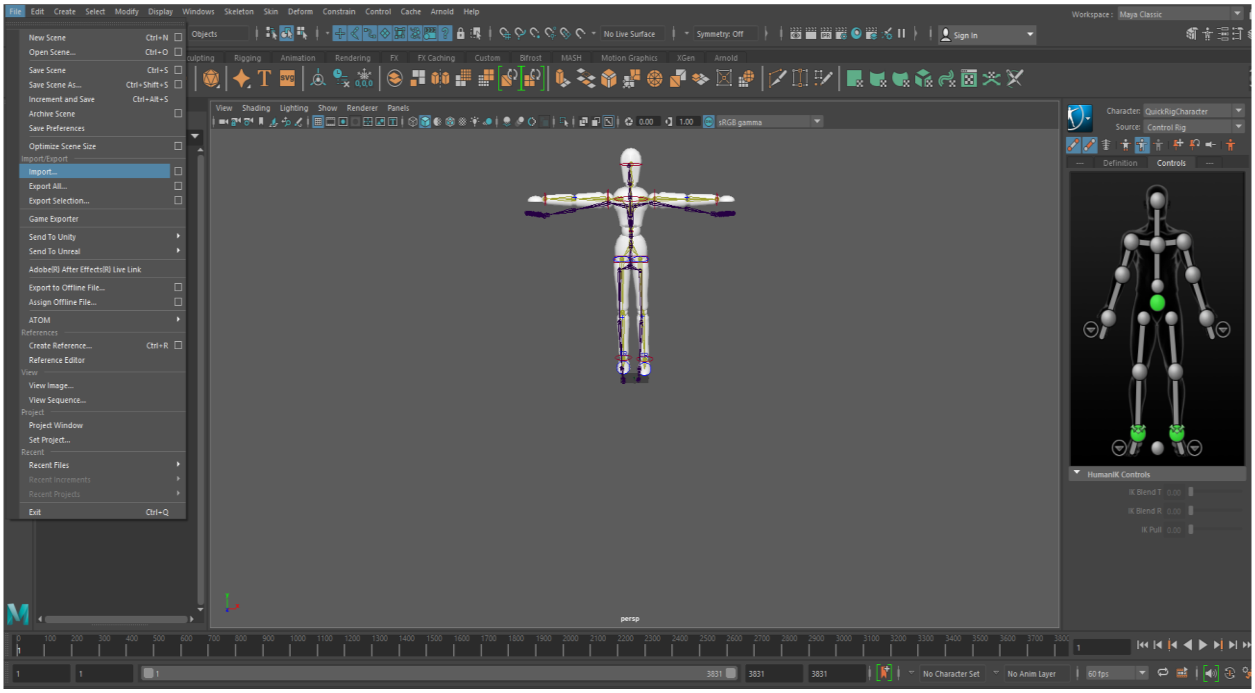


**Figure 23: Adding a control rig to the 3d model**

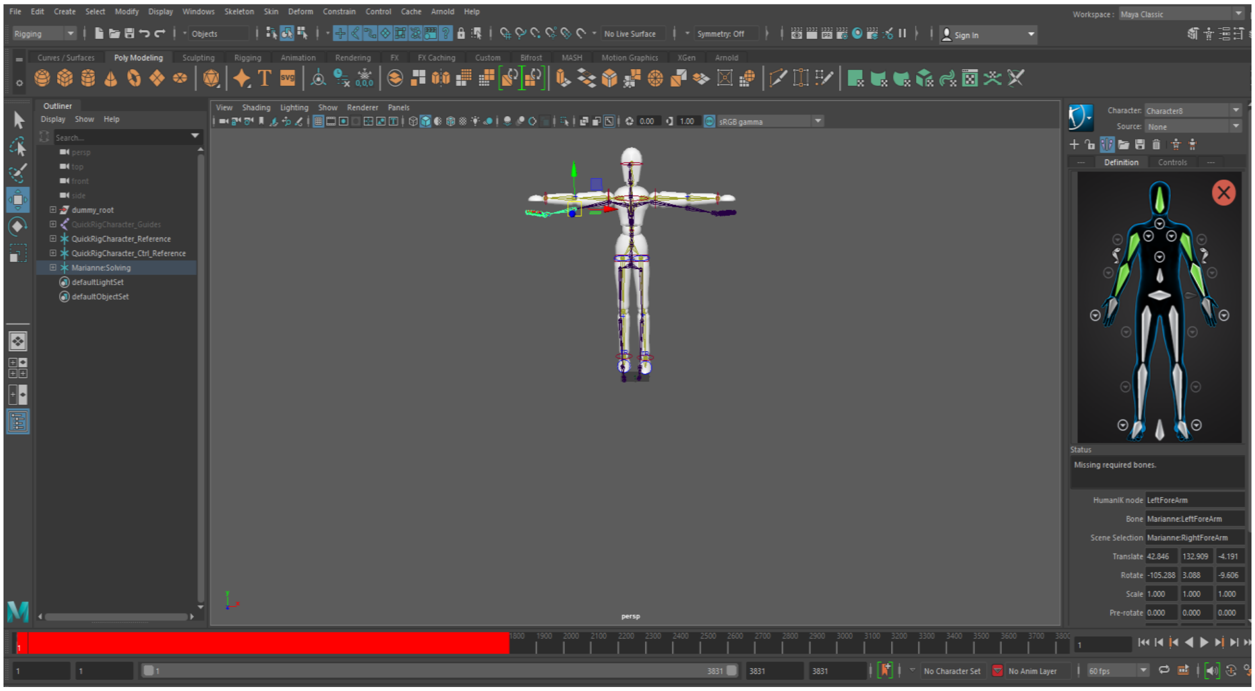
3. Import the FBX file containing the motion capture animation for the dance move.



Character: QuickRigCharacter
Source: Control Rig

Figure 24: Importing the animation file into the project

4. Create a character definition for the motion capture file, assigning all the body parts of the existing control rig to the definition's body parts.



Character: Character8
Source: None

Status
Missing required bones.

HumanIK node LeftForeArm
Bone Marianne:LeftForeArm
Scene Selection Marianne:RightForeArm
Translate 42.846 132.909 -4.191
Rotate -105.288 3.088 -9.606
Scale 1.000 1.000 1.000
Pre-rotate 0.000 0.000 0.000

Figure 25: Creating a character definition for the animation file

5. Switch the model's control rig with the control rig of the motion capture animation file, and check that the animation runs as it was supposed to run.

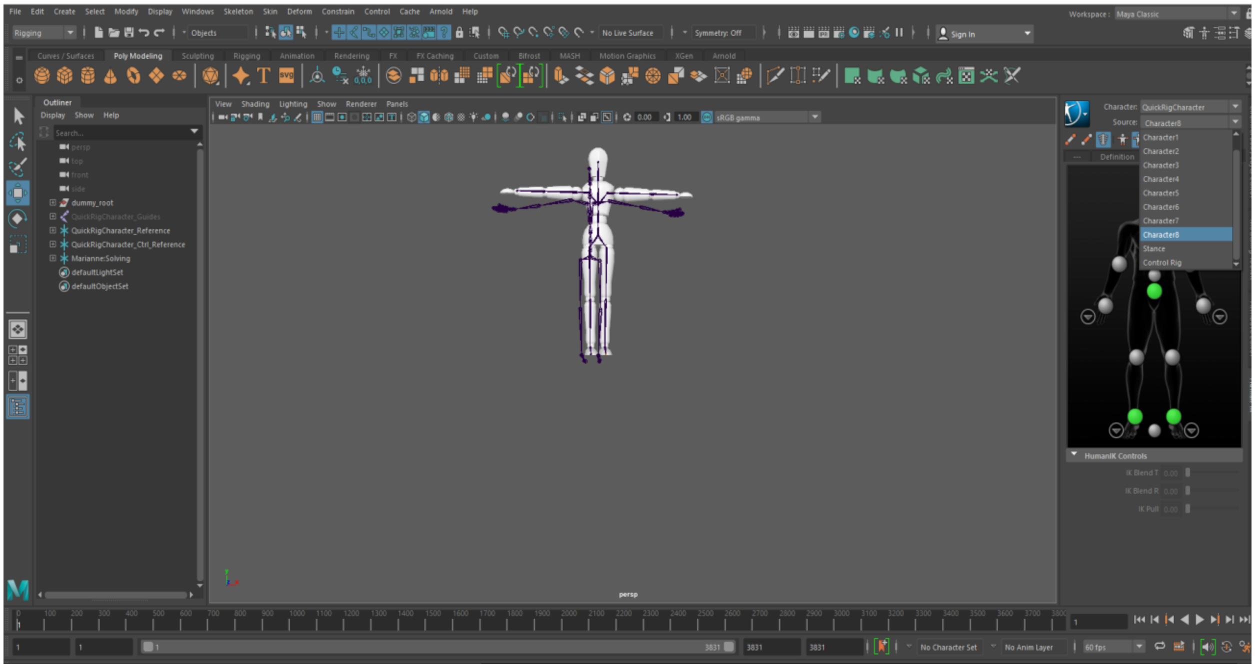


**Figure 26: Adapting the animated control rig to the 3d model**

6. Bake the animation to the skeleton of the model.

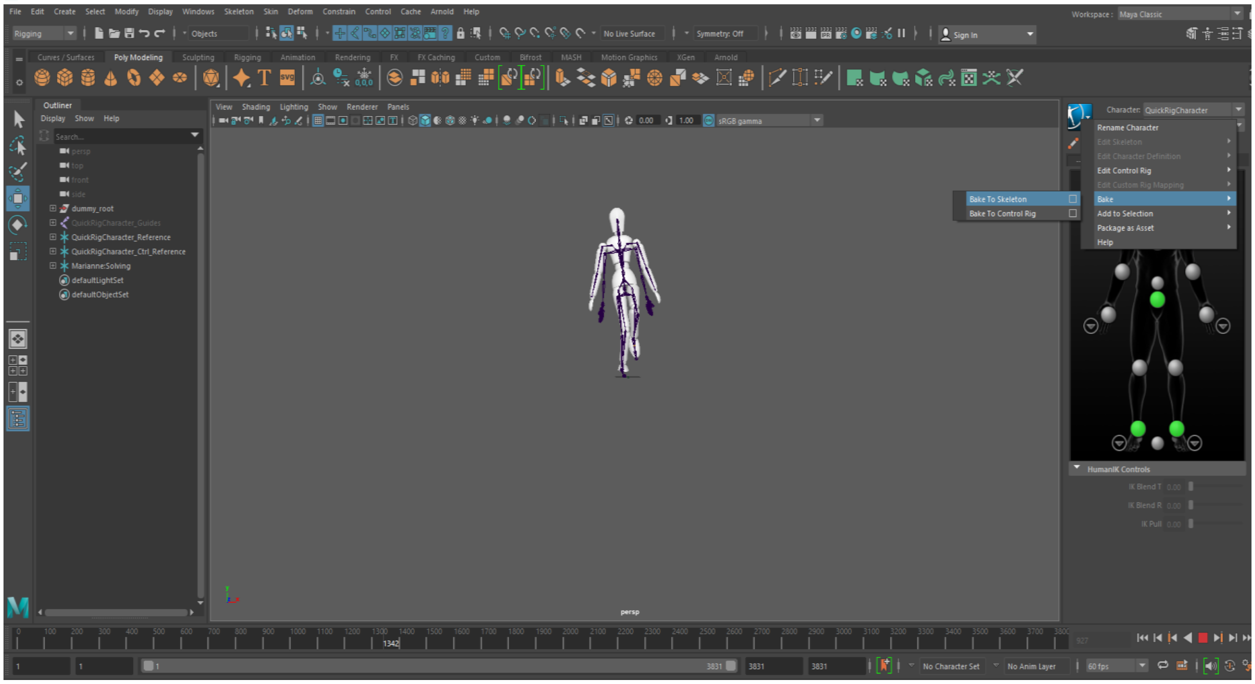


**Figure 27: Baking the animation to the 3d model**

7. Export the animated model to an FBX file.

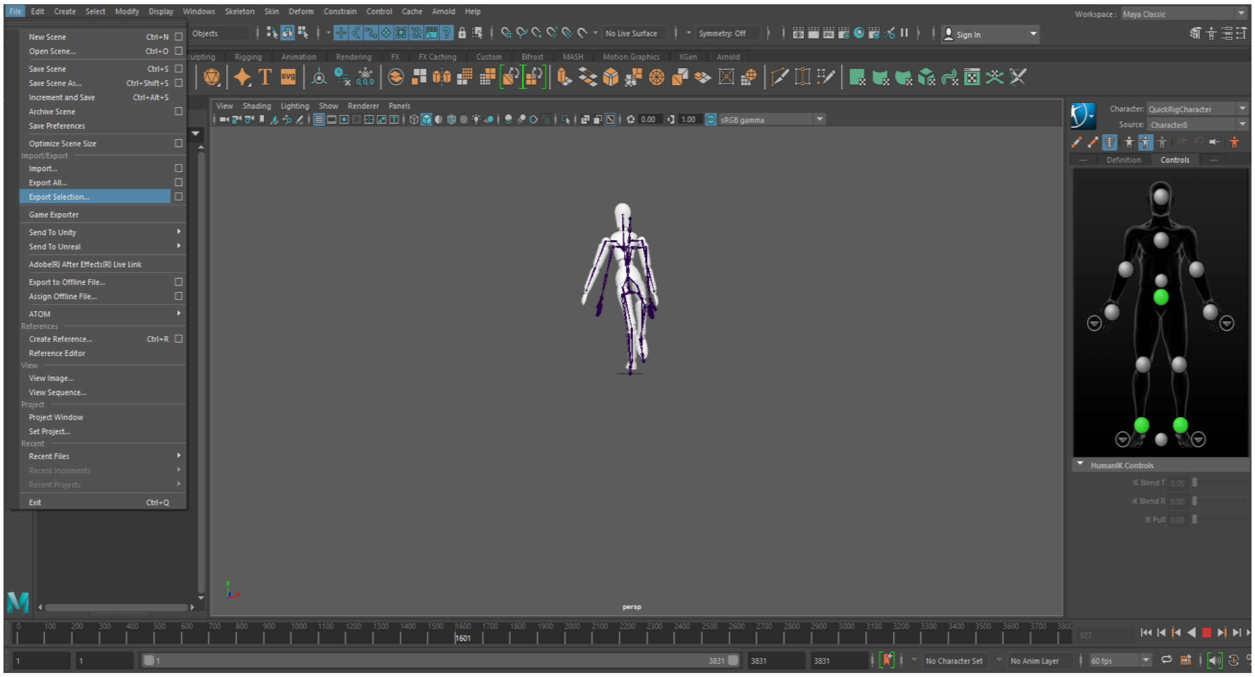


**Figure 28: Exporting the animated model to an FBX file**

8. Convert the exported FBX animated model to a glb/glTF 2.0 file, which is compatible with A-Frame.

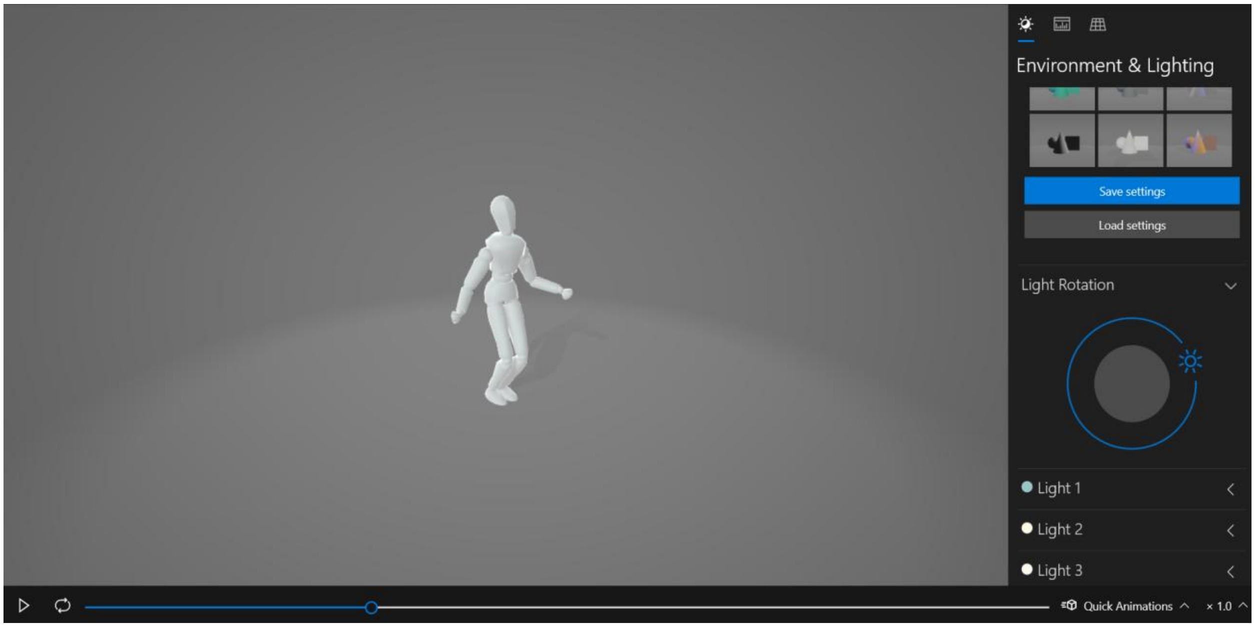


**Figure 29: The final animation file in glTF 2.0/glb format**

These steps (Figures 22-29) were followed for all twelve motion capture animation files.

## 4.4 Developing the Application

As mentioned in Section 4.4, the technologies used for the development are the MEAN stack model, along with the A-Frame framework. Essentially, the application is an Angular project with A-Frame running concurrently under the browser window. This section provides an insight of the structure of the project, and all the modules implemented. It also provides a presentation of the final product with screen captures.

### 4.4.1 The Structure of the Project

As any other MEAN stack project, this project also requires both front-end and back-end development. The front-end development is associated with everything regarding what is rendered on the page, components and services. On the other hand, the back-end takes care of everything that has to run in the back, setting up a server, the connection to the database, registering schemas and routes and getting HTTP requests through. The structure of the project is further explained below.

#### The front-end of the application

Since A-Frame can render only a single scene on the screen at once, the project is divided into two main modules, the module regarding the experience which requires A-Frame's 3D scene, and the module regarding the profiling stage which is rendered in Angular's DOM and does not require a 3D scene. The structure of the front-end is presented in detail below:

- The game module, which consist of the following files:

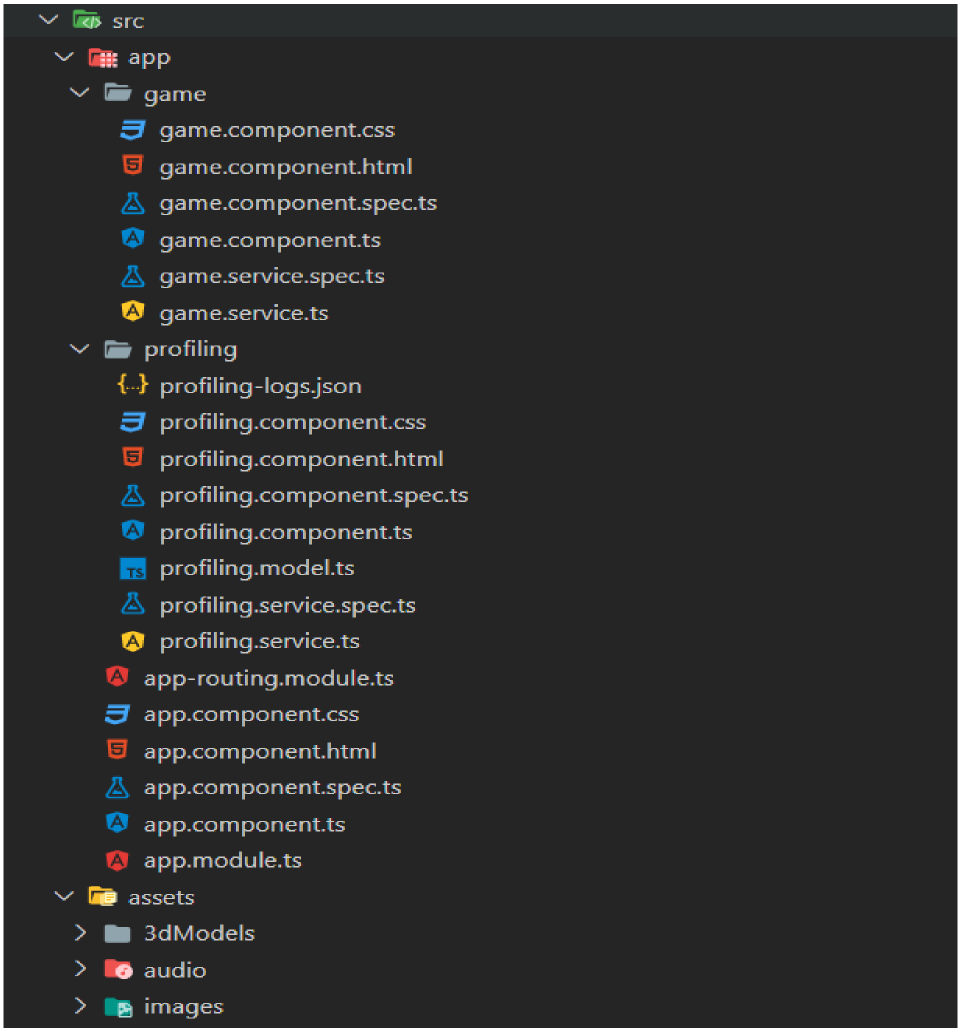
  1. **game.component.html**, is the file which contains the A-Frame 3D scene. This means that it contains all the elements that are rendered on the page during the experience.

  2. **game.component.ts**, is the file which contains all the functions implementing features provided by the application, the functions implementing the different functionalities the experience requires and the functions implementing the transitions between different states and screens of the experience. It also contains all the state-changing variables, and data objects used in the application.

  3. **game.component.css**, is the file that contains the styling decisions regarding a particular component.

  4. **game.service.ts,** is the file where we have registered all the A-Frame components. An A-Frame component, just like an Angular component, holds a bucket of data in the form of one or more component properties. Components use this data to modify entities. The components registered in this service are also necessary for the modification of state-defining variables, and function calls which are located in the game.component.ts file.

- The profiling module, which consists of the following files:
  1. **profiling.component.html**, is the file which contains the profiling form that the user is asked to fill out at the end of the experience. The form is an Angular reactive form and is divided in four steps with an Angular Material stepper [45].
  2. **profiling.component.ts**, is the file which contains the form controls necessary for the functionality of the profiling form, all the functions required for updating the value and the validity of these form controls and the submission of the form.
  3. **profiling.component.css**, is the file that contains the styling choices regarding a particular component.
  4. **profiling.service.ts**, is the file which contains the function for the registration of the user to the database the application is connected to, meaning that it sends a POST request to the application's back-end through the appropriate route.
- The assets folder, which is essential in this case, since it stores all the illustrations, the audio files and the 3d models used in the annotation process.



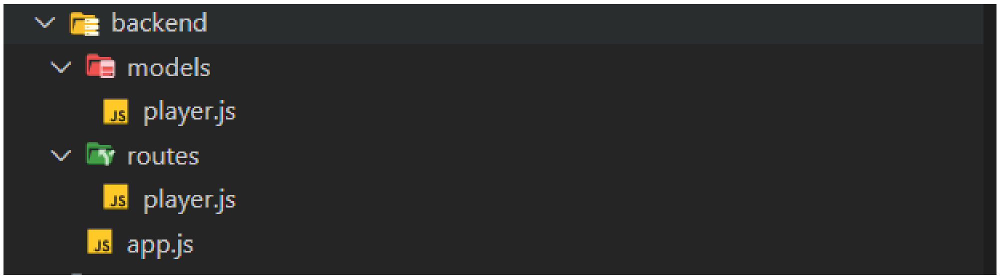
```
v src
  v app
    v game
        game.component.css
        game.component.html
        game.component.spec.ts
        game.component.ts
        game.service.spec.ts
        game.service.ts
    v profiling
        profiling-logs.json
        profiling.component.css
        profiling.component.html
        profiling.component.spec.ts
        profiling.component.ts
        profiling.model.ts
        profiling.service.spec.ts
        profiling.service.ts
      app-routing.module.ts
      app.component.css
      app.component.html
      app.component.spec.ts
      app.component.ts
      app.module.ts
  v assets
    > 3dModels
    > audio
    > images
```

**Figure 30: Architecture of the front-end implementation code**

### The back-end of the application

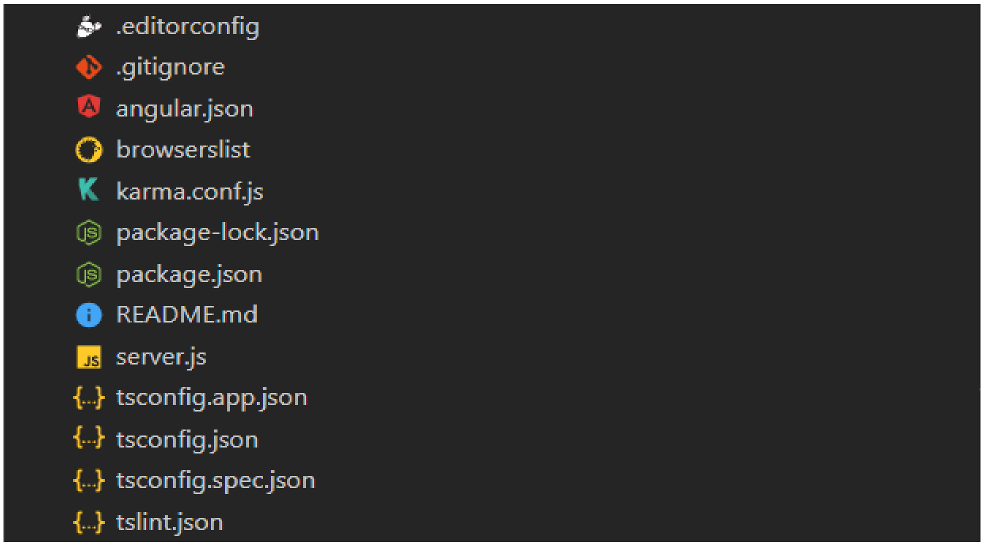
The back-end of the application is divided into the following parts:

- A folder for the models. In general, this folder contains the JavaScript files which are created to define and export schemas for the respective collections in the MongoDB database. It includes a file called **player.js**, which defines and exports the schema used for the collections where all the registered players are stored.
- A folder for the back-end routes. Generally, this folder contains the JavaScript files created to define the routes used by the application to handle different HTTP requests to the various collections of the database. In this case, this folder contains a file called **player.js**, which defines the route that handles the POST request sent after the submission of the profiling-form during the registration of a new user.
- The **app.js** script file, which is responsible for establishing a connection to the MongoDB database. It also tells the application which back-end routes it should use.
- The **server.js** script file, which is called externally in Node.js and sets up the server the application listens to for any possible connections.



```
backend
  models
    player.js
  routes
    player.js
  app.js
```

**Figure 31: Architecture of the back-end implementation code**



```
.editorconfig
.gitignore
angular.json
browserslist
karma.conf.js
package-lock.json
package.json
README.md
server.js
tsconfig.app.json
tsconfig.json
tsconfig.spec.json
tslint.json
```

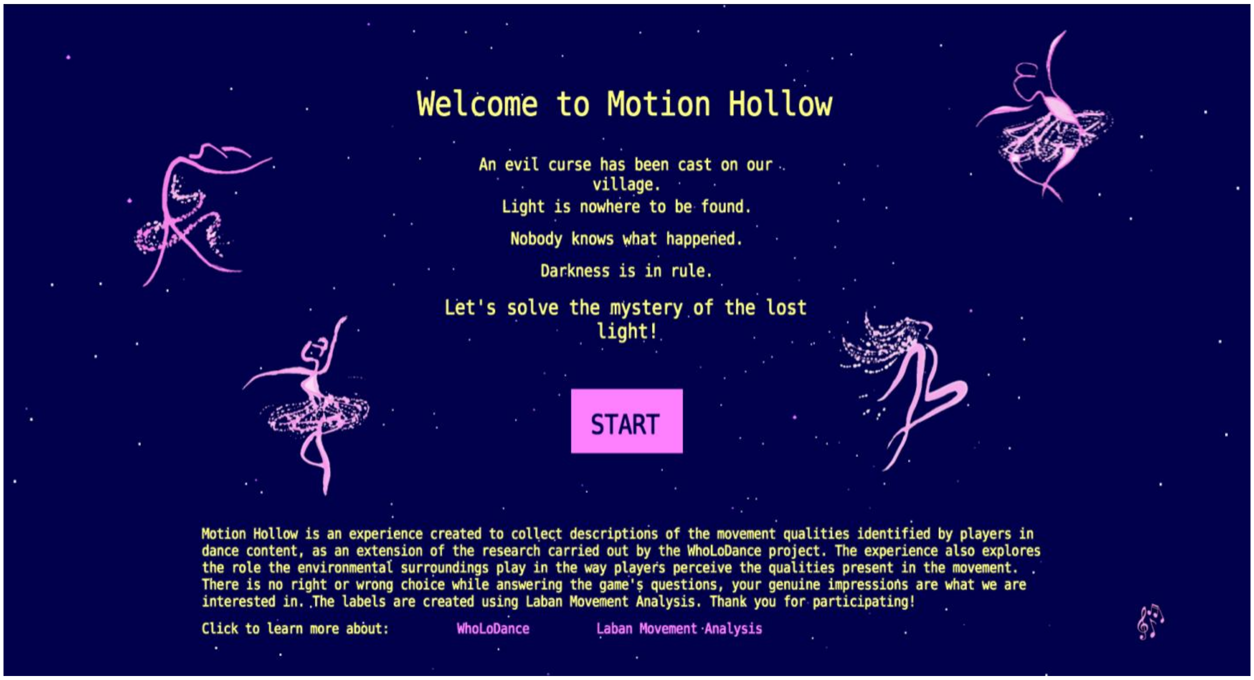
**Figure 32: Location of the server.js script file**

### 4.4.2 Presenting the Developed Prototype

This section provides a presentation of the developed experience prototype at its final stage, by going through all the screens of the experience. It is quite interesting to note that the application was developed following the steps defined in Chapter 3 very closely, a fact confirmed by the similarity the screens of the final experience show to the Low-Fidelity wireframes presented in Section 3.8. To make the presentation more realistic, we will be presenting the screens of the experience in the exact order they are displayed during gameplay, just like a walkthrough. A video game walkthrough is an instruction-based guide which aims towards helping a player to complete a particular stage of a game or the entire game [42].

Unfortunately, since this presentation consists of screenshots and text-based explanations, the animations, audio effects and animated transitions between the consecutive screens cannot be included. Sample code parts of the developed application can be found in the Appendix.

### 1. Starting Screen



Welcome to Motion Hollow

An evil curse has been cast on our village.
Light is nowhere to be found.
Nobody knows what happened.
Darkness is in rule.

Let's solve the mystery of the lost light!

Motion Hollow is an experience created to collect descriptions of the movement qualities identified by players in dance content, as an extension of the research carried out by the WhoLoDance project. The experience also explores the role the environmental surroundings play in the way players perceive the qualities present in the movement. There is no right or wrong choice while answering the game's questions, your genuine impressions are what we are interested in. The labels are created using Laban Movement Analysis. Thank you for participating!

Click to learn more about: WhoLoDance Laban Movement Analysis

Figure 33: The Motion Hollow experience - Starting screen

This is the welcoming page of the experience (Figure 33). It provides an introduction to the story and a brief explanation of the purpose of the study. It also offers links to the WhoLoDance project website[1] and to the LMA Wikipedia page at the bottom of the screen. Additionally, there is a button for starting and pausing the background music of the experience at the bottom-right of the screen. The sound tweaking button is present on the screen at all times, except for the districts where the background music is

[1] http://www.wholodance.eu/

inactive in order to give space to each district's environmental sound. Now let's say that the user clicks on the START button.

## 2. Storytelling Book

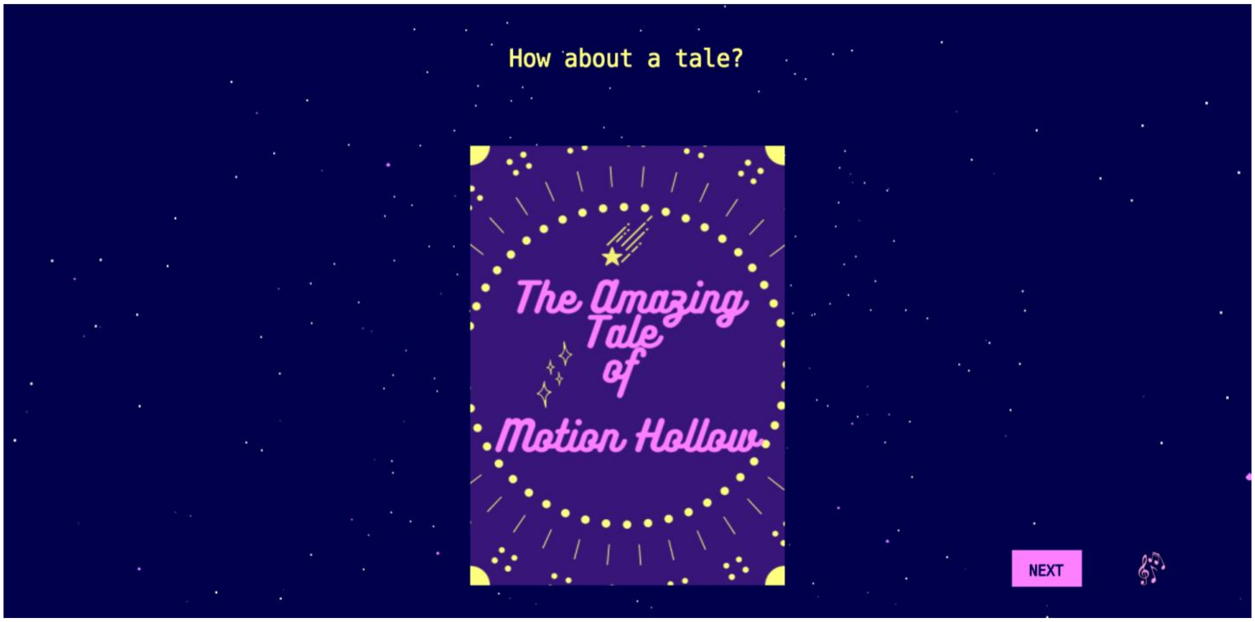
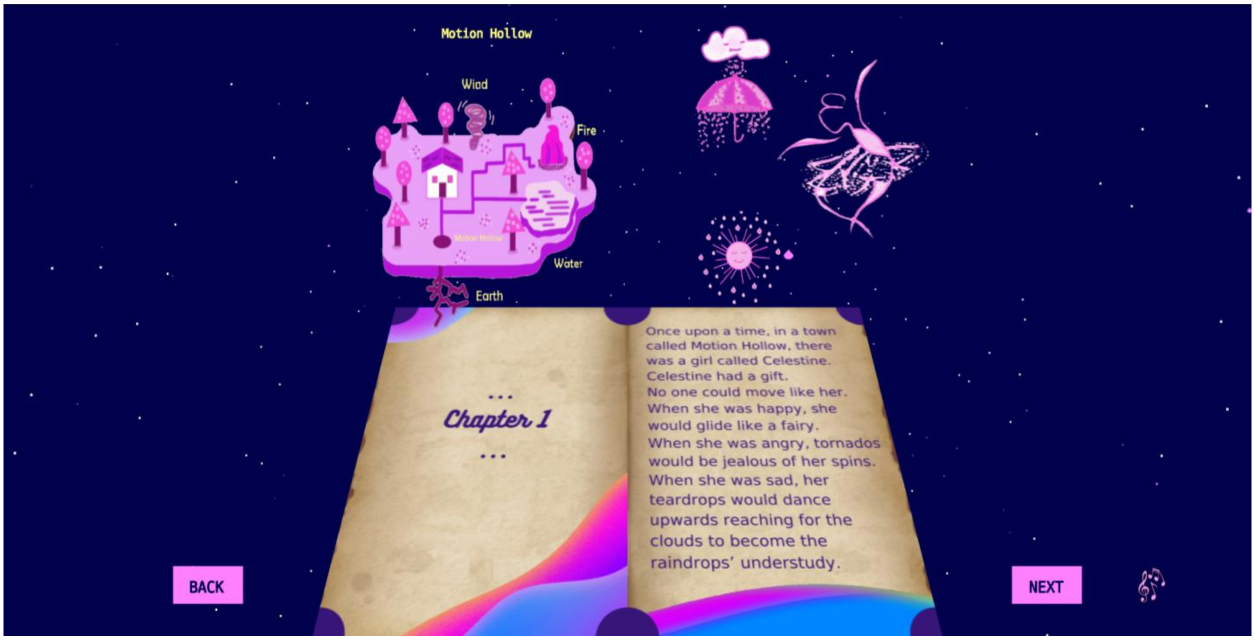


**Figure 34: The Motion Hollow experience – Book, Not opened**

The user is given a book (Figure 34). They click on the NEXT button to open the book.



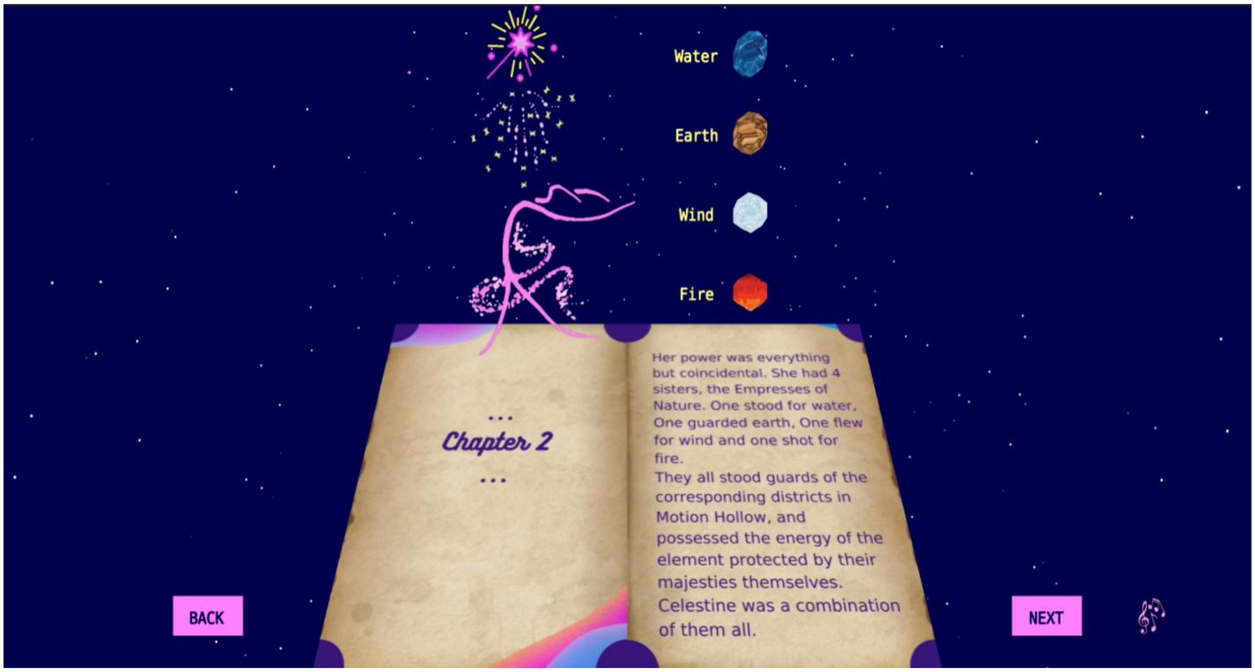
Once upon a time, in a town called Motion Hollow, there was a girl called Celestine.
Celestine had a gift.
No one could move like her.
When she was happy, she would glide like a fairy.
When she was angry, tornados would be jealous of her spins.
When she was sad, her teardrops would dance upwards reaching for the clouds to become the raindrops' understudy.

**Figure 35: The Motion Hollow experience - Book, Chapter 1**

The user has now opened the book and is presented with the story's first chapter (Figure 35). An audio recorded narration of the chapter is now playing, however the user has the option to read the chapter from the book at their pace if they want to. The script used for the first chapter is the following:

*“Once upon a time in a town called Motion Hollow, there was a girl called Celestine.*
*Celestine had a gift. No one could move like her.*
*When she was happy, she would glide like a fairy.*
*When she was angry, tornados would be jealous of her spins.*
*When she was sad, her teardrops would dance upwards reaching for the clouds to become the raindrops’ understudy.”*

If the user clicks on the NEXT button the page will be flipped and the audio narration will stop playing. If the user clicks on the BACK button the book will be closed. Now let’s say the user clicks on the NEXT button.



...
Chapter 2
...

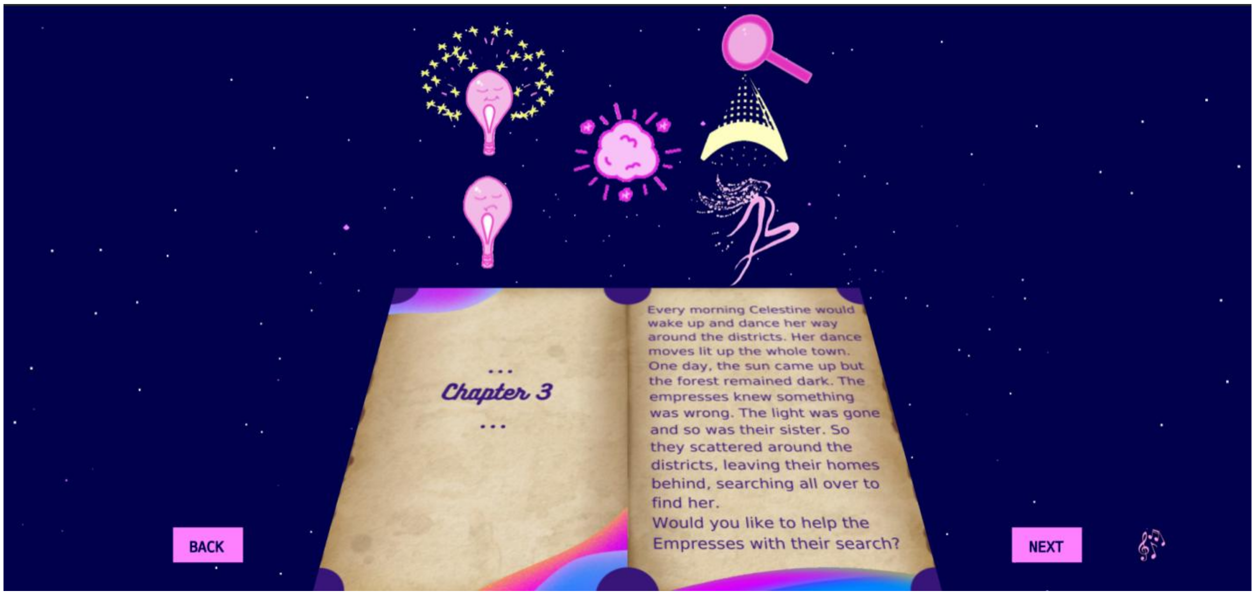
Her power was everything but coincidental. She had 4 sisters, the Empresses of Nature. One stood for water, One guarded earth, One flew for wind and one shot for fire.
They all stood guards of the corresponding districts in Motion Hollow, and possessed the energy of the element protected by their majesties themselves.
Celestine was a combination of them all.

**Figure 36: The Motion Hollow experience - Book, Chapter 2**

The user has now flipped a page and is presented with the story’s second chapter (Figure 36). An audio recorded narration of the chapter is now playing, but the user can read the chapter from the book if they want to. The script used for the second chapter is the following:

*“Her power was everything but coincidental. She had 4 sisters, the Empresses of Nature. One stood for water, One guarded earth, One flew for wind and one shot for fire. They all stood guards of the corresponding districts in Motion Hollow, and possessed the energy of the element protected by their majesties themselves. Celestine was a combination of them all.”*

After listening/reading the chapter, the user now faces two options, either click on the NEXT button and flip forward to the next page and be given a new chapter, or click on the BACK button and revisit Chapter 1. Let’s say the user clicks on the NEXT button.


...
Chapter 3
...

Every morning Celestine would wake up and dance her way around the districts. Her dance moves lit up the whole town. One day, the sun came up but the forest remained dark. The empresses knew something was wrong. The light was gone and so was their sister. So they scattered around the districts, leaving their homes behind, searching all over to find her.
Would you like to help the Empresses with their search?

**Figure 37: The Motion Hollow experience - Book, Chapter 3**

The user has now flipped a page forward again, and is presented with the story's third and final chapter (Figure 37). An audio recorded narration of the chapter is now playing. The script used for the third chapter is the following:
*"Every morning Celestine would wake up and dance her way around the districts. Her dance moves lit up the whole town. One day, the sun came up but the forest remained dark. The empresses knew something was wrong. The light was gone and so was their sister. So they scattered around the districts, leaving their homes behind, searching all over to find her. Would you like to help the Empresses with their search?"*

After listening/reading the chapter, the user once again faces two options, either click on the NEXT button and flip forward to the next page and finish the book, or click on the BACK button and revisit Chapter 2. Let's say the user clicks on the NEXT button.

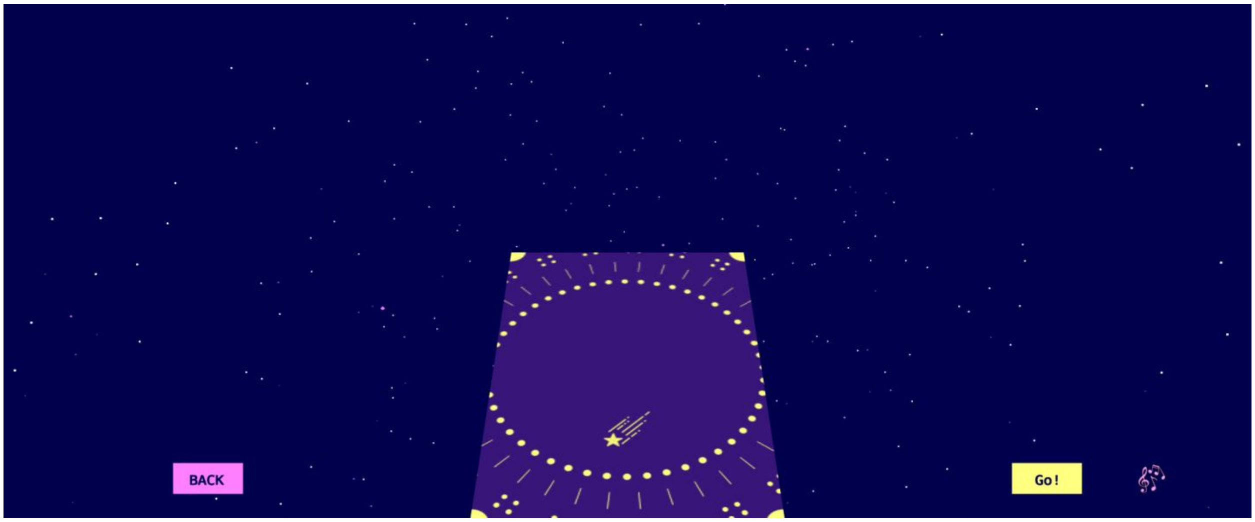


**Figure 38: The Motion Hollow experience - Book, Finished**

The player, having finished the book, now knows the story of the experience and is familiar with their task. As a result, the user clicks on the GO button to move on with the mission (Figure 38).

## 3. Motion Hollow Map

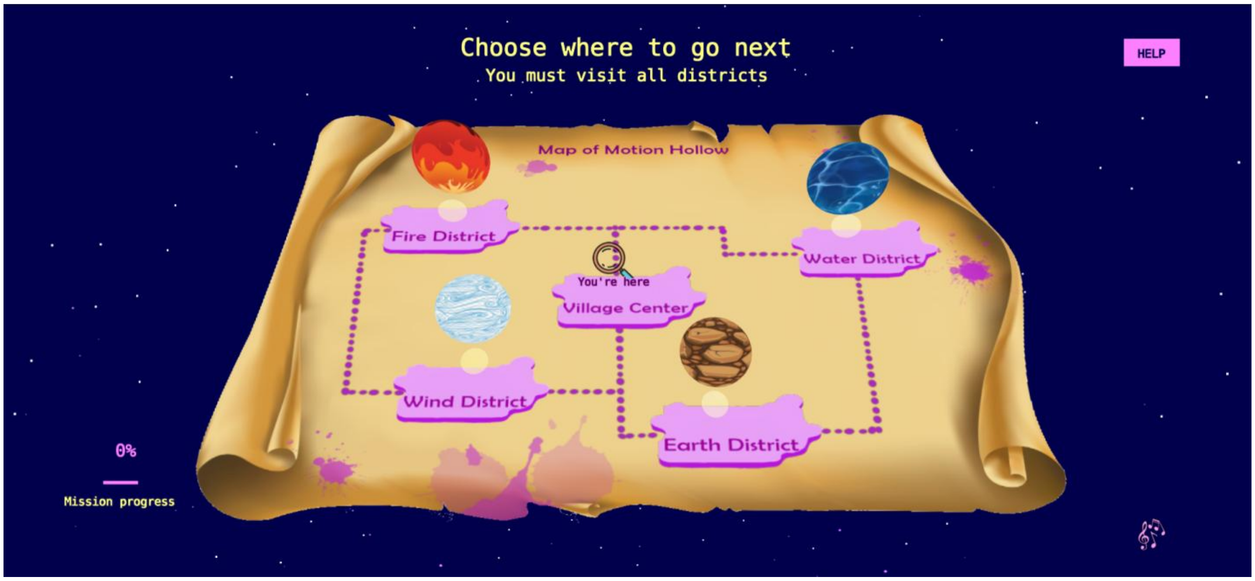
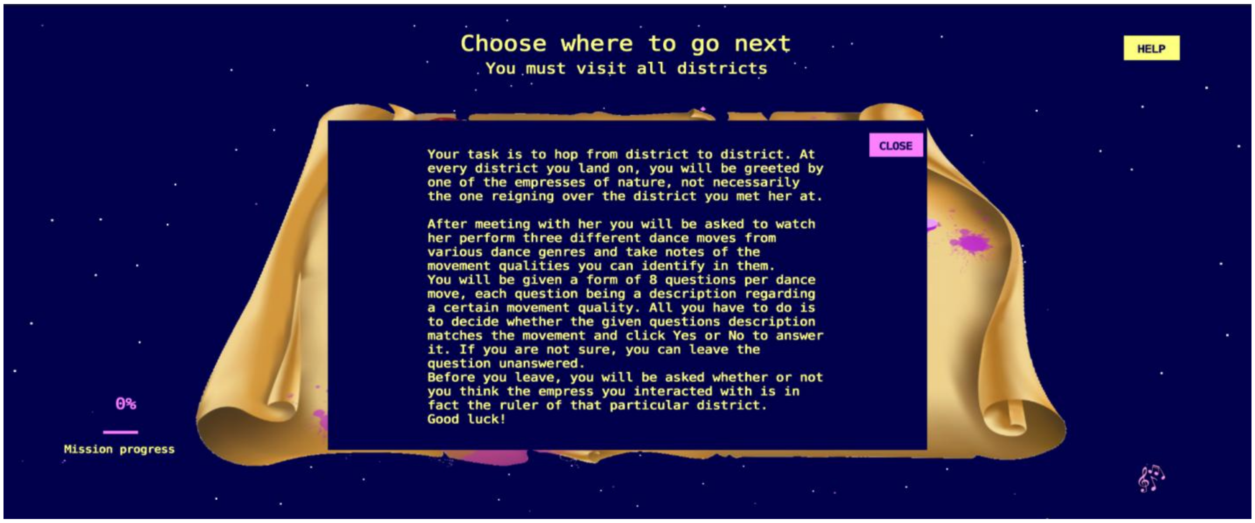


**Figure 39: The Motion Hollow experience – Map**

The user is given a map to Motion Hollow with portal links to all the districts (Figure 39). The user is now listening to an audio narration of instructions on how they have to proceed. On the bottom-left of the screen there is a progress bar which shows what percentage of the mission is already complete. On the top-right corner of the screen there is a HELP button. Let's say the user did not pay attention to the audio narration, and needs to be told what to do once more, so they click on the HELP button.



Your task is to hop from district to district. At every district you land on, you will be greeted by one of the empresses of nature, not necessarily the one reigning over the district you met her at.

After meeting with her you will be asked to watch her perform three different dance moves from various dance genres and take notes of the movement qualities you can identify in them.
You will be given a form of 8 questions per dance move, each question being a description regarding a certain movement quality. All you have to do is to decide whether the given questions description matches the movement and click Yes or No to answer it. If you are not sure, you can leave the question unanswered.
Before you leave, you will be asked whether or not you think the empress you interacted with is in fact the ruler of that particular district.
Good luck!

**Figure 40: The Motion Hollow experience - Elaborate instructions**

A set of elaborate instructions is now visible on the screen along with an audio narration of those instructions playing simultaneously (Figure 40). The script for the elaborate instructions is the following:
*"Your task is to hop from district to district. At every district you land on, you will be greeted by one of the empresses of nature, not necessarily the one reigning over the district you met her at. After meeting with her you will be asked to watch her perform three different dance moves from various dance genres and take notes of the movement qualities you can identify in them. You will be given a form of 8 questions per dance move, each question being a description regarding a certain movement quality. All you have to do is to decide whether the given question's description matches the movement and click Yes or No to answer it. If you are not sure, you can leave the question unanswered. Before you leave, you will be asked whether or not you think the empress you interacted with is in fact the ruler of that particular district. Good luck!"*

The user now reads what they have to do and understands their task eventually. So they click on the CLOSE button on the instruction window and clicks on the Fire District Sphere on the map.

## 4. Travelling to a District

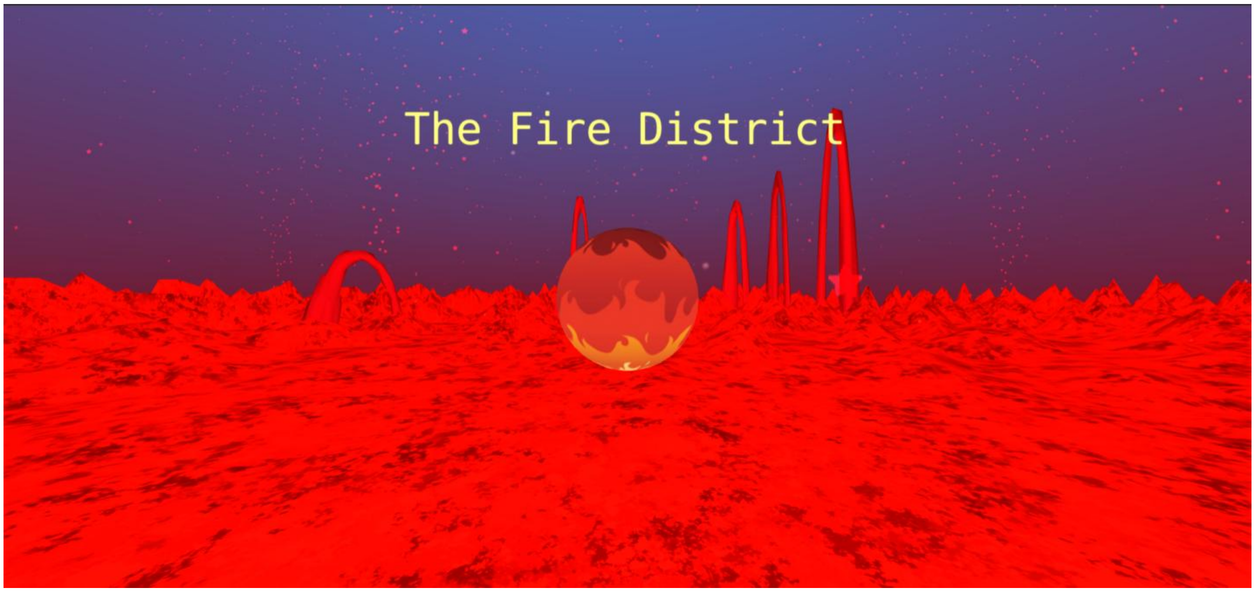
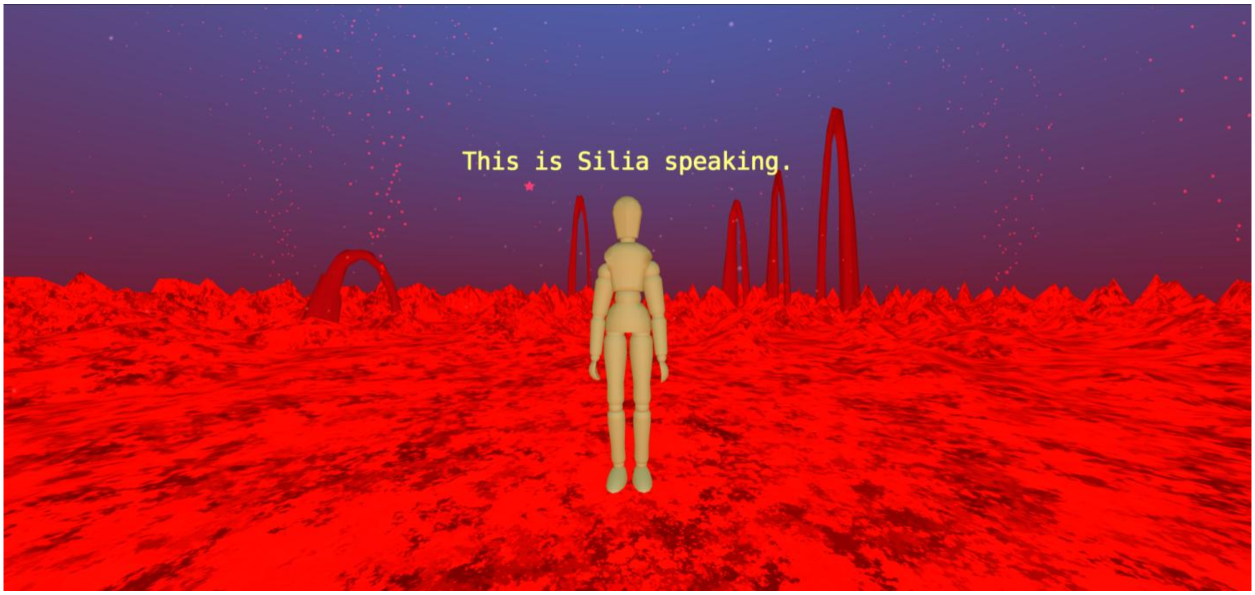


**Figure 41: The Motion Hollow experience - The Fire District, Arrival**

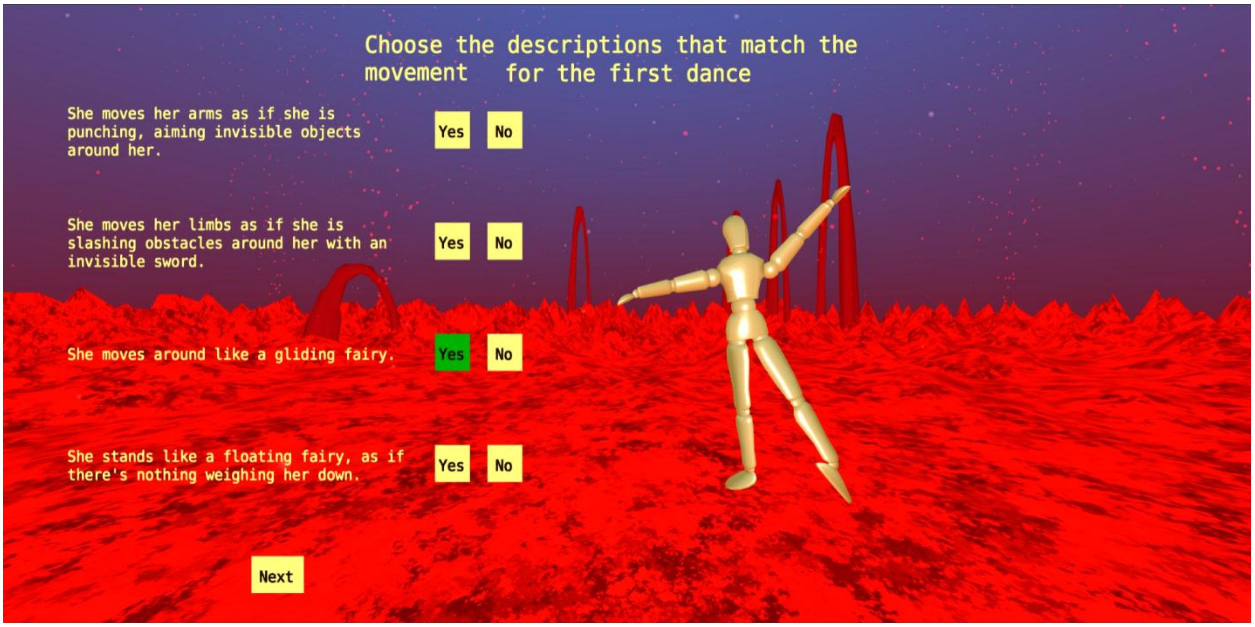
The user has been teleported to the Fire District (Figure 41). After taking a look of the environment they are surrounded by, they have to find one of the Empresses in order to proceed to the annotation process and move on to the rest of the districts. In order to avoid repeating every district-related action four times during this walkthrough, we will only be presenting the annotation process once, showing one dance move in the Fire District. However, a visual presentation of the other district environments will be provided later on.



This is Silia speaking.

**Figure 42: The Motion Hollow experience - The Fire District, Meeting the Empress**

The user has now met the empress and is ready to start annotating movement qualities on the first dance move (Figure 42).



Choose the descriptions that match the movement for the first dance

She moves her arms as if she is punching, aiming invisible objects around her.

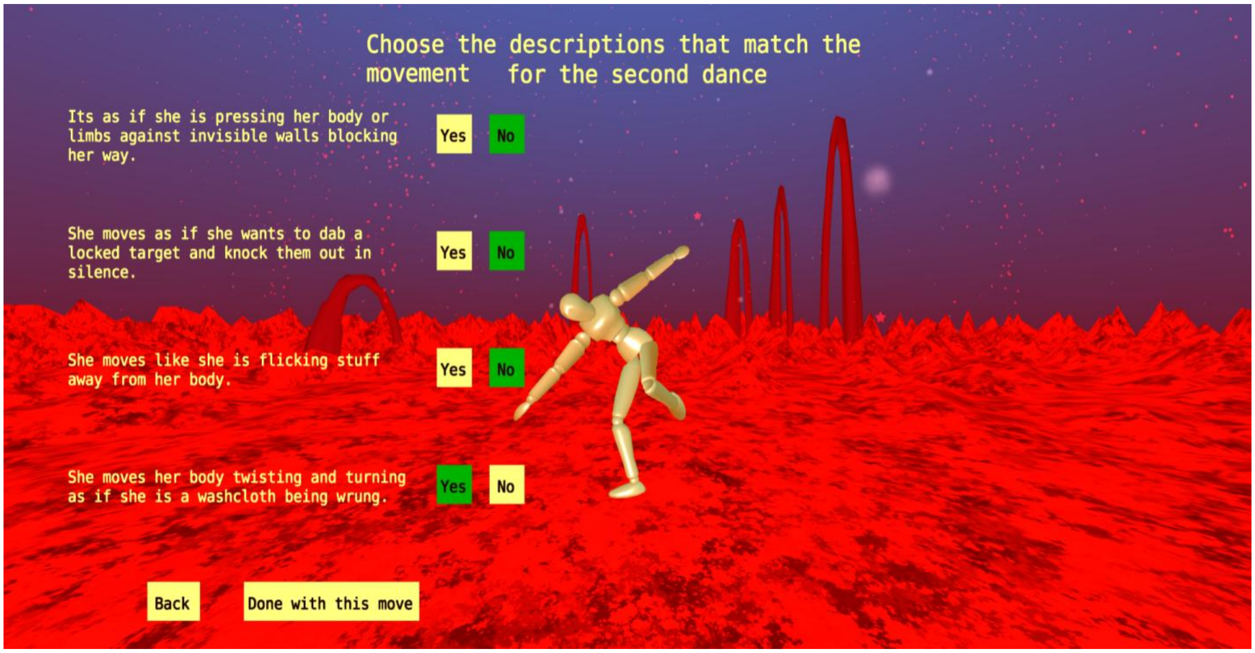
She moves her limbs as if she is slashing obstacles around her with an invisible sword.

She moves around like a gliding fairy.

She stands like a floating fairy, as if there's nothing weighing her down.

**Figure 43: The Motion Hollow experience - The Fire District, Annotation process, Set #1 LMA Efforts used in this set of descriptions: punch, slash, glide, float**

The annotation process has started. The 3D model representing the empress starts performing the first dance move and the user is given the first set of descriptive clauses associated with the LMA efforts. The first set includes **punching**, **slashing**, **gliding** and **floating** (Figure 43). The user clicks YES for the ones they identify in the movement and NO for the ones that cannot be identified. The user has the option to not interact with a description if they are not sure about whether or not the respective quality exists in the movement. When the user is done annotating, they click the NEXT button to get the second set of descriptions.



Choose the descriptions that match the movement for the second dance

Its as if she is pressing her body or limbs against invisible walls blocking her way.

She moves as if she wants to dab a locked target and knock them out in silence.

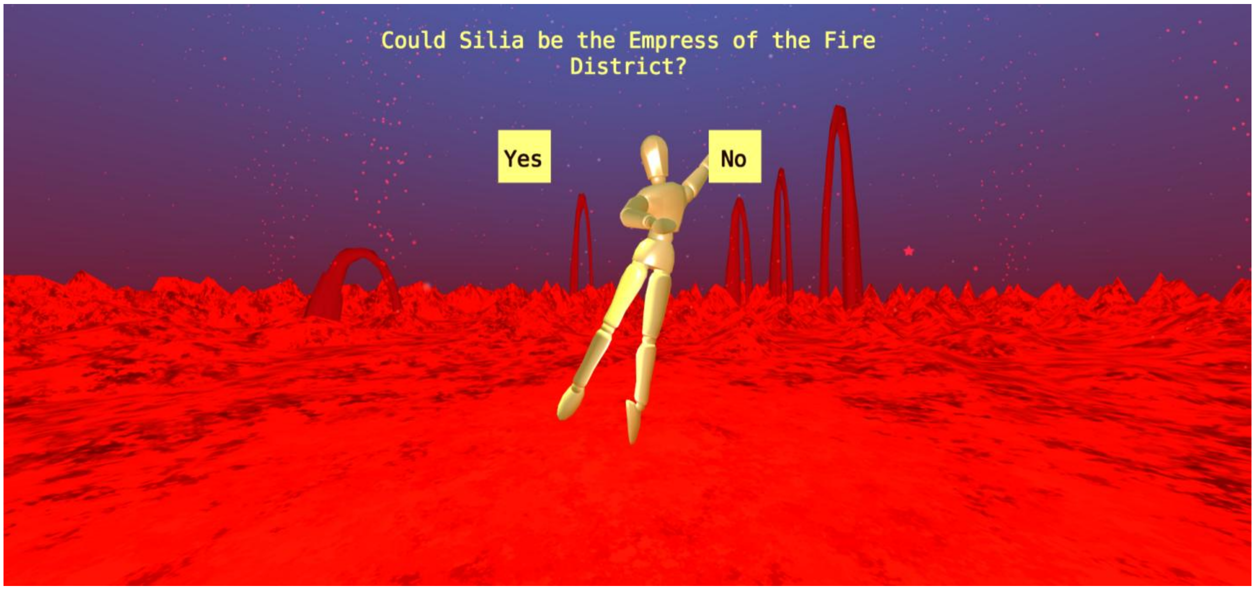
She moves like she is flicking stuff away from her body.

She moves her body twisting and turning as if she is a washcloth being wrung.

**Figure 44: The Motion Hollow experience - The Fire District, Annotation process, Set #2 LMA Efforts used in this set of descriptions: press, dab, flick, wring**

The annotation process has continues with the second set of descriptions. This set includes **pressing**, **dabbing**, **flicking** and **wringing** (Figure 44). The user clicks YES for the ones they identify in the movement and NO for the ones that cannot be identified. The user has the option to not interact with a description if they are not sure about whether or not the respective quality exists in the movement. The user has the option to revisit the first set of descriptions by clicking on the BACK button, reevaluate their answers and make changes before submitting them and moving on to the next dance move. When the user is done with the current move, they click the DONE WITH THIS MOVE button to get to the next dance move.

After the user is done annotating all three dance of the current district, they are now ready to return to the Motion Hollow map and travel to a different district. However before leaving the Fire District the user has to answer one more question: Whether they think the empress they met with is the ruler of the Fire District or not. The user has to base their decision on the way the empress moved during the performance of the moves, but they also have to greatly rely on their intuition.

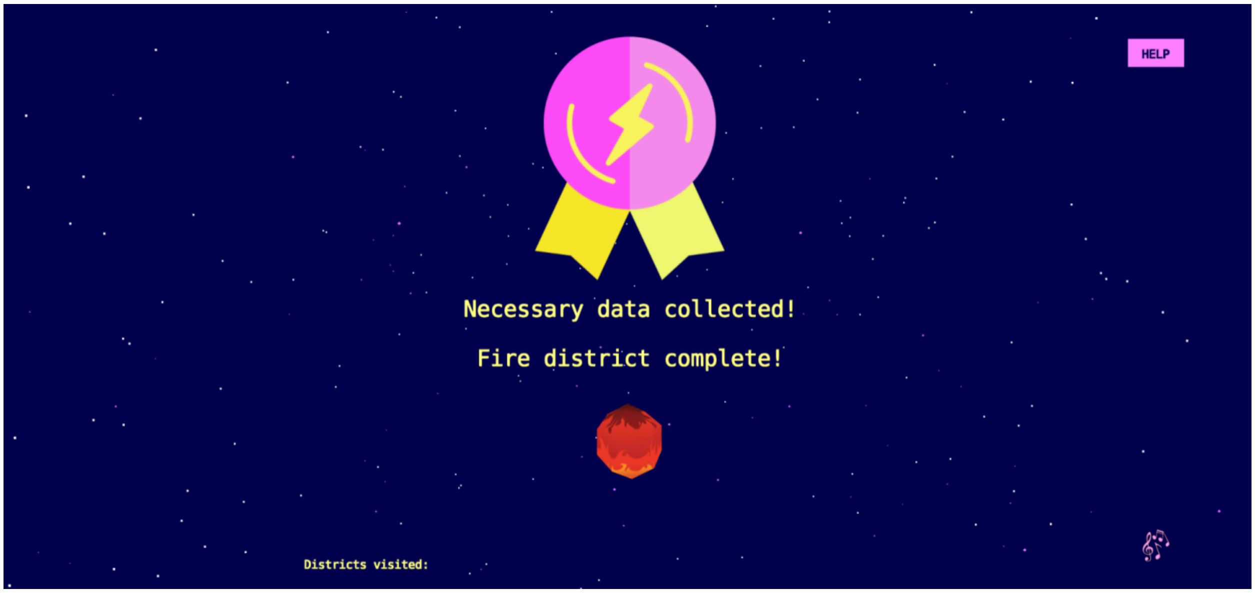


Could Silia be the Empress of the Fire District?

**Figure 45: The Motion Hollow experience - The Fire District, Empress Question**

The user is prompted with aforementioned question (Figure 45) and, after giving it much consideration, they decide that the empress cannot be the empress of the Fire District. So they click on the NO button and leave the Fire District.

## 5. Returning to the Motion Hollow Map



Necessary data collected!

Fire district complete!

**Figure 46: The Motion Hollow experience - The Fire District, Completion Badge**

Before returning to the map the user is awarded with a completion badge for contributing data by annotating movement qualities in the Fire District. The badge is added to the user's dashboard at the bottom-center of the page (Figure 46).

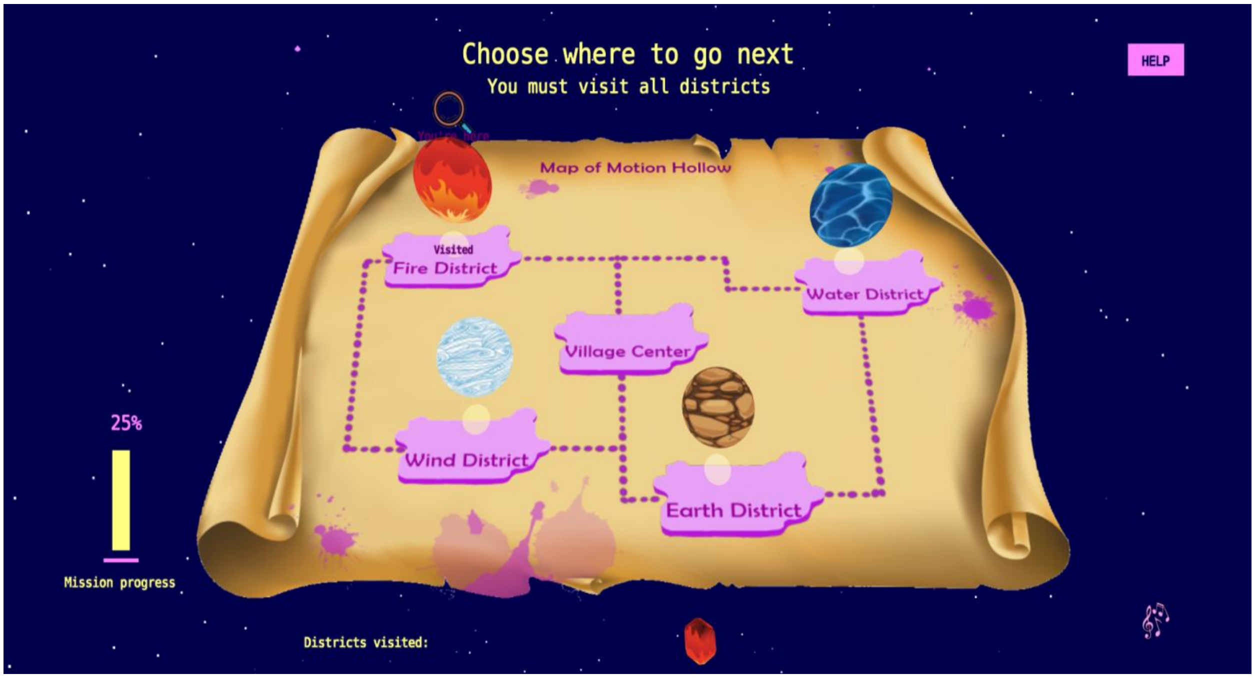


**Figure 47: The Motion Hollow experience – Map, after returning from the Fire District (1st district visited)**

The user has now returned to the map (Figure 47). The Fire District completion badge has been added to their dashboard, and the progress bar has been increased, marking the percentage of the mission's completion to 25%. The user's pawn is at the Fire District and they have to choose where they want to go next. After deciding, the user clicks on the district sphere and is teleported to the district.

## 6. Visiting the rest of the Districts

The user has to repeat steps 4 and 5 for the Earth, Water and Wind Districts as well. However, since these steps are identical for all districts, we will not be showing them again in this walkthrough. Instead, we will only be presenting the districts as environments to provide a visual. Two screenshots will be presented for each district, one for the first screen the user sees after being teleported and one for the first interaction they have with the empress. Figures 48 to 53 provide screenshots of the Earth, Wind and Water district environments.

## The Earth District

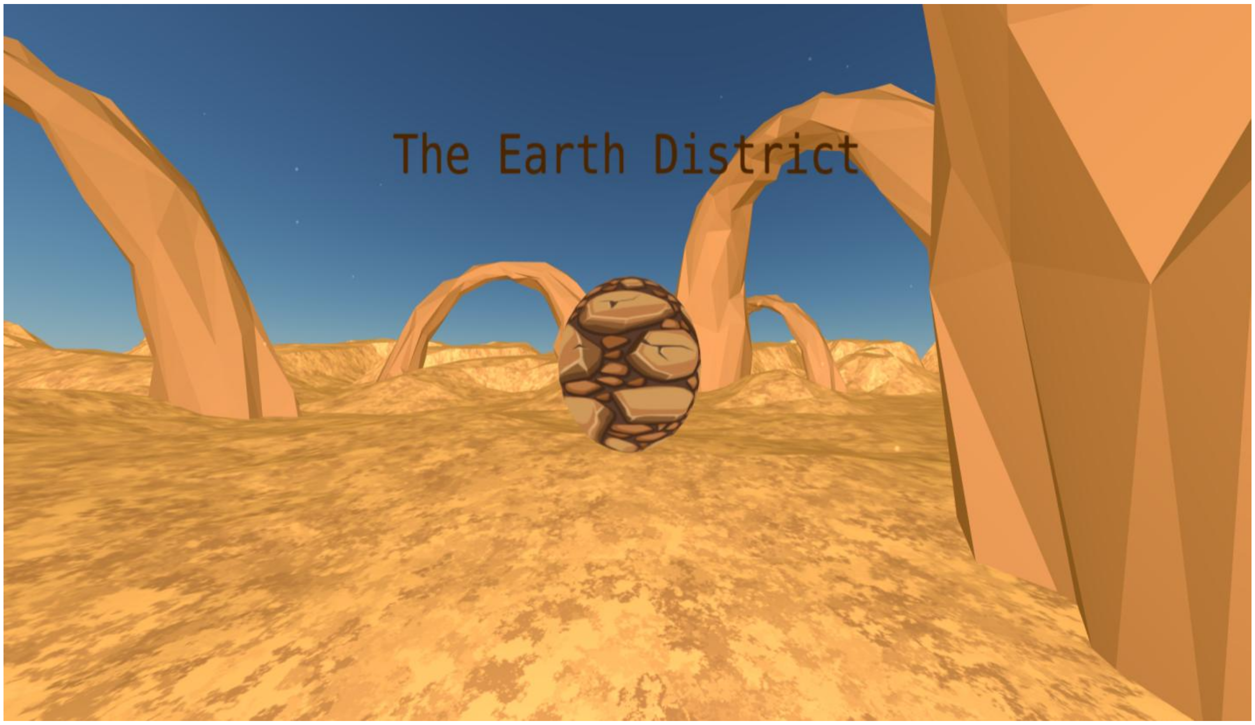


**Figure 48: The Motion Hollow experience - The Earth District, Arrival**

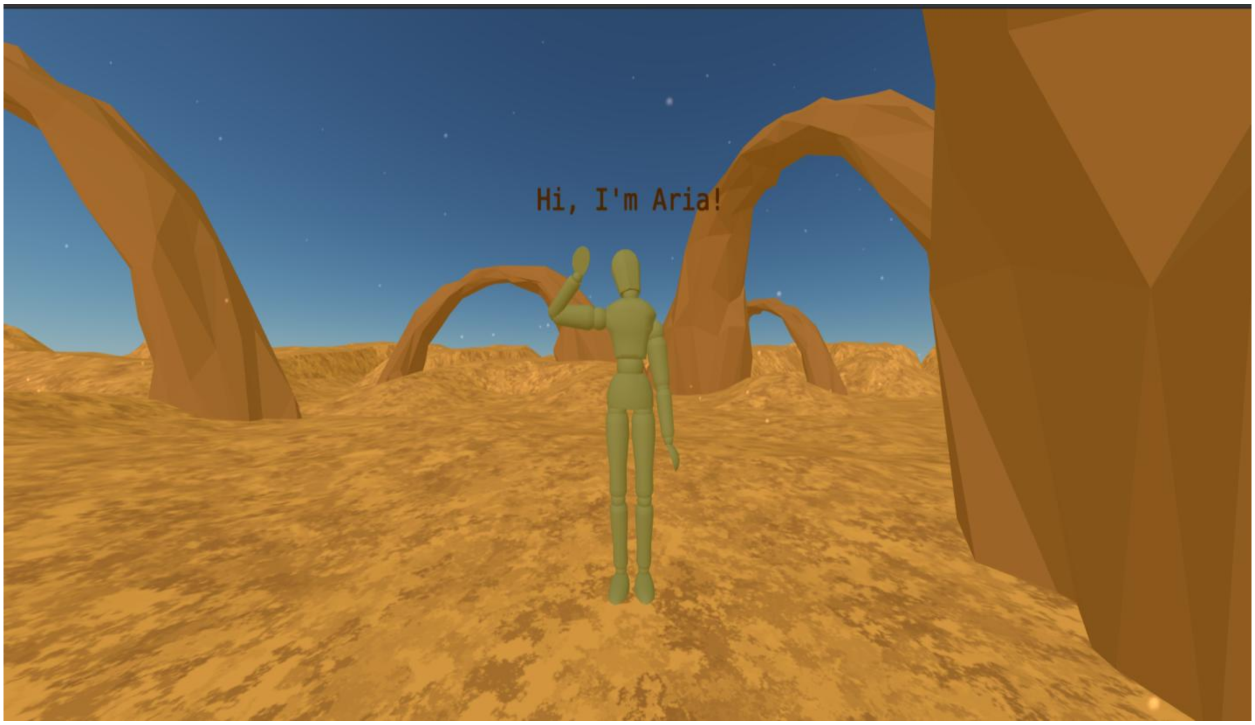


**Figure 49: The Motion Hollow experience - The Earth District, Empress greeting**

## The Wind District

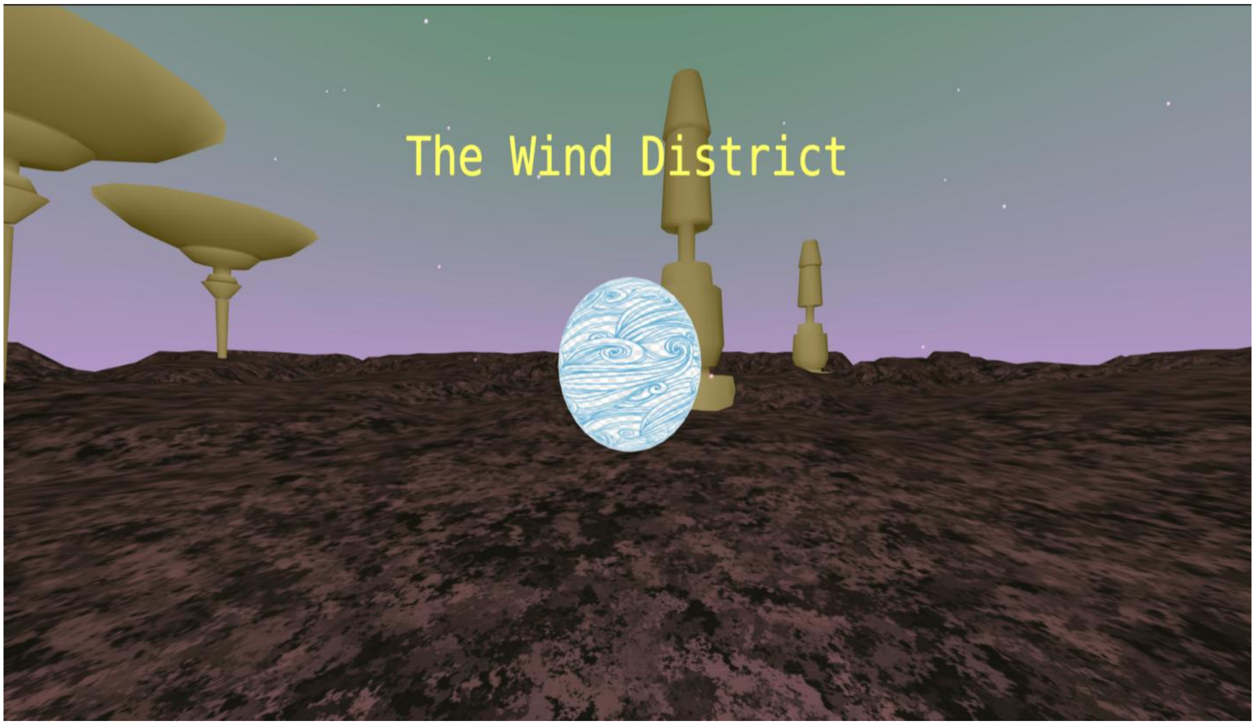


**Figure 50: The Motion Hollow experience - The Wind District, Arrival**

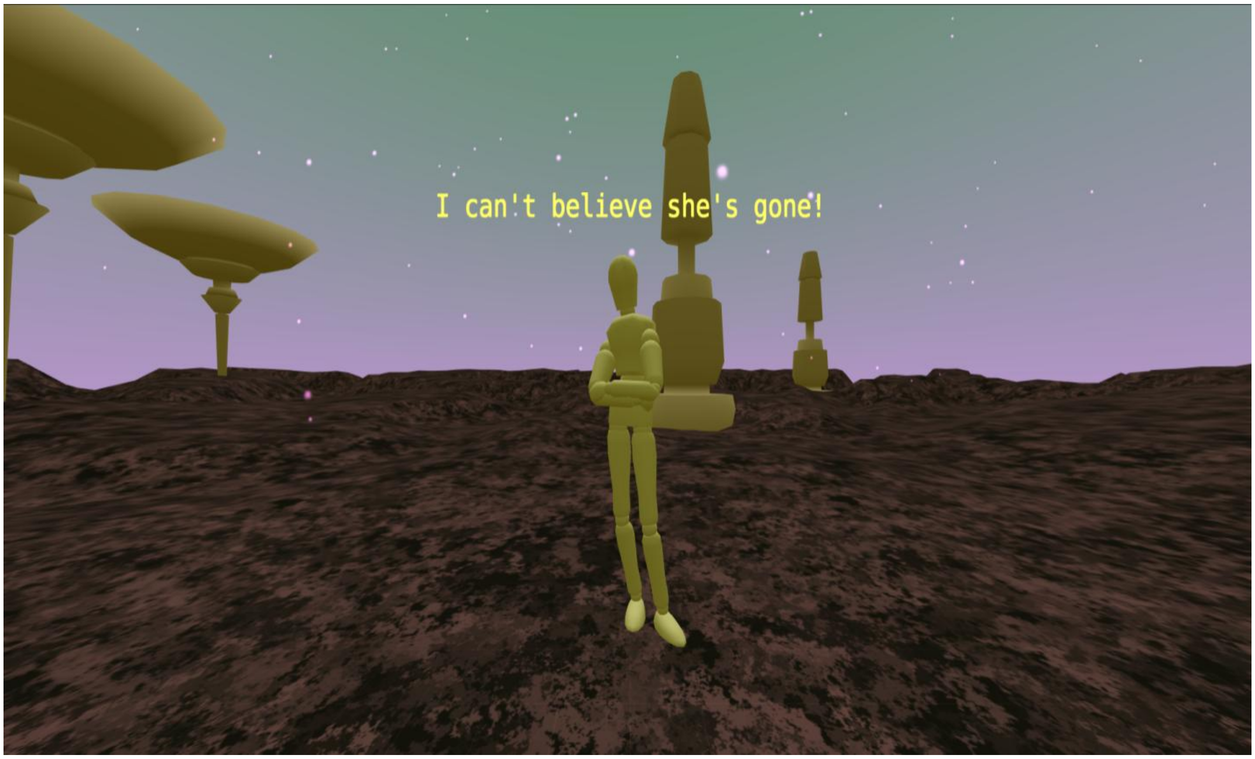


**Figure 51: The Motion Hollow experience - The Wind District, Empress greeting**

### The Water District

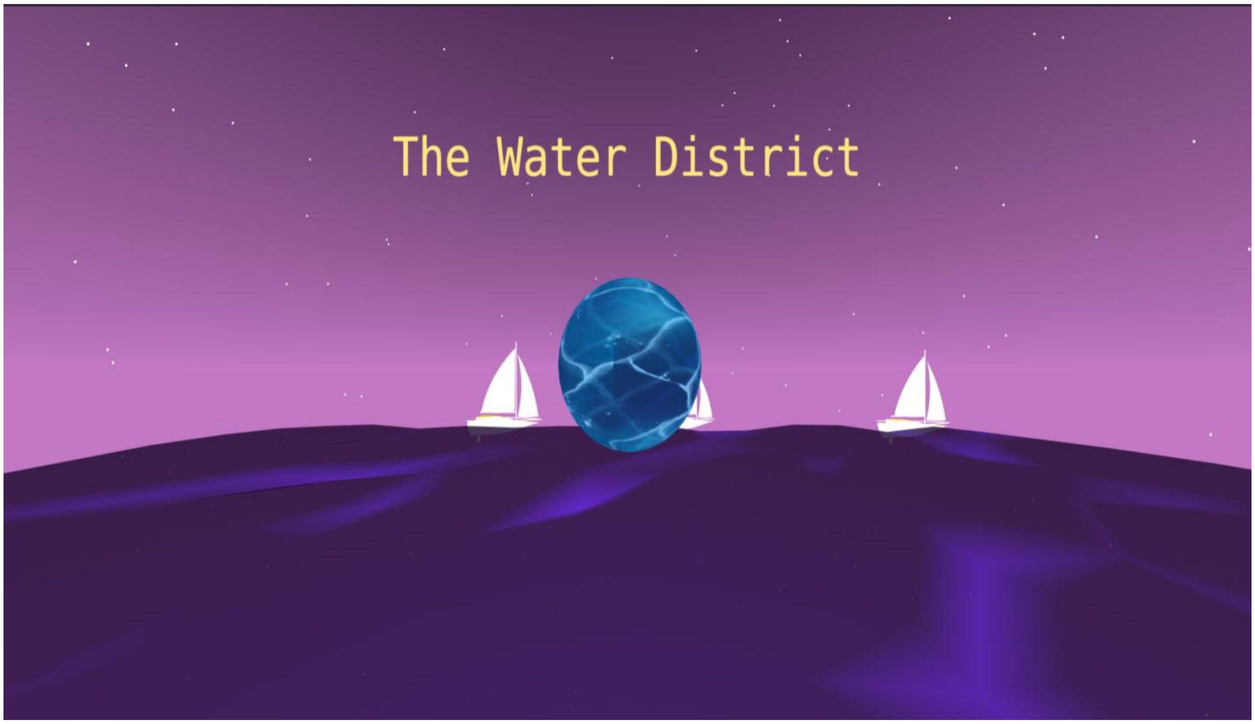


**Figure 52: The Motion Hollow experience - The Water District, Arrival**

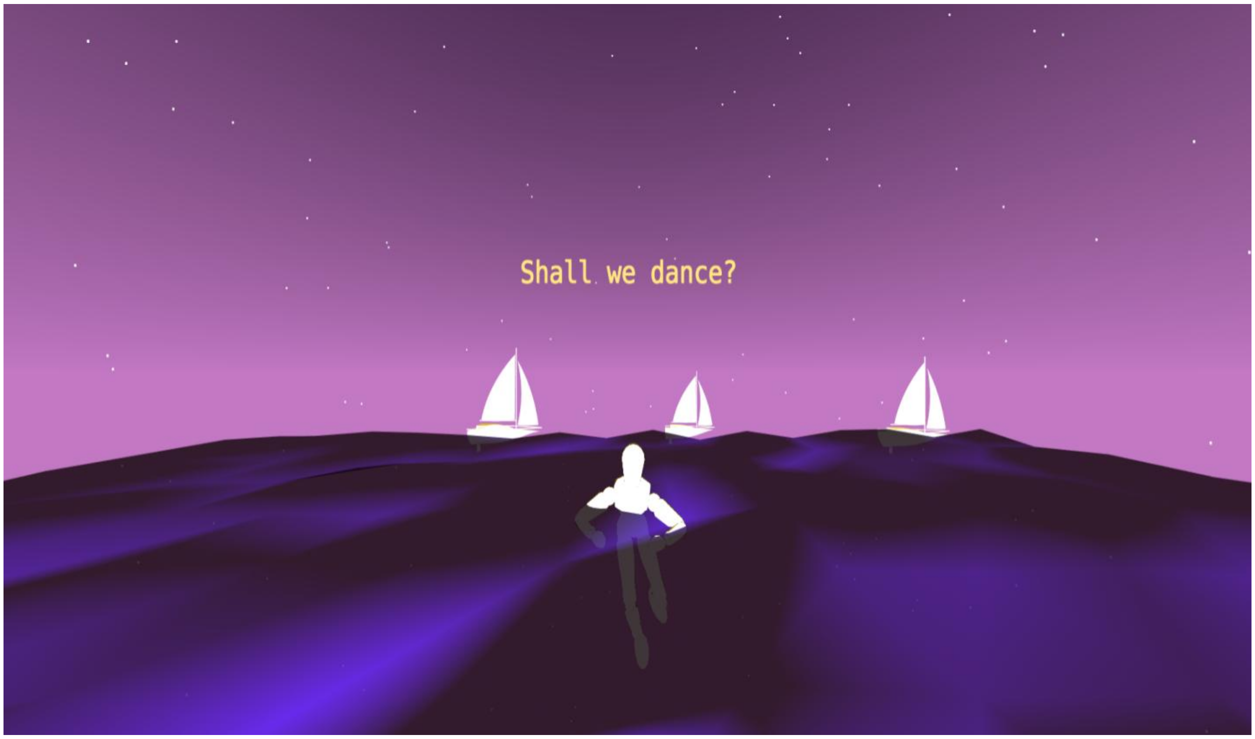


**Figure 53: The Motion Hollow experience - The Water District, Empress greeting**

## 7. Completing the Journey

After completing the task required in the last district, the user does not return to the map. The last completion badge is added to the user's dashboard (Figure 54) and the user is immediately navigated to the last part of the experience, the revelation of the story finale by all four empresses (Figure 55), which will not be revealed in order not to spoil the experience should anyone want to participate in the future.

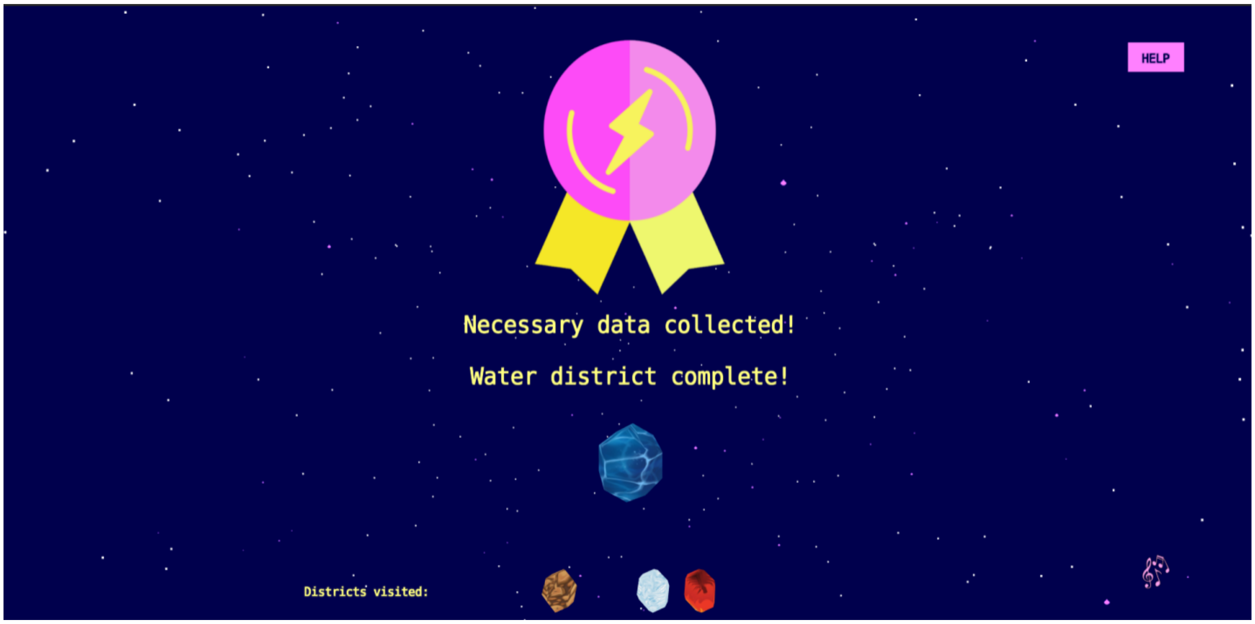


**Figure 54: The Motion Hollow experience - Mission complete**

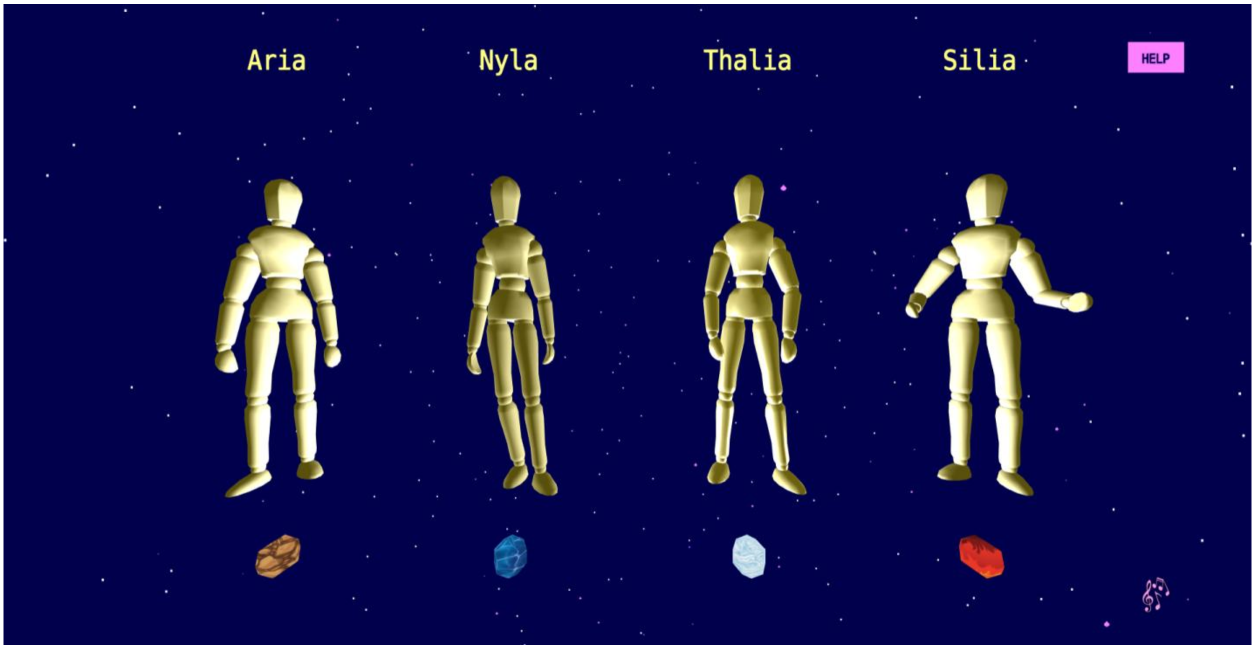
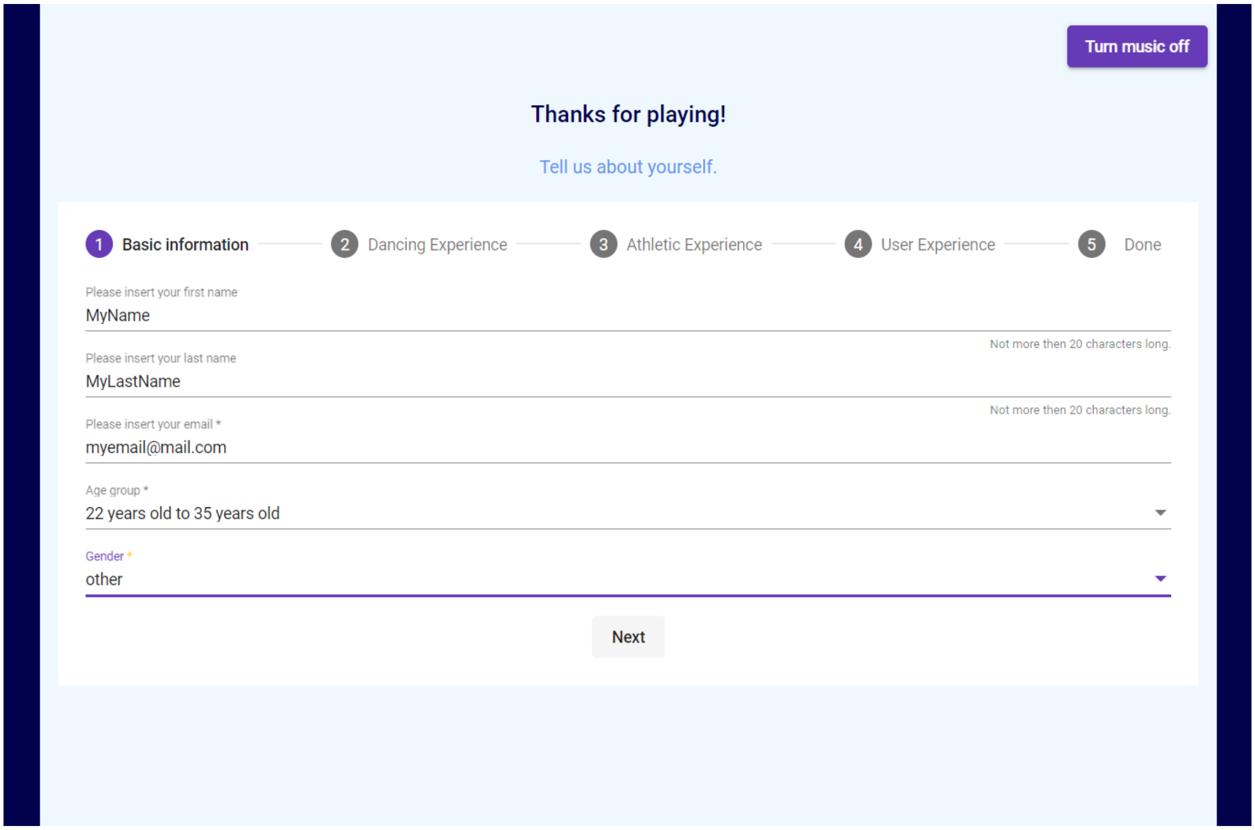


**Figure 55: The Motion Hollow experience - Story finale**

## 8. Filling out the Profiling Form

The user is now given a profiling form which they are required to fill out with their personal information, information about their dancing and athletic experience, should they have any. Additionally, they are asked to answer some questions regarding the user experience. This profiling form is experimental, and its content is bound to change after receiving some feedback from the evaluation of the experience.

### Personal Info



Thanks for playing!

Tell us about yourself.

Please insert your first name
MyName
Not more then 20 characters long.

Please insert your last name
MyLastName
Not more then 20 characters long.

Please insert your email *
myemail@mail.com

Age group *
22 years old to 35 years old

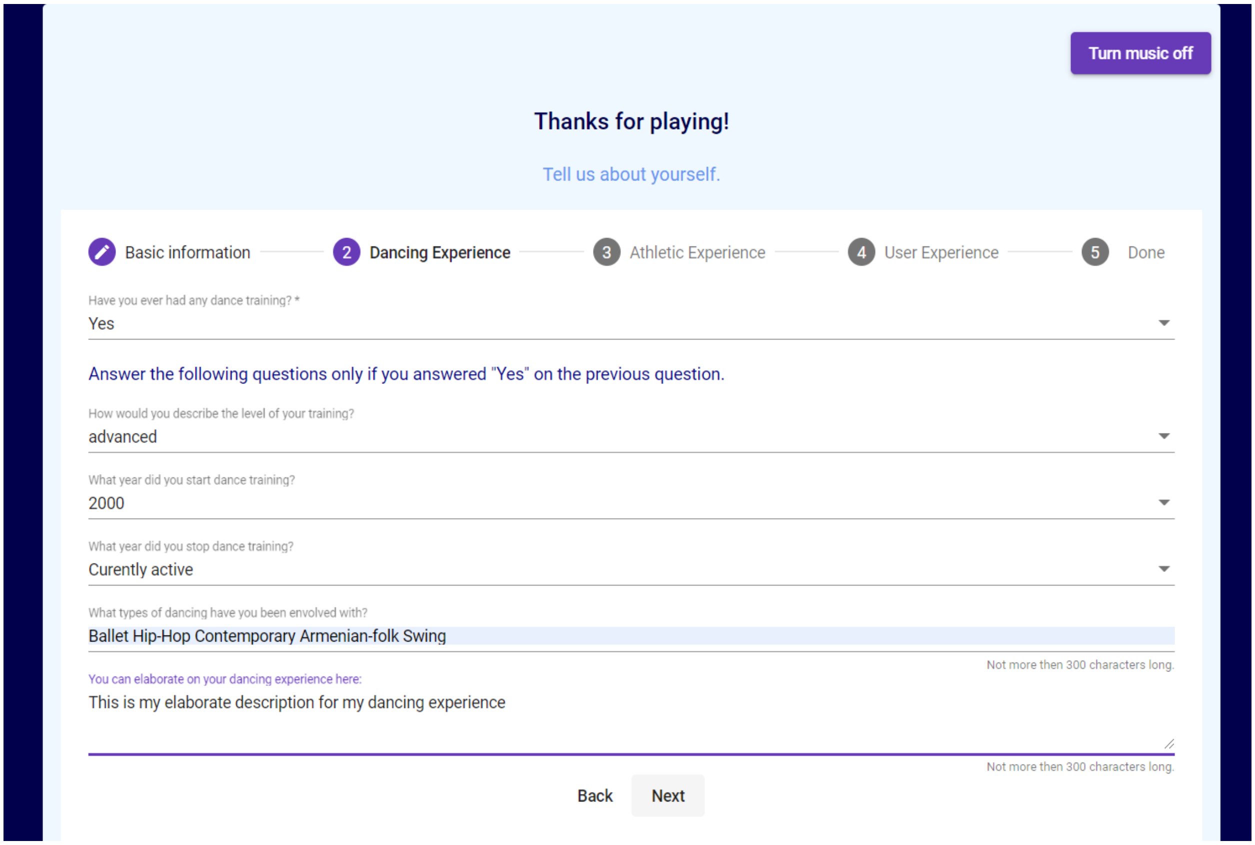
Gender *
other

**Figure 56: The Motion Hollow experience - Profiling form, personal info**

This step of the profiling form asks for the user's basic personal information (Figure 56). The fields requiring the first name and the last name are optional, since the user may not want to share this information. On the other hand, the fields regarding the age and the gender are required, since they are meaningful pieces of information when it comes to classifying a user as part of a certain user group. The field asking the user's email address is also required, since it acts both as a unique identifier for the submitted form data and as a way to contact the user if necessary at any point after the submission.

So, after the user has filled in the fields of this step, they click on the NEXT button to move to the next step of the profiling process.

## Dancing Experience



Thanks for playing!

Tell us about yourself.

Have you ever had any dance training? *
Yes

Answer the following questions only if you answered "Yes" on the previous question.

How would you describe the level of your training?
advanced

What year did you start dance training?
2000

What year did you stop dance training?
Curently active

What types of dancing have you been envolved with?
Ballet Hip-Hop Contemporary Armenian-folk Swing

Not more then 300 characters long.

You can elaborate on your dancing experience here:
This is my elaborate description for my dancing experience

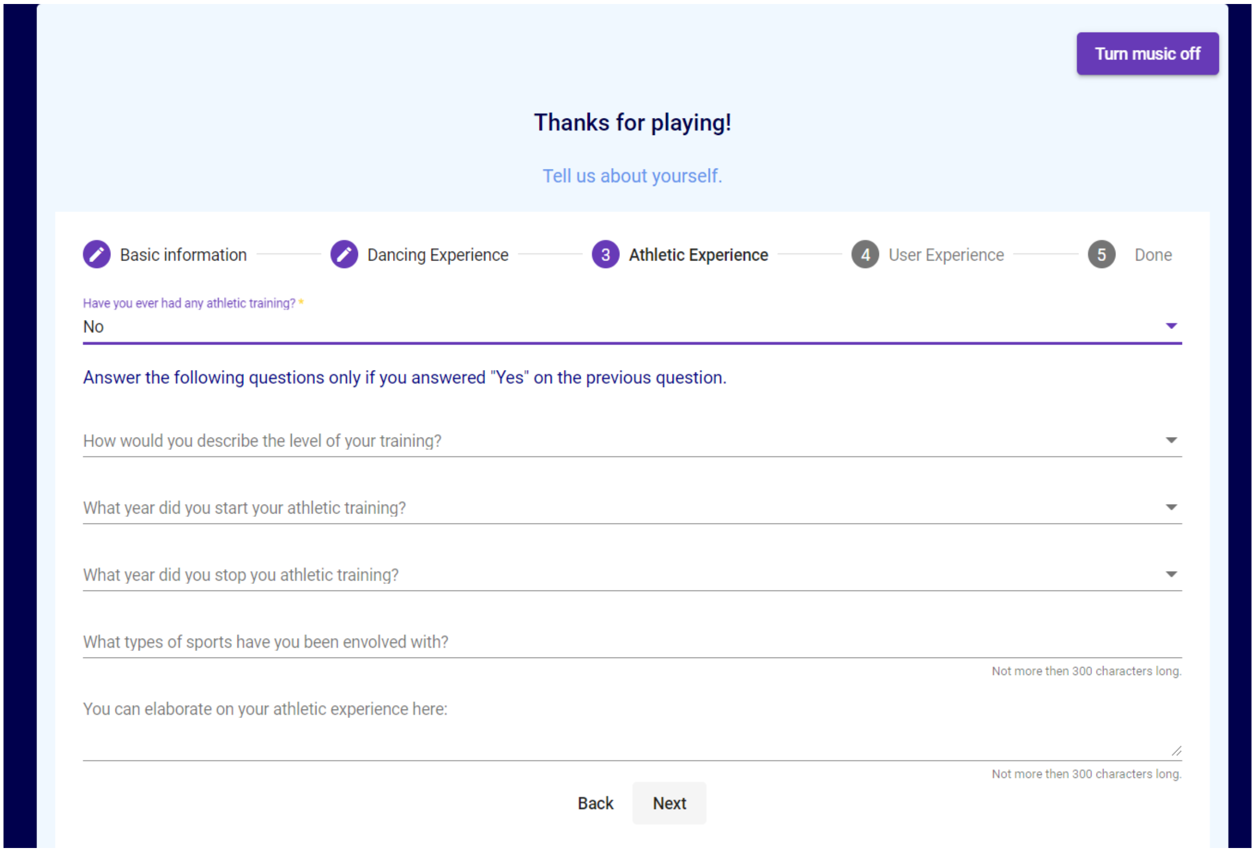
Not more then 300 characters long.

**Figure 57: The Motion Hollow experience - Profiling form, dancing experience**

In this step, the form asks the user to fill out information regarding their dancing experience (Figure 57). First the user has to answer whether they have been involved with dance at any point in their lives or not. If the answer is positive, they have to answer the following questions as well. Specifically, the user is asked to fill in their level of dance training, and their active training years. They are also required to fill in the types of dancing they have ever taken up. If the user feels like the information they provided is inadequate for the study, a text input field is provided for them to elaborately describe their experience associated with dance. This might be information about the dance education they have received, competitions or recitals they have participated in, dance groups they might have joined etc.

After the user has answered the questions, they click on NEXT to move on to the next step. The user also has the option to revisit a previously completed step to revise their answers and make any alterations if they want to.

## Athletic Experience



Turn music off

Thanks for playing!

Tell us about yourself.

Have you ever had any athletic training? *
No

Answer the following questions only if you answered "Yes" on the previous question.

How would you describe the level of your training?

What year did you start your athletic training?

What year did you stop you athletic training?

What types of sports have you been envolved with?

Not more then 300 characters long.

You can elaborate on your athletic experience here:

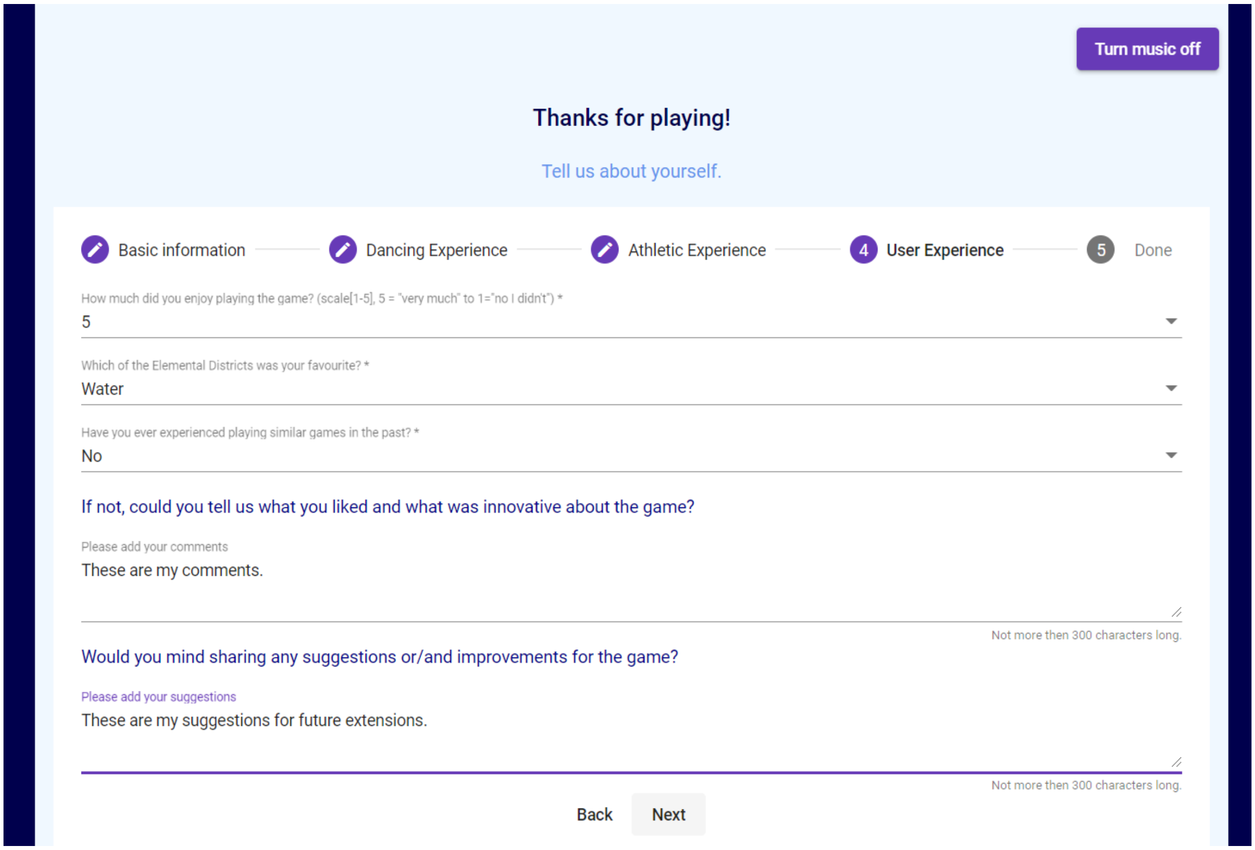
Not more then 300 characters long.

**Figure 58: The Motion Hollow experience - Profiling form, athletic experience**

In this step, the form asks the user to fill out information regarding their athletic experience (Figure 58). First they have to answer whether they have been involved with any type of sports at any point in their lives or not. If the answer is positive, they have to answer the following questions as well. Specifically, the user is asked to fill in their level of athletic training, and their active training years. They are also required to fill in the types of sports they have ever taken up in the respective field. If the user feels like the information they provided is inadequate for the study, a text input field is provided for them to elaborately describe their athletic experience. This information might refer to the different periods they were involved with different sports, championships they might have participated in or won, medals they have been awarded with etc.

After the user has completed this step, they click on NEXT to move on to the next step. The user also has the option to revisit all previously completed steps to revise their answers and make any changes if they want to.

## User Experience



Thanks for playing!

Tell us about yourself.

How much did you enjoy playing the game? (scale[1-5], 5 = "very much" to 1="no I didn't") *
5

Which of the Elemental Districts was your favourite? *
Water

Have you ever experienced playing similar games in the past? *
No

If not, could you tell us what you liked and what was innovative about the game?

Please add your comments
These are my comments.

Not more then 300 characters long.

Would you mind sharing any suggestions or/and improvements for the game?

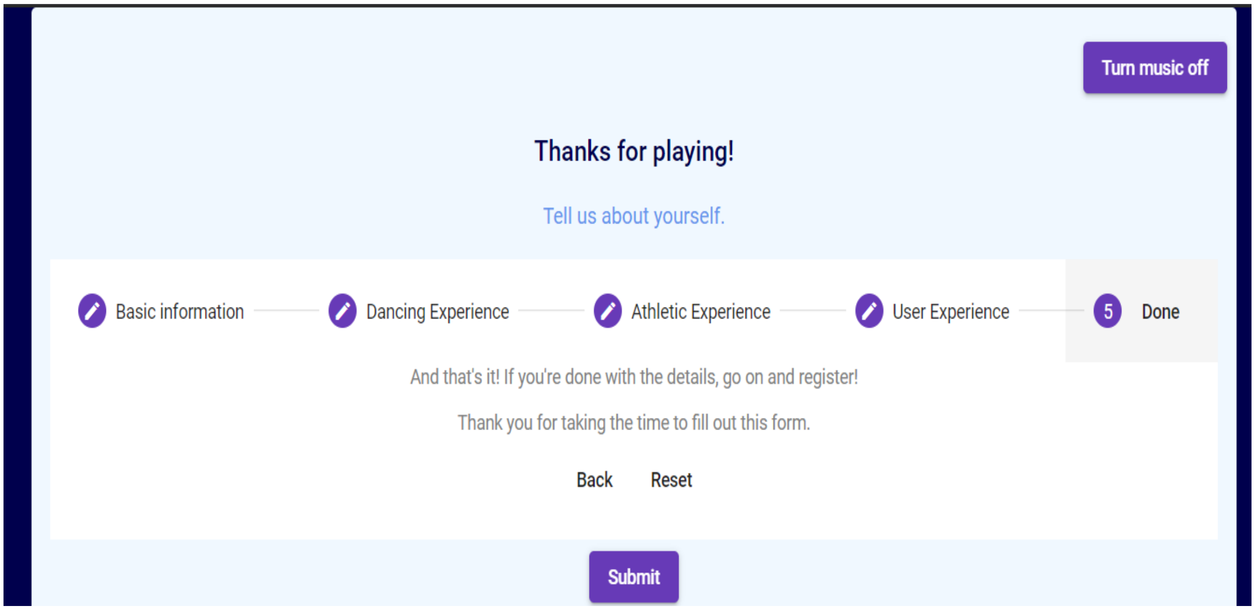
Please add your suggestions
These are my suggestions for future extensions.

Not more then 300 characters long.

**Figure 59: The Motion Hollow experience - Profiling form, user experience**

In this step the user is required to answer some questions regarding the usability of the experience (Figure 59). They are also required to answer some questions regarding the extent to which they enjoyed participating in it. Specifically, the user is required to rate the experience on a scale from 1 to 5, based on how much they enjoyed participating in it. They also have to fill in their favorite elemental district, and answer whether or not they have ever participated in a similar experience before. Last but not least, the form provides two optional text input fields, one for the user to write their positive comments regarding the experience and one for suggestions about parts that could have been executed differently, or recommendations about future extensions.

After the user has completed this step, they click on NEXT to be able to submit the form. The user also has the option to revisit all previously completed steps to revise their answers and make any changes if they want to. So after clicking NEXT the user is navigated to the following and final step and the SUBMIT button shows up on the page. In this step the user is not required to do anything, they are just informed that the profiling session is complete and that if they feel okay with their answers they can submit the form to the server. On the other hand, if they want to start over, they can click on the RESET button (Figure 60).



Thanks for playing!

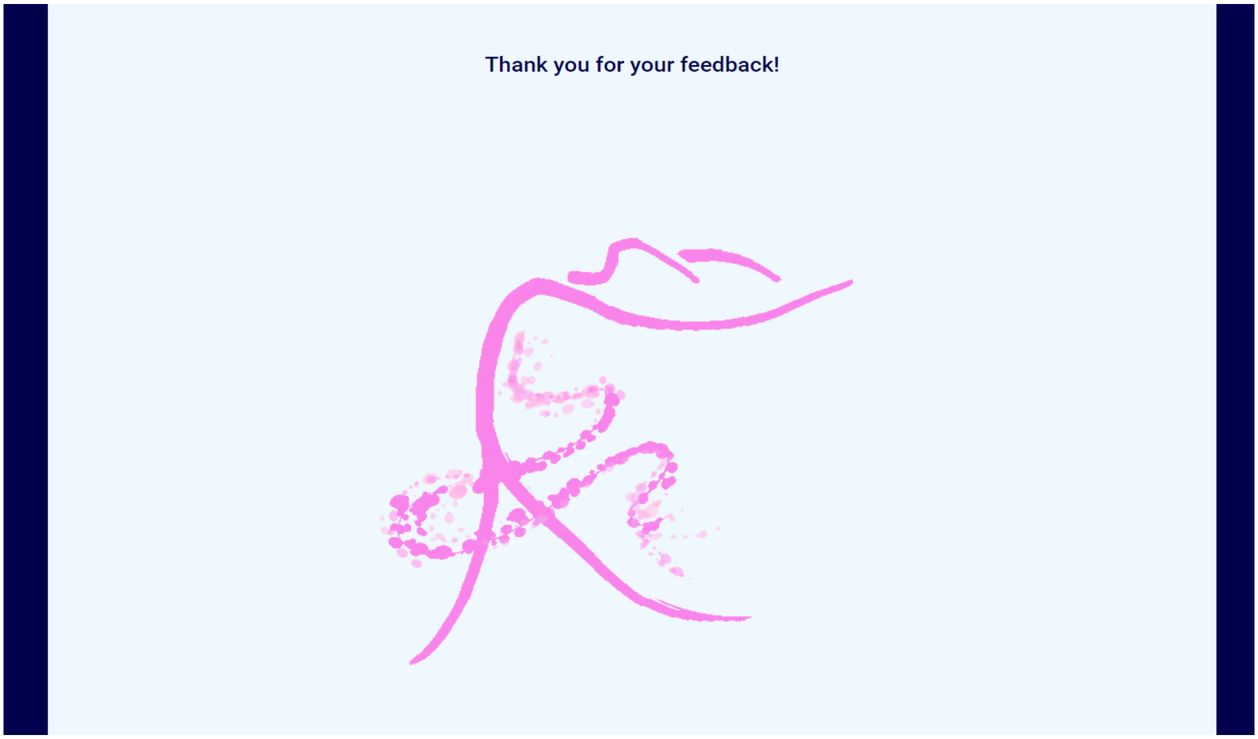
Tell us about yourself.

And that's it! If you're done with the details, go on and register!

Thank you for taking the time to fill out this form.

**Figure 60: Motion Hollow experience - Profiling form, completed**

After submitting the form, the user is presented with a “Thank you” screen (Figure 61).



Thank you for your feedback!

**Figure 61: The Motion Hollow experience - Thank you screen**

And this is pretty much it. The Motion Hollow experience walkthrough is complete with the rendering of this screen (Figure 61).

## 4.5 Evaluation and Feedback

### 4.5.1 Introduction

This section presents selected comments, suggestions and overall feedback derived from the formative evaluation of the Motion Hollow experience. The different aspects examined during the evaluation are the following:

- Usability of the application
- Aesthetics and “look and feel” of the application
- The content of the application

### 4.5.2 Basic Information about the Evaluation

The evaluation was carried out with 8 participants, 7 of them were women and one man. The participants were within the ages of 24 to 51. The method used was to observe the participants while using the application, either in a physically co-located face-to-face session or remotely. Only one of the evaluations was carried out in person using the researcher’s computer. For the rest of them, the experience was hosted on a remote server and the participants accessed the experience remotely using a link, while they wrote their comments and suggestions in a shared document file. Participants were encouraged to “think aloud” during the session, i.e. to express their thoughts and comments out loud while evaluating the experience.

### 4.5.3 Feedback regarding the Application’s Usability

In this section we will not be focusing on small usability issues such as unusual gaps between texts, button labels and colors and miswritten words, since those problems were immediately fixed upon detection. Instead, we will be presenting some of the most prominent and commonly observed usability issues.

The storytelling book component attracted a lot of attention and comments. Some participants noted that it would be quite useful to be able to flip the book’s pages by clicking on the pages themselves, instead of just having NEXT and BACK buttons as page-flipping controls. When it comes to the book’s audio narration, it was suggested by one participant to add extra audio controls that would allow the user to play and pause the narration of the chapter whenever they want and replay the recording after it ends.

Several users said that the experience took a long time to complete (approximately from 30 minutes to an hour), which seems like a really long duration for a crowdsourcing experience. In order to address this problem, several users suggested that a SKIP button be added to the annotation process of every district after the completion of the first dance move. This way the users will be able to skip the second and third dance move annotations if they want to. One participant thought that it would be extremely useful to add state-saving and state-resuming functionalities to the experience, meaning that the user would be able to save their progress and exit the application and resume the application’s state whenever they’d like. This would give the user the option to complete the experience in parts, whenever they have some time available to do so.

Another suggestion was adding a pagination system for the annotation description sets. Some users also thought that it would be more interesting if the environmental sounds of the districts were set at a higher volume.

When it comes to the application's responsiveness, some participants experienced text elements being rendered slightly out of the screen's bounds. We were anticipating such issues might come up since the responsiveness of the application has been implemented completely manually due to the lack of automatic handlers for the frame's responsiveness to the window's size in the A-Frame framework.

In spite of the issues described above, all participants described the application as simple to navigate through and overall quite user-friendly. No major usability issues were reported, the application was fully functional during all of the evaluations. The majority of the participants reported an overall smooth production run.

### 4.5.4 Feedback regarding the Application's "Look and Feel"

Aesthetically speaking, all participants reported to have enjoyed the application's general "look and feel". The colors, the illustrations, the animations and the sound effects were reported unanimously as beautiful and pleasant. Two participants said that the experience turned out to be quite soothing as well, due to a combination of narration recordings, ambient sounds and 3D models dancing smoothly on the screen. Last but not least, one participant mentioned that the dance motion capture animations look very well adapted on the dummy-doll 3D models, thus making the movement of the models look very natural and human-like.

### 4.5.5 Feedback regarding the Content of the Experience

Usability issues aside, the evaluation sessions were focused mainly on the content of the experience, the story, the theoretical context or maybe even possible plot-holes.

Starting from the storytelling book, two participants reported being a little confused by the narration of the story while viewing the book. They suggested that using simpler sentences for the narration, or maybe even a simpler scenario, could help make things clearer to the users. One participant reported that they misunderstood the audio narrations and thought that Celestine was not the one missing, but they thought that one of the four Elemental Empresses was missing instead. Another participant also said that the division of the story in 3 Chapters made the first part of the experience unnecessarily long and suggested that some parts are combined into one.

The Elemental Districts attracted quite some attention as well. When it comes to the interactions and greetings between the user and the Empresses, one participant said that they would have preferred if this part was omitted. They suggested that it would be better and more straightforward if the districts took the users to the annotation process right away. However, the rest of the participants reported to enjoy the Empresses' personalities and whimsical conversations. Moving on to the annotation process, several users suggested that it would be very helpful if they could have some sort of memo table mapping the LMA Efforts used in the description sets to movement qualities using the four motion factors, or maybe just explaining the LMA Effort actions intuitively.

One participant thought that the fact that there are no right or wrong answers to the annotation was not entirely clear to them and that they would have liked being reminded of that fact frequently while annotating qualities on the dance moves.

Additionally, most of the participants reported that they would have liked finding out which empress rules on which district and whether or not their answers to the questions associated with the empresses were correct (at least according to our judgment). One particular participant suggested that it would have been very helpful if there was a recap of all 3 dance moves of each district when they are asked whether or not the current empress is the ruler of the current district. That way they could review all the dance moves and make an informed decision.

Despite the issues described above, all participants reported to have enjoyed the experience. They said that the game-like environment, the beautiful graphics and the storyline can trick the users into partaking in crowdsourcing tasks that they would have probably turned down if presented in a more conventional form.

## 4.6 Future Extensions

The comments and suggestions presented in the previous section will be taken under careful consideration in the process of developing an improved version of the Motion Hollow experience. However, apart from these improvements, we have also thought of some extra functionalities that would be desirable for the application to have. Maybe we won’t end up implementing all of them, but it is important to make an outline of the ones we consider to be most useful and most possible to implement.

As mentioned in the previous section, adding state-defining functionality to the application would be highly beneficial, and thus it is something we probably want to look into in the future. The user will be able to save their progress and safely exit the application. When reopening the application, they will be able to pick up right where they left off, without having lost anything. In order to implement such a system, we would probably have to add an account authorization system to the application, so that the users can log in and log out using their credentials.

Another idea for a future extension is adding data visualization after the completion of the experience. Specifically, after reaching the end of the experience, the user will be prompted with all the data collected from the crowdsourcing carried out so far and they will be able to compare their answers to the answers registered to the database. The data will be presented using data visualization techniques such as layouts, charts, various colors, prioritizing orderings etc. It would be very interesting for the users to be able to compare their performance to the ones of others, and see the extent to which their perception of movement qualities differs from the perception of other users.

## 4.7 Conclusions

The main topic of this thesis was to design and develop a web-based crowdsourcing experience which would encourage the users to engage in the process of annotating movement qualities on dance content without feeling like they are doing a chore. Based on the feedback we received from the evaluation process, the Motion Hollow experience proves to succeed in its goal. In particular, the addition of the game elements proved to be a big asset to the experience. Elements like a story, a theme, stages, awards, sound effects, animations and a mission progress bar proved to be motivational. Even though the user knows that they are contributing to a study, the rest of the experience bears no resemblance to simple question-answering quizzes. The goal of finding the main character Celestine as required by the storyline is where the user focuses their attention on, thus forgetting that they are  partaking in a data collection activity.

However, there are things that need to be altered in order to make the experience better. For example, it might be beneficial to simplify the storyline of the experience, in order to keep the objectives clear and avoid confusion. Additionally, shortening the experience, adding skipping or state saving and resuming functionalities could prove very useful, since several users suggested that it took a long time to complete and they would have liked the flexibility of being able to pause and resume their progress at different time points.

To summarize, there are many improvements that can be applied to make the Motion Hollow experience better and more user-friendly. To be able to define the most desirable outcome and work towards achieving it, the application must be evaluated by a larger number of users. Nonetheless, the application managed to make an otherwise dull crowdsourcing procedure more appealing and pleasant. Putting all issues aside, the participants contributed to the study without feeling bored or wanting to quit at any point, which is what we were trying to accomplish in the first place.

# TABLE OF ABBREVIATIONS

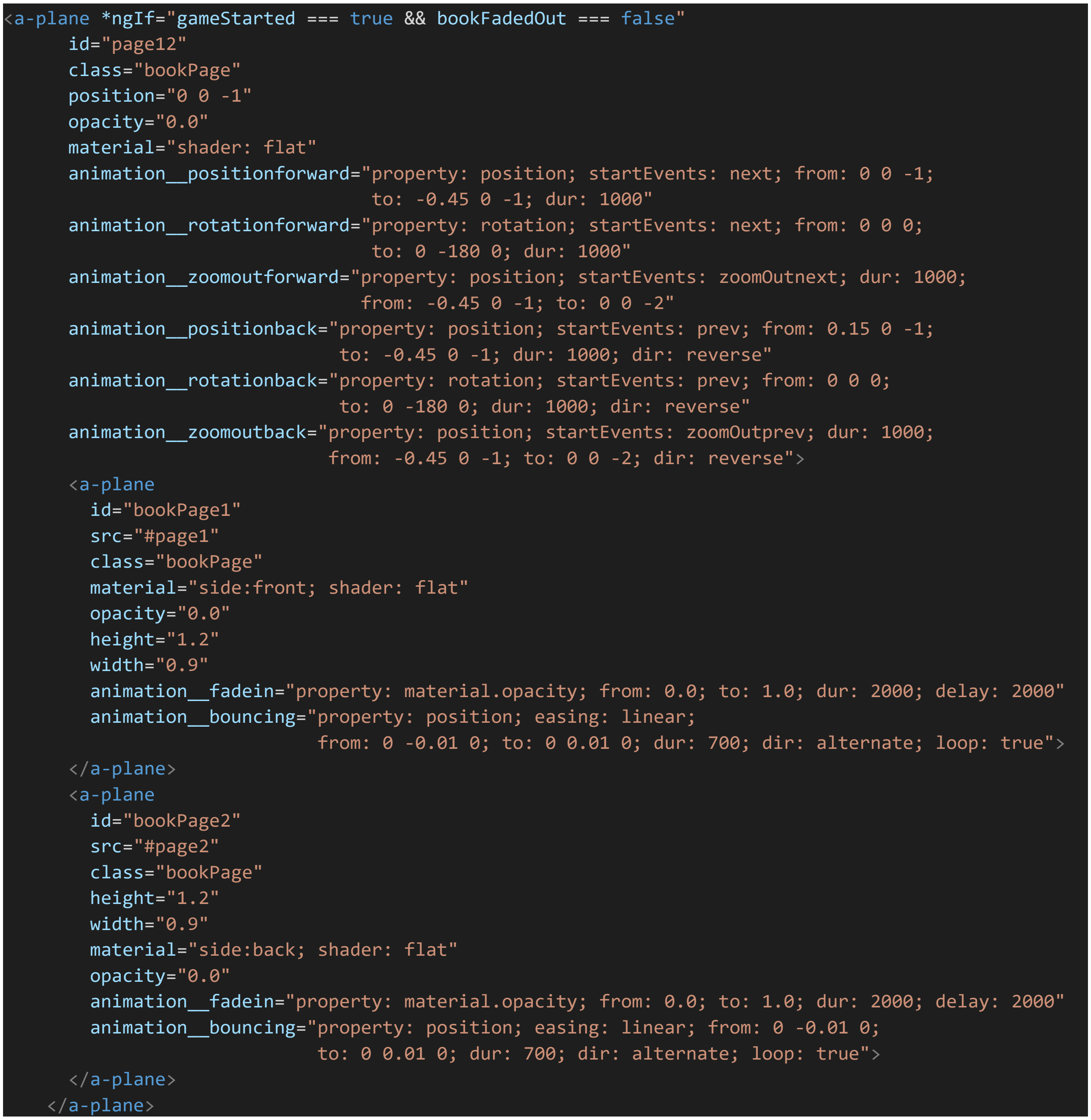
| | |
|---|---|
| **LMA** | Laban Movement Analysis |
| **HTA** | Hierarchical Task Analysis |
| **UI** | User Interface |
| **WebVR** | Web Virtual Reality |
| **UX** | User Experience |

# APPENDIX

This appendix provides sample code from the Motion Hollow project. The complete project is publicly available through a repository on Github[2].

## Sample code from the Motion Hollow project, file game.component.html

## Code for the first 2 pages of the storytelling book

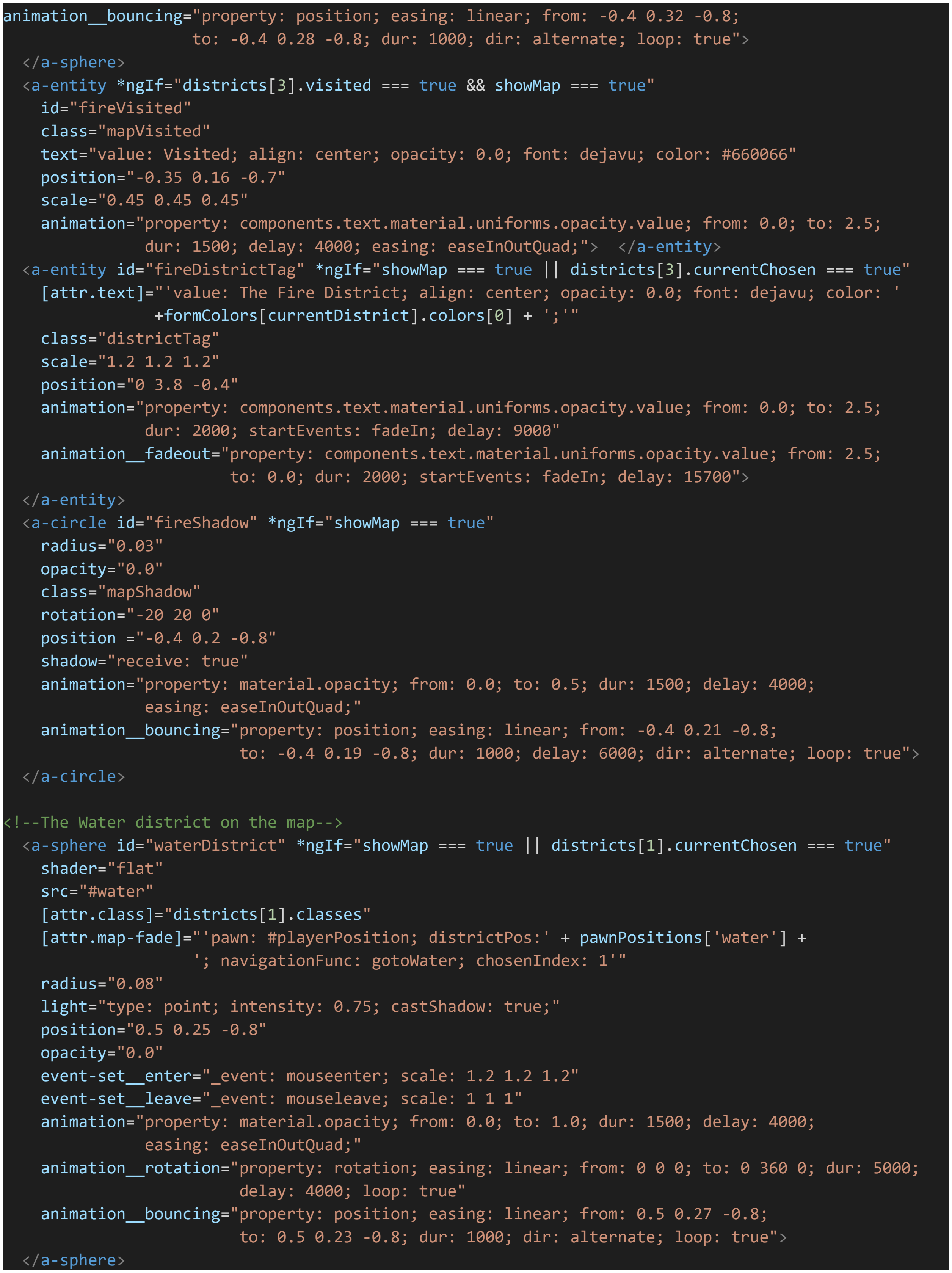


```
<a-plane *ngIf="gameStarted === true && bookFadedOut === false"
      id="page12"
      class="bookPage"
      position="0 0 -1"
      opacity="0.0"
      material="shader: flat"
      animation__positionforward="property: position; startEvents: next; from: 0 0 -1;
                                  to: -0.45 0 -1; dur: 1000"
      animation__rotationforward="property: rotation; startEvents: next; from: 0 0 0;
                                  to: 0 -180 0; dur: 1000"
      animation__zoomoutforward="property: position; startEvents: zoomOutnext; dur: 1000;
                                 from: -0.45 0 -1; to: 0 0 -2"
      animation__positionback="property: position; startEvents: prev; from: 0.15 0 -1;
                               to: -0.45 0 -1; dur: 1000; dir: reverse"
      animation__rotationback="property: rotation; startEvents: prev; from: 0 0 0;
                               to: 0 -180 0; dur: 1000; dir: reverse"
      animation__zoomoutback="property: position; startEvents: zoomOutprev; dur: 1000;
                              from: -0.45 0 -1; to: 0 0 -2; dir: reverse">
      <a-plane
        id="bookPage1"
        src="#page1"
        class="bookPage"
        material="side:front; shader: flat"
        opacity="0.0"
        height="1.2"
        width="0.9"
        animation__fadein="property: material.opacity; from: 0.0; to: 1.0; dur: 2000; delay: 2000"
        animation__bouncing="property: position; easing: linear;
                             from: 0 -0.01 0; to: 0 0.01 0; dur: 700; dir: alternate; loop: true">
      </a-plane>
      <a-plane
        id="bookPage2"
        src="#page2"
        class="bookPage"
        height="1.2"
        width="0.9"
        material="side:back; shader: flat"
        opacity="0.0"
        animation__fadein="property: material.opacity; from: 0.0; to: 1.0; dur: 2000; delay: 2000"
        animation__bouncing="property: position; easing: linear; from: 0 -0.01 0;
                             to: 0 0.01 0; dur: 700; dir: alternate; loop: true">
      </a-plane>
    </a-plane>
```

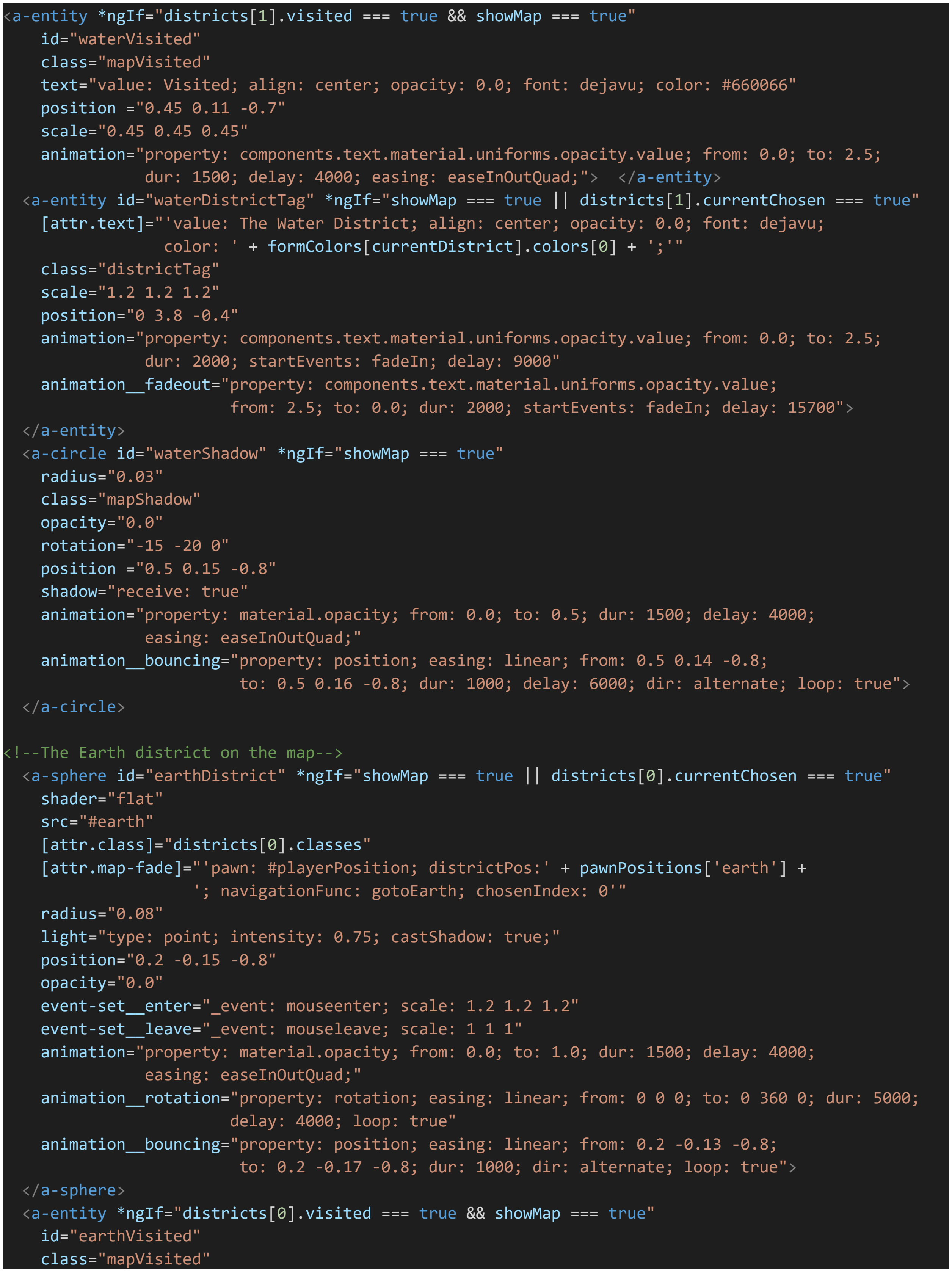
[2] https://github.com/LoriKougiou/thesis-Motion-Hollow

## Code for the Motion Hollow map

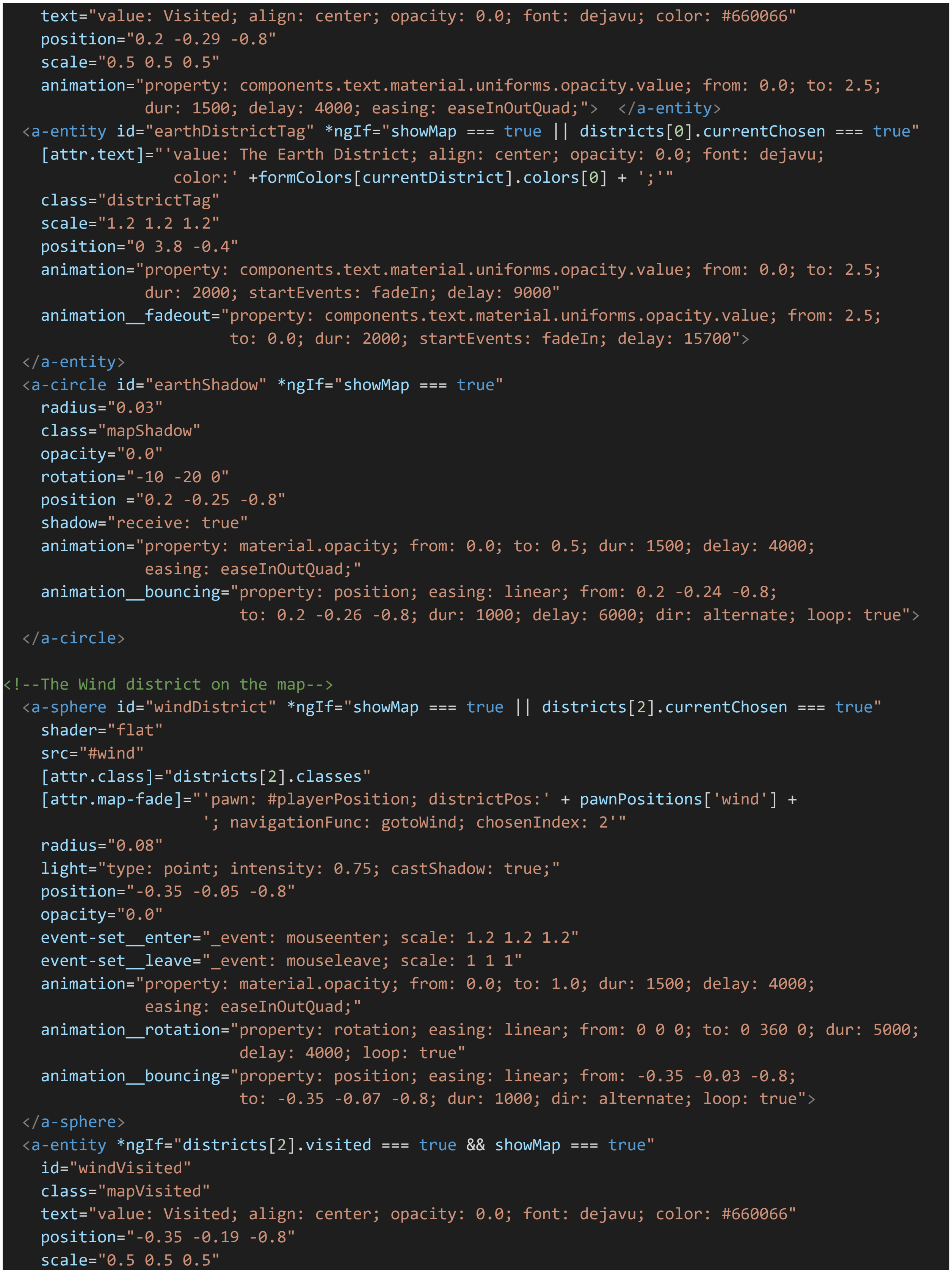
```
<!--Map entity - Everything associated with the map-->
  <a-entity id="mapText1" *ngIf="showMap === true"
    text="value: Choose where to go next; align: center; opacity: 0.0; font: dejavu;
          color: #ffff80"
    class="mapPrompt"
    scale="1.5 1.5 1.5"
    position="0 0.65 -0.9"
    animation="property: components.text.material.uniforms.opacity.value; from: 0.0;
               to: 2.5; dur: 2000">
    <a-entity id="mapText2" *ngIf="showMap === true"
      text="value:You must visit all districts; align: center; opacity: 0.0;
            font: dejavu; color: #ffff80"
      class="mapPrompt"
      scale="0.7 0.7 0.7"
      position="0 -0.05 0"
      animation="property: components.text.material.uniforms.opacity.value; from: 0.0;
                 to: 2.5; dur: 2000">
    </a-entity>
  </a-entity>
  <a-image id="mapBackground" *ngIf="showMap === true"
    src="#map"
    class="mapImg"
    position="0 0 -1"
    rotation="-10 0 0"
    opacity="0.0"
    scale="2.3 1.2 1.2"
    animation="property: material.opacity; from: 0.0; to: 1.0; dur: 1500; delay: 2000;
               easing: easeInOutQuad;"
    animation__bouncing="property: position; easing: linear; from: 0 0.01 -1;
                        to: 0 -0.01 -1; dur: 700; dir: alternate; loop: true">
  </a-image>

<!--Spheres representing the districts-->

<!--The Fire district on the map-->
  <a-sphere  *ngIf="showMap === true || districts[3].currentChosen === true"
    id="fireDistrict"
    shader="flat"
    src="#fire"
    [attr.class]="districts[3].classes"
    [attr.map-fade]="'pawn: #playerPosition; districtPos:' + pawnPositions['fire'] +
                    '; navigationFunc: gotoFire; chosenIndex: 3'"
    radius="0.08"
    light="type: point; intensity: 0.75; castShadow: true;"
    position="-0.4 0.3 -0.8"
    opacity="0.0"
    event-set__enter="_event: mouseenter; scale: 1.2 1.2 1.2"
    event-set__leave="_event: mouseleave; scale: 1 1 1"
    animation="property: material.opacity; from: 0.0; to: 1.0; dur: 1500; delay: 4000;
               easing: easeInOutQuad;"
    animation__rotation="property: rotation; easing: linear; from: 0 0 0; to: 0 360 0;
```



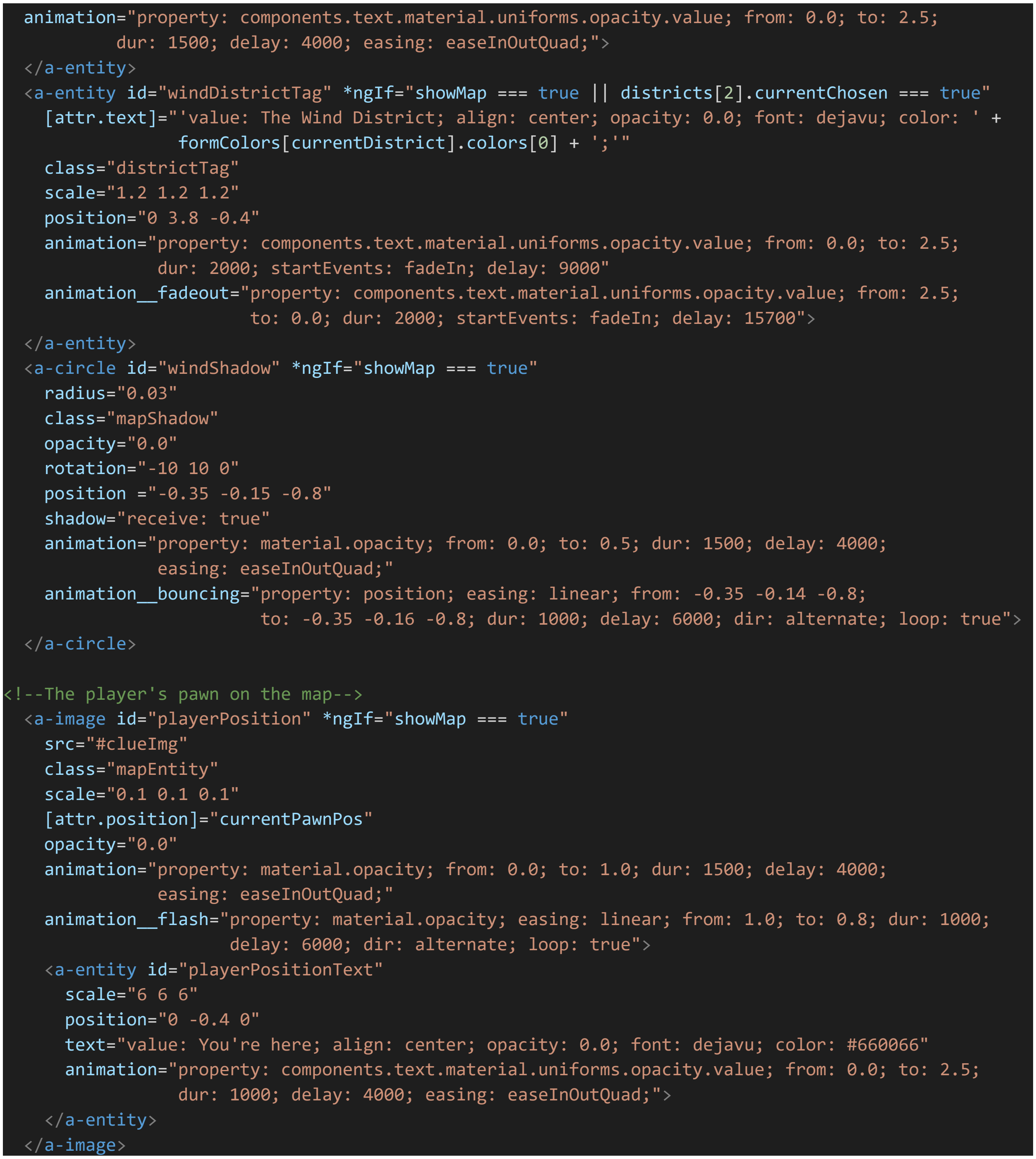
```
animation__bouncing="property: position; easing: linear; from: -0.4 0.32 -0.8;
                    to: -0.4 0.28 -0.8; dur: 1000; dir: alternate; loop: true">
  </a-sphere>
  <a-entity *ngIf="districts[3].visited === true && showMap === true"
    id="fireVisited"
    class="mapVisited"
    text="value: Visited; align: center; opacity: 0.0; font: dejavu; color: #660066"
    position="-0.35 0.16 -0.7"
    scale="0.45 0.45 0.45"
    animation="property: components.text.material.uniforms.opacity.value; from: 0.0; to: 2.5;
               dur: 1500; delay: 4000; easing: easeInOutQuad;">  </a-entity>
  <a-entity id="fireDistrictTag" *ngIf="showMap === true || districts[3].currentChosen === true"
    [attr.text]="'value: The Fire District; align: center; opacity: 0.0; font: dejavu; color: '
                +formColors[currentDistrict].colors[0] + ';'"
    class="districtTag"
    scale="1.2 1.2 1.2"
    position="0 3.8 -0.4"
    animation="property: components.text.material.uniforms.opacity.value; from: 0.0; to: 2.5;
               dur: 2000; startEvents: fadeIn; delay: 9000"
    animation__fadeout="property: components.text.material.uniforms.opacity.value; from: 2.5;
                        to: 0.0; dur: 2000; startEvents: fadeIn; delay: 15700">
  </a-entity>
  <a-circle id="fireShadow" *ngIf="showMap === true"
    radius="0.03"
    opacity="0.0"
    class="mapShadow"
    rotation="-20 20 0"
    position ="-0.4 0.2 -0.8"
    shadow="receive: true"
    animation="property: material.opacity; from: 0.0; to: 0.5; dur: 1500; delay: 4000;
               easing: easeInOutQuad;"
    animation__bouncing="property: position; easing: linear; from: -0.4 0.21 -0.8;
                         to: -0.4 0.19 -0.8; dur: 1000; delay: 6000; dir: alternate; loop: true">
  </a-circle>

<!--The Water district on the map-->
  <a-sphere id="waterDistrict" *ngIf="showMap === true || districts[1].currentChosen === true"
    shader="flat"
    src="#water"
    [attr.class]="districts[1].classes"
    [attr.map-fade]="'pawn: #playerPosition; districtPos:' + pawnPositions['water'] +
                    '; navigationFunc: gotoWater; chosenIndex: 1'"
    radius="0.08"
    light="type: point; intensity: 0.75; castShadow: true;"
    position="0.5 0.25 -0.8"
    opacity="0.0"
    event-set__enter="_event: mouseenter; scale: 1.2 1.2 1.2"
    event-set__leave="_event: mouseleave; scale: 1 1 1"
    animation="property: material.opacity; from: 0.0; to: 1.0; dur: 1500; delay: 4000;
               easing: easeInOutQuad;"
    animation__rotation="property: rotation; easing: linear; from: 0 0 0; to: 0 360 0; dur: 5000;
                         delay: 4000; loop: true"
    animation__bouncing="property: position; easing: linear; from: 0.5 0.27 -0.8;
                         to: 0.5 0.23 -0.8; dur: 1000; dir: alternate; loop: true">
  </a-sphere>
```



```
<a-entity *ngIf="districts[1].visited === true && showMap === true"
    id="waterVisited"
    class="mapVisited"
    text="value: Visited; align: center; opacity: 0.0; font: dejavu; color: #660066"
    position ="0.45 0.11 -0.7"
    scale="0.45 0.45 0.45"
    animation="property: components.text.material.uniforms.opacity.value; from: 0.0; to: 2.5;
               dur: 1500; delay: 4000; easing: easeInOutQuad;">  </a-entity>
  <a-entity id="waterDistrictTag" *ngIf="showMap === true || districts[1].currentChosen === true"
    [attr.text]="'value: The Water District; align: center; opacity: 0.0; font: dejavu;
                 color: ' + formColors[currentDistrict].colors[0] + ';'"
    class="districtTag"
    scale="1.2 1.2 1.2"
    position="0 3.8 -0.4"
    animation="property: components.text.material.uniforms.opacity.value; from: 0.0; to: 2.5;
               dur: 2000; startEvents: fadeIn; delay: 9000"
    animation__fadeout="property: components.text.material.uniforms.opacity.value;
                        from: 2.5; to: 0.0; dur: 2000; startEvents: fadeIn; delay: 15700">
  </a-entity>
  <a-circle id="waterShadow" *ngIf="showMap === true"
    radius="0.03"
    class="mapShadow"
    opacity="0.0"
    rotation="-15 -20 0"
    position ="0.5 0.15 -0.8"
    shadow="receive: true"
    animation="property: material.opacity; from: 0.0; to: 0.5; dur: 1500; delay: 4000;
               easing: easeInOutQuad;"
    animation__bouncing="property: position; easing: linear; from: 0.5 0.14 -0.8;
                         to: 0.5 0.16 -0.8; dur: 1000; delay: 6000; dir: alternate; loop: true">
  </a-circle>

<!--The Earth district on the map-->
  <a-sphere id="earthDistrict" *ngIf="showMap === true || districts[0].currentChosen === true"
    shader="flat"
    src="#earth"
    [attr.class]="districts[0].classes"
    [attr.map-fade]="'pawn: #playerPosition; districtPos:' + pawnPositions['earth'] +
                    '; navigationFunc: gotoEarth; chosenIndex: 0'"
    radius="0.08"
    light="type: point; intensity: 0.75; castShadow: true;"
    position="0.2 -0.15 -0.8"
    opacity="0.0"
    event-set__enter="_event: mouseenter; scale: 1.2 1.2 1.2"
    event-set__leave="_event: mouseleave; scale: 1 1 1"
    animation="property: material.opacity; from: 0.0; to: 1.0; dur: 1500; delay: 4000;
               easing: easeInOutQuad;"
    animation__rotation="property: rotation; easing: linear; from: 0 0 0; to: 0 360 0; dur: 5000;
                        delay: 4000; loop: true"
    animation__bouncing="property: position; easing: linear; from: 0.2 -0.13 -0.8;
                         to: 0.2 -0.17 -0.8; dur: 1000; dir: alternate; loop: true">
  </a-sphere>
  <a-entity *ngIf="districts[0].visited === true && showMap === true"
    id="earthVisited"
    class="mapVisited"
```

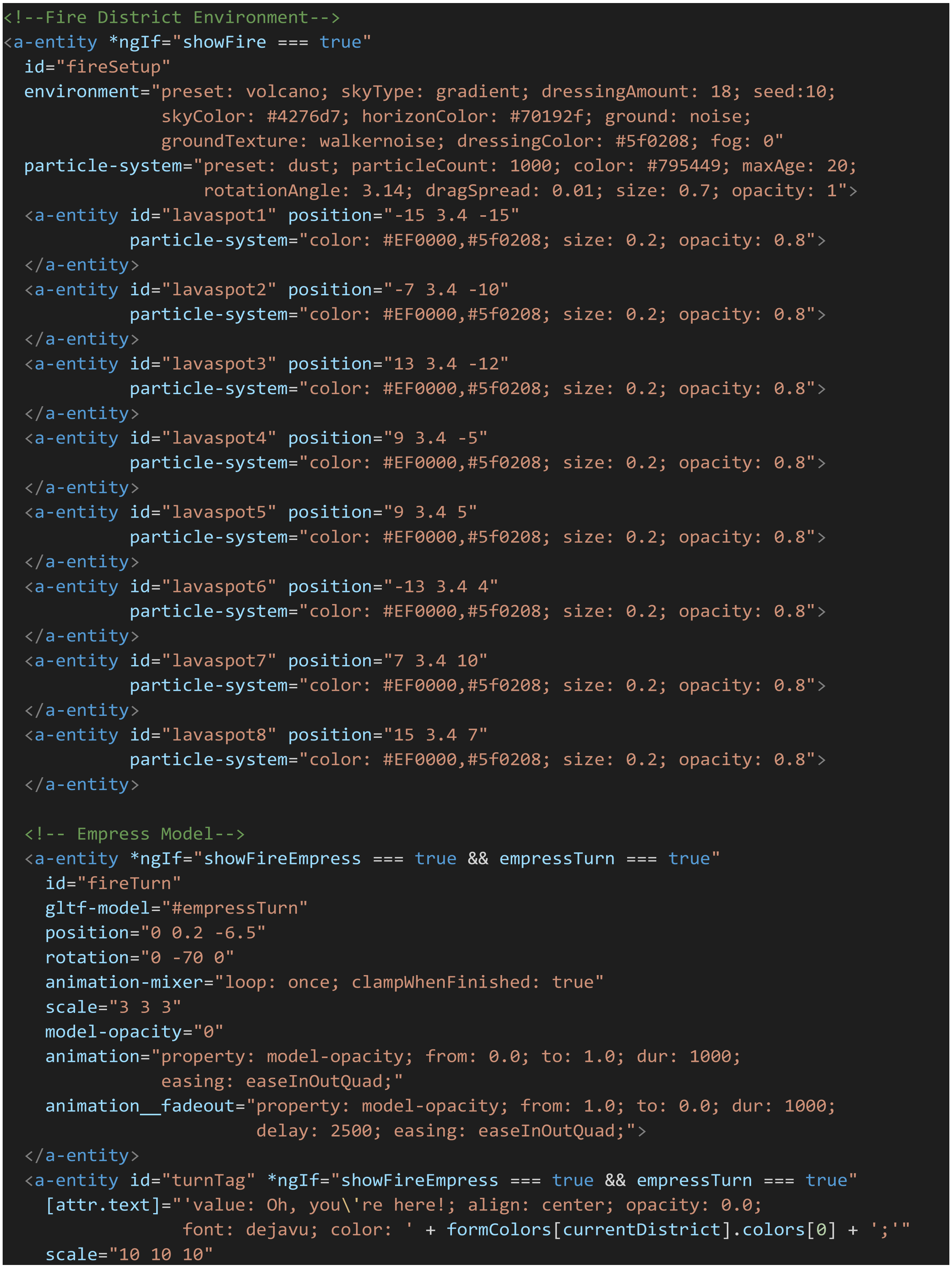


```
    text="value: Visited; align: center; opacity: 0.0; font: dejavu; color: #660066"
    position="0.2 -0.29 -0.8"
    scale="0.5 0.5 0.5"
    animation="property: components.text.material.uniforms.opacity.value; from: 0.0; to: 2.5;
               dur: 1500; delay: 4000; easing: easeInOutQuad;">  </a-entity>
  <a-entity id="earthDistrictTag" *ngIf="showMap === true || districts[0].currentChosen === true"
    [attr.text]="'value: The Earth District; align: center; opacity: 0.0; font: dejavu;
                  color:' +formColors[currentDistrict].colors[0] + ';'"
    class="districtTag"
    scale="1.2 1.2 1.2"
    position="0 3.8 -0.4"
    animation="property: components.text.material.uniforms.opacity.value; from: 0.0; to: 2.5;
               dur: 2000; startEvents: fadeIn; delay: 9000"
    animation__fadeout="property: components.text.material.uniforms.opacity.value; from: 2.5;
                        to: 0.0; dur: 2000; startEvents: fadeIn; delay: 15700">
  </a-entity>
  <a-circle id="earthShadow" *ngIf="showMap === true"
    radius="0.03"
    class="mapShadow"
    opacity="0.0"
    rotation="-10 -20 0"
    position ="0.2 -0.25 -0.8"
    shadow="receive: true"
    animation="property: material.opacity; from: 0.0; to: 0.5; dur: 1500; delay: 4000;
               easing: easeInOutQuad;"
    animation__bouncing="property: position; easing: linear; from: 0.2 -0.24 -0.8;
                         to: 0.2 -0.26 -0.8; dur: 1000; delay: 6000; dir: alternate; loop: true">
  </a-circle>

<!--The Wind district on the map-->
  <a-sphere id="windDistrict" *ngIf="showMap === true || districts[2].currentChosen === true"
    shader="flat"
    src="#wind"
    [attr.class]="districts[2].classes"
    [attr.map-fade]="'pawn: #playerPosition; districtPos:' + pawnPositions['wind'] +
                     '; navigationFunc: gotoWind; chosenIndex: 2'"
    radius="0.08"
    light="type: point; intensity: 0.75; castShadow: true;"
    position="-0.35 -0.05 -0.8"
    opacity="0.0"
    event-set__enter="_event: mouseenter; scale: 1.2 1.2 1.2"
    event-set__leave="_event: mouseleave; scale: 1 1 1"
    animation="property: material.opacity; from: 0.0; to: 1.0; dur: 1500; delay: 4000;
               easing: easeInOutQuad;"
    animation__rotation="property: rotation; easing: linear; from: 0 0 0; to: 0 360 0; dur: 5000;
                         delay: 4000; loop: true"
    animation__bouncing="property: position; easing: linear; from: -0.35 -0.03 -0.8;
                         to: -0.35 -0.07 -0.8; dur: 1000; dir: alternate; loop: true">
  </a-sphere>
  <a-entity *ngIf="districts[2].visited === true && showMap === true"
    id="windVisited"
    class="mapVisited"
    text="value: Visited; align: center; opacity: 0.0; font: dejavu; color: #660066"
    position="-0.35 -0.19 -0.8"
    scale="0.5 0.5 0.5"
```

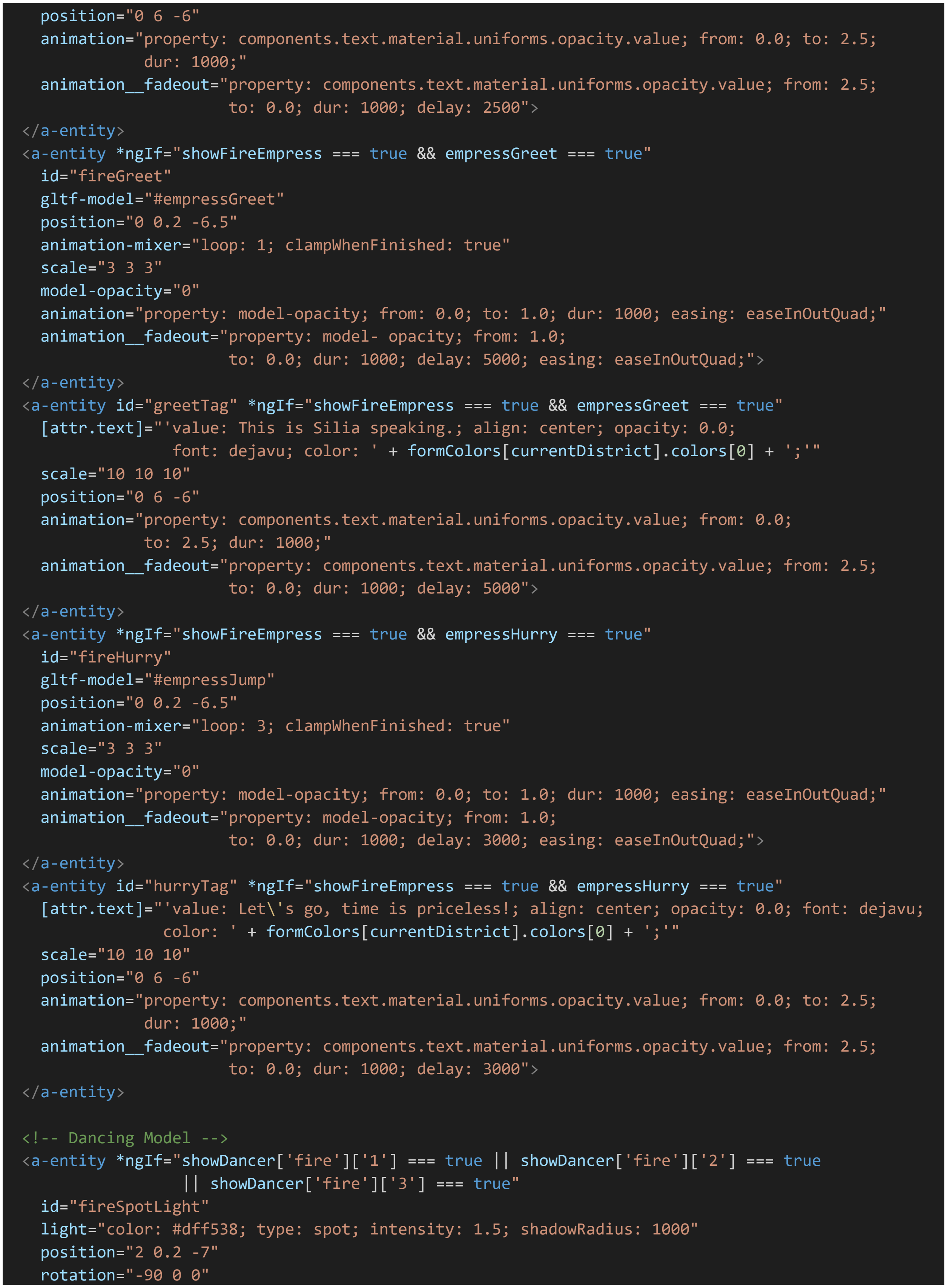


```
  animation="property: components.text.material.uniforms.opacity.value; from: 0.0; to: 2.5;
           dur: 1500; delay: 4000; easing: easeInOutQuad;">
  </a-entity>
  <a-entity id="windDistrictTag" *ngIf="showMap === true || districts[2].currentChosen === true"
    [attr.text]="'value: The Wind District; align: center; opacity: 0.0; font: dejavu; color: ' +
                 formColors[currentDistrict].colors[0] + ';'"
    class="districtTag"
    scale="1.2 1.2 1.2"
    position="0 3.8 -0.4"
    animation="property: components.text.material.uniforms.opacity.value; from: 0.0; to: 2.5;
               dur: 2000; startEvents: fadeIn; delay: 9000"
    animation__fadeout="property: components.text.material.uniforms.opacity.value; from: 2.5;
                        to: 0.0; dur: 2000; startEvents: fadeIn; delay: 15700">
  </a-entity>
  <a-circle id="windShadow" *ngIf="showMap === true"
    radius="0.03"
    class="mapShadow"
    opacity="0.0"
    rotation="-10 10 0"
    position ="-0.35 -0.15 -0.8"
    shadow="receive: true"
    animation="property: material.opacity; from: 0.0; to: 0.5; dur: 1500; delay: 4000;
               easing: easeInOutQuad;"
    animation__bouncing="property: position; easing: linear; from: -0.35 -0.14 -0.8;
                         to: -0.35 -0.16 -0.8; dur: 1000; delay: 6000; dir: alternate; loop: true">
  </a-circle>

<!--The player's pawn on the map-->
  <a-image id="playerPosition" *ngIf="showMap === true"
    src="#clueImg"
    class="mapEntity"
    scale="0.1 0.1 0.1"
    [attr.position]="currentPawnPos"
    opacity="0.0"
    animation="property: material.opacity; from: 0.0; to: 1.0; dur: 1500; delay: 4000;
               easing: easeInOutQuad;"
    animation__flash="property: material.opacity; easing: linear; from: 1.0; to: 0.8; dur: 1000;
                      delay: 6000; dir: alternate; loop: true">
    <a-entity id="playerPositionText"
      scale="6 6 6"
      position="0 -0.4 0"
      text="value: You're here; align: center; opacity: 0.0; font: dejavu; color: #660066"
      animation="property: components.text.material.uniforms.opacity.value; from: 0.0; to: 2.5;
                 dur: 1000; delay: 4000; easing: easeInOutQuad;">
    </a-entity>
  </a-image>
```

## Code for the Fire District Environment



```
<!--Fire District Environment-->
<a-entity *ngIf="showFire === true"
  id="fireSetup"
  environment="preset: volcano; skyType: gradient; dressingAmount: 18; seed:10;
               skyColor: #4276d7; horizonColor: #70192f; ground: noise;
               groundTexture: walkernoise; dressingColor: #5f0208; fog: 0"
  particle-system="preset: dust; particleCount: 1000; color: #795449; maxAge: 20;
                   rotationAngle: 3.14; dragSpread: 0.01; size: 0.7; opacity: 1">
  <a-entity id="lavaspot1" position="-15 3.4 -15"
            particle-system="color: #EF0000,#5f0208; size: 0.2; opacity: 0.8">
  </a-entity>
  <a-entity id="lavaspot2" position="-7 3.4 -10"
            particle-system="color: #EF0000,#5f0208; size: 0.2; opacity: 0.8">
  </a-entity>
  <a-entity id="lavaspot3" position="13 3.4 -12"
            particle-system="color: #EF0000,#5f0208; size: 0.2; opacity: 0.8">
  </a-entity>
  <a-entity id="lavaspot4" position="9 3.4 -5"
            particle-system="color: #EF0000,#5f0208; size: 0.2; opacity: 0.8">
  </a-entity>
  <a-entity id="lavaspot5" position="9 3.4 5"
            particle-system="color: #EF0000,#5f0208; size: 0.2; opacity: 0.8">
  </a-entity>
  <a-entity id="lavaspot6" position="-13 3.4 4"
            particle-system="color: #EF0000,#5f0208; size: 0.2; opacity: 0.8">
  </a-entity>
  <a-entity id="lavaspot7" position="7 3.4 10"
            particle-system="color: #EF0000,#5f0208; size: 0.2; opacity: 0.8">
  </a-entity>
  <a-entity id="lavaspot8" position="15 3.4 7"
            particle-system="color: #EF0000,#5f0208; size: 0.2; opacity: 0.8">
  </a-entity>

  <!-- Empress Model-->
  <a-entity *ngIf="showFireEmpress === true && empressTurn === true"
    id="fireTurn"
    gltf-model="#empressTurn"
    position="0 0.2 -6.5"
    rotation="0 -70 0"
    animation-mixer="loop: once; clampWhenFinished: true"
    scale="3 3 3"
    model-opacity="0"
    animation="property: model-opacity; from: 0.0; to: 1.0; dur: 1000;
               easing: easeInOutQuad;"
    animation__fadeout="property: model-opacity; from: 1.0; to: 0.0; dur: 1000;
                        delay: 2500; easing: easeInOutQuad;">
  </a-entity>
  <a-entity id="turnTag" *ngIf="showFireEmpress === true && empressTurn === true"
    [attr.text]="'value: Oh, you\'re here!; align: center; opacity: 0.0;
                 font: dejavu; color: ' + formColors[currentDistrict].colors[0] + ';'"
    scale="10 10 10"
```

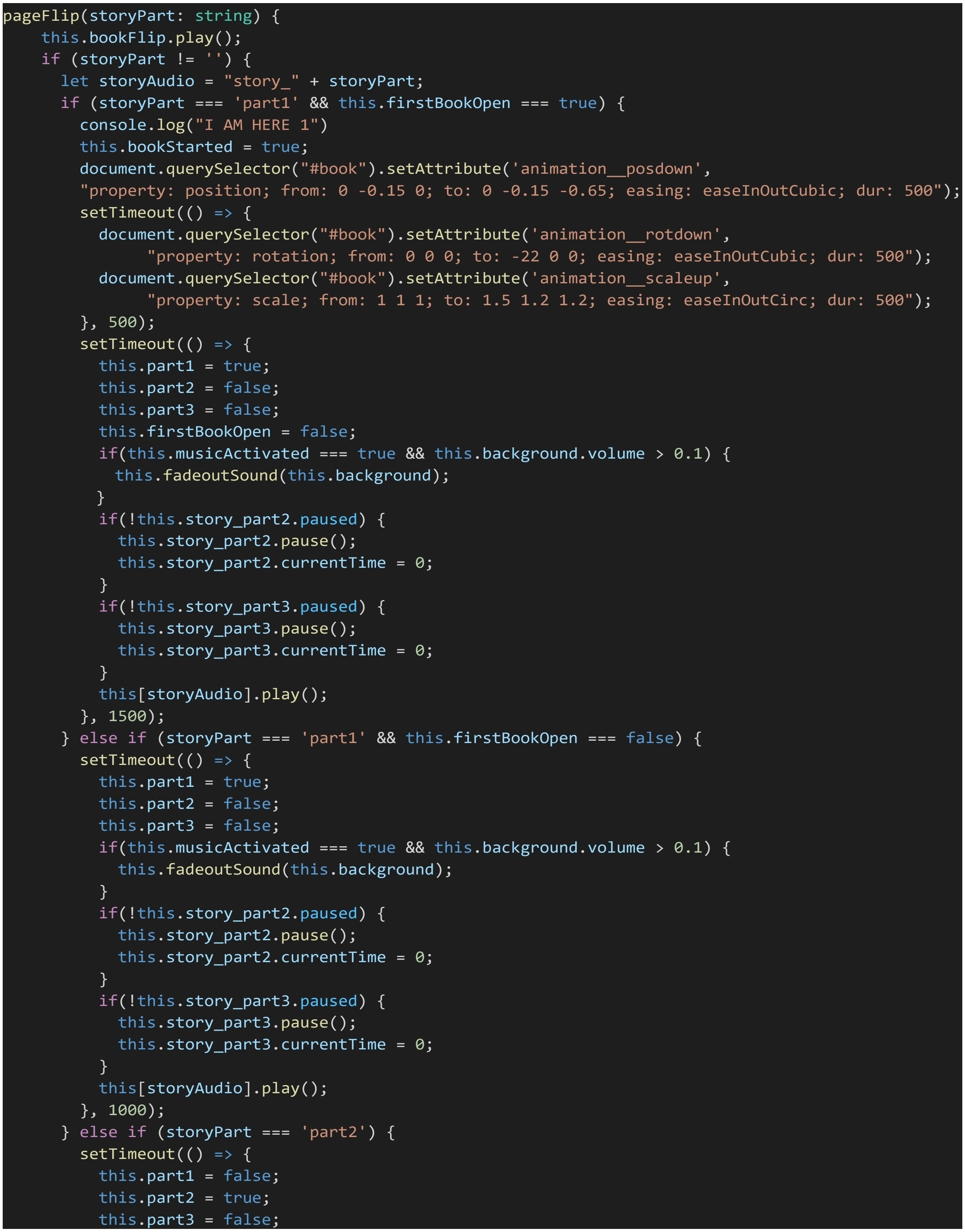


```
  position="0 6 -6"
  animation="property: components.text.material.uniforms.opacity.value; from: 0.0; to: 2.5;
             dur: 1000;"
  animation__fadeout="property: components.text.material.uniforms.opacity.value; from: 2.5;
                      to: 0.0; dur: 1000; delay: 2500">
</a-entity>
<a-entity *ngIf="showFireEmpress === true && empressGreet === true"
  id="fireGreet"
  gltf-model="#empressGreet"
  position="0 0.2 -6.5"
  animation-mixer="loop: 1; clampWhenFinished: true"
  scale="3 3 3"
  model-opacity="0"
  animation="property: model-opacity; from: 0.0; to: 1.0; dur: 1000; easing: easeInOutQuad;"
  animation__fadeout="property: model- opacity; from: 1.0;
                      to: 0.0; dur: 1000; delay: 5000; easing: easeInOutQuad;">
</a-entity>
<a-entity id="greetTag" *ngIf="showFireEmpress === true && empressGreet === true"
  [attr.text]="'value: This is Silia speaking.; align: center; opacity: 0.0;
                font: dejavu; color: ' + formColors[currentDistrict].colors[0] + ';'"
  scale="10 10 10"
  position="0 6 -6"
  animation="property: components.text.material.uniforms.opacity.value; from: 0.0;
             to: 2.5; dur: 1000;"
  animation__fadeout="property: components.text.material.uniforms.opacity.value; from: 2.5;
                      to: 0.0; dur: 1000; delay: 5000">
</a-entity>
<a-entity *ngIf="showFireEmpress === true && empressHurry === true"
  id="fireHurry"
  gltf-model="#empressJump"
  position="0 0.2 -6.5"
  animation-mixer="loop: 3; clampWhenFinished: true"
  scale="3 3 3"
  model-opacity="0"
  animation="property: model-opacity; from: 0.0; to: 1.0; dur: 1000; easing: easeInOutQuad;"
  animation__fadeout="property: model-opacity; from: 1.0;
                      to: 0.0; dur: 1000; delay: 3000; easing: easeInOutQuad;">
</a-entity>
<a-entity id="hurryTag" *ngIf="showFireEmpress === true && empressHurry === true"
  [attr.text]="'value: Let\'s go, time is priceless!; align: center; opacity: 0.0; font: dejavu;
               color: ' + formColors[currentDistrict].colors[0] + ';'"
  scale="10 10 10"
  position="0 6 -6"
  animation="property: components.text.material.uniforms.opacity.value; from: 0.0; to: 2.5;
             dur: 1000;"
  animation__fadeout="property: components.text.material.uniforms.opacity.value; from: 2.5;
                      to: 0.0; dur: 1000; delay: 3000">
</a-entity>

<!-- Dancing Model -->
<a-entity *ngIf="showDancer['fire']['1'] === true || showDancer['fire']['2'] === true
                 || showDancer['fire']['3'] === true"
  id="fireSpotLight"
  light="color: #dff538; type: spot; intensity: 1.5; shadowRadius: 1000"
  position="2 0.2 -7"
  rotation="-90 0 0"
```

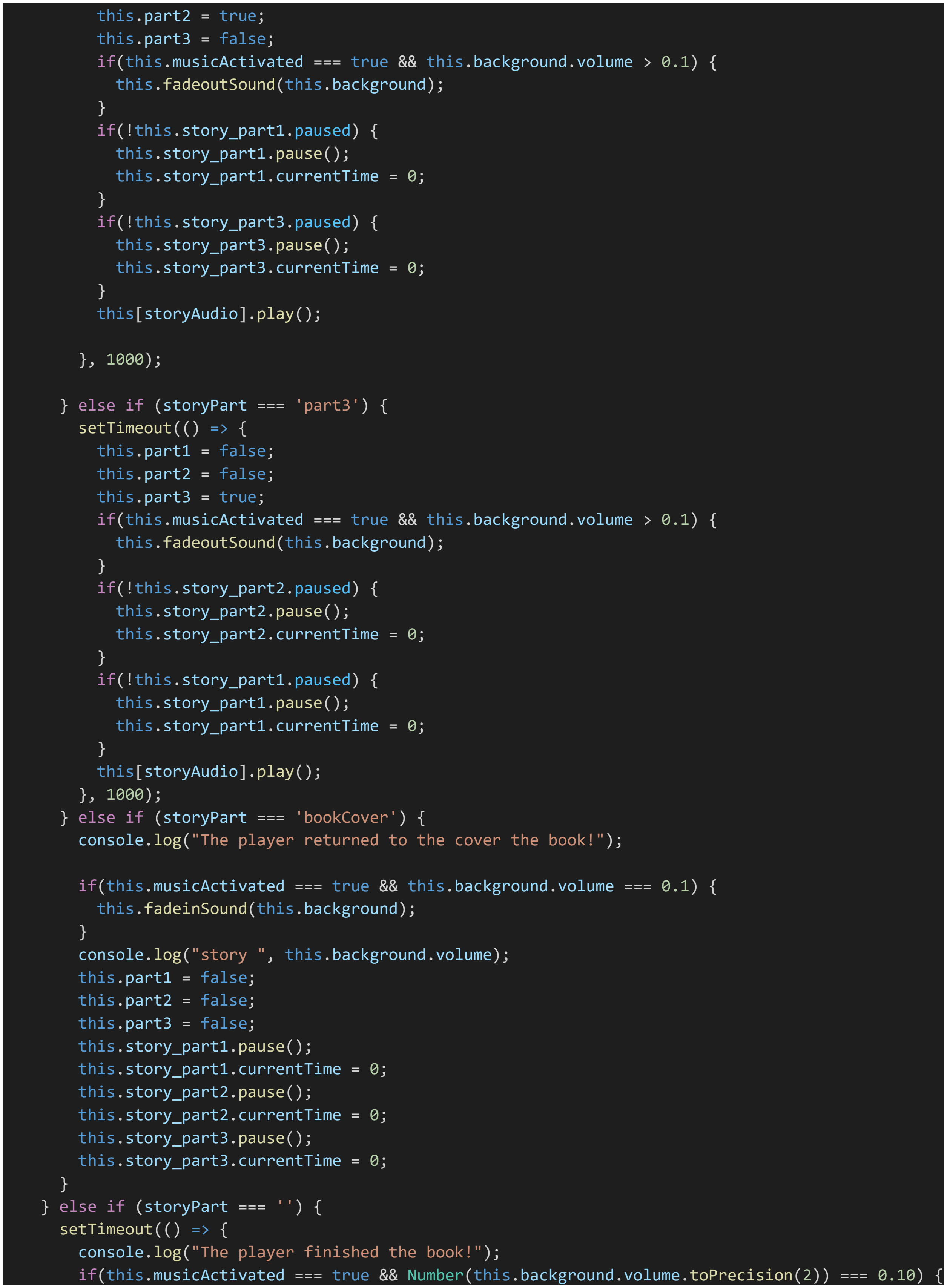
```
    scale="1 1 1"
    animation="property: material.opacity; from: 0.0; to: 1.0; dur: 1000;">
  </a-entity>
  <a-entity *ngIf="showDancer['fire']['1'] === true"
    id="fireModel1"
    position="-2.5 0.2 -8"
    gltf-model="#centerPirouette"
    animation-mixer
    scale="3 3 3"
    model-opacity="0"
    animation="property: model-opacity; from: 0.0; to: 1.0; dur: 1000; easing: easeInOutQuad;">
  </a-entity>
  <a-entity *ngIf="showDancer['fire']['1'] === true"
    id="fireMoveTag1"
    [attr.text]="'value: for the first dance; align: left; opacity: 0.0; font: dejavu; color: ' + fo
rmColors[currentDistrict].colors[0] + ';'"
    scale="3 3 3"
    position="0.8 4.92 -2"
    animation="property: components.text.material.uniforms.opacity.value; from: 0.0; to: 2.5;
               dur: 2000; delay: 1000;">
  </a-entity>
  <a-entity *ngIf="showDancer['fire']['2'] ===  true"
    id="fireModel2"
    position="2 0.2 -7.5"
    gltf-model="#weightTravelSpace"
    animation-mixer
    scale="3 3 3"
    model-opacity="0"
    animation="property: model-opacity; from: 0.0; to: 1.0; dur: 1000; easing: easeInOutQuad;">
  </a-entity>
  <a-entity *ngIf="showDancer['fire']['2'] === true"
    id="fireMoveTag2"
    [attr.text]="'value: for the second dance; align: left; opacity: 0.0; font: dejavu; color: '
                 + formColors[currentDistrict].colors[0] + ';'"
    scale="3 3 3"
    position="0.8 4.92 -2"
    animation="property: components.text.material.uniforms.opacity.value; from: 0.0; to: 2.5;
               dur: 1000; delay: 1000">
  </a-entity>
  <a-entity *ngIf="showDancer['fire']['3'] ===  true"
    id="fireModel3"
    position="1.2 0.2 -8.5"
    gltf-model="#centerJump"
    animation-mixer
    scale="3.2 3.2 3.2"
    model-opacity="0"
    animation="property: model-opacity; from: 0.0; to: 1.0; dur: 1000; easing: easeInOutQuad;">
  </a-entity>
  <a-entity *ngIf="showDancer['fire']['3'] ===  true && showEmpressQuestion === false"
    id="fireMoveTag3"
    [attr.text]="'value: for the third dance; align: left; opacity: 0.0; font: dejavu; color: '
                  + formColors[currentDistrict].colors[0] + ';'"
    scale="3 3 3"
    position="0.8 4.92 -2"
    animation="property: components.text.material.uniforms.opacity.value; from: 0.0; to: 2.5;
               dur: 1000; delay: 1000">  </a-entity> </a-entity>
```

## Sample code from the Motion Hollow project, file game.component.ts

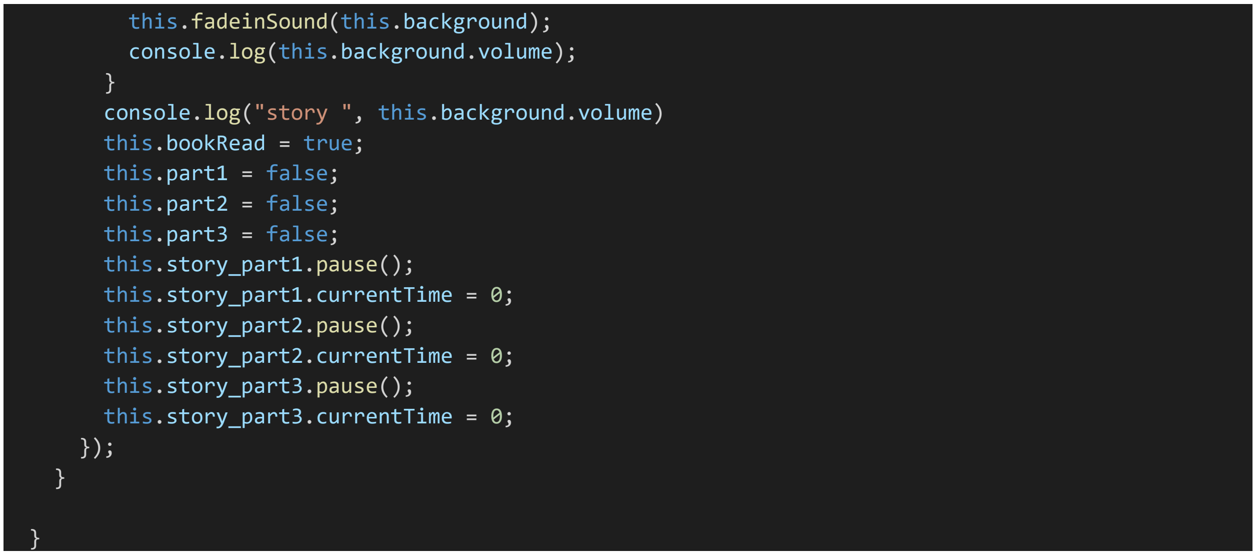
## Function that handles the page-flips of the storytelling book



```
pageFlip(storyPart: string) {
  this.bookFlip.play();
  if (storyPart != '') {
   let storyAudio = "story_" + storyPart;
   if (storyPart === 'part1' && this.firstBookOpen === true) {
    console.log("I AM HERE 1")
    this.bookStarted = true;
    document.querySelector("#book").setAttribute('animation__posdown',
    "property: position; from: 0 -0.15 0; to: 0 -0.15 -0.65; easing: easeInOutCubic; dur: 500");
    setTimeout(() => {
     document.querySelector("#book").setAttribute('animation__rotdown',
        "property: rotation; from: 0 0 0; to: -22 0 0; easing: easeInOutCubic; dur: 500");
     document.querySelector("#book").setAttribute('animation__scaleup',
        "property: scale; from: 1 1 1; to: 1.5 1.2 1.2; easing: easeInOutCirc; dur: 500");
    }, 500);
    setTimeout(() => {
     this.part1 = true;
     this.part2 = false;
     this.part3 = false;
     this.firstBookOpen = false;
     if(this.musicActivated === true && this.background.volume > 0.1) {
      this.fadeoutSound(this.background);
     }
     if(!this.story_part2.paused) {
      this.story_part2.pause();
      this.story_part2.currentTime = 0;
     }
     if(!this.story_part3.paused) {
      this.story_part3.pause();
      this.story_part3.currentTime = 0;
     }
     this[storyAudio].play();
    }, 1500);
   } else if (storyPart === 'part1' && this.firstBookOpen === false) {
    setTimeout(() => {
     this.part1 = true;
     this.part2 = false;
     this.part3 = false;
     if(this.musicActivated === true && this.background.volume > 0.1) {
      this.fadeoutSound(this.background);
     }
     if(!this.story_part2.paused) {
      this.story_part2.pause();
      this.story_part2.currentTime = 0;
     }
     if(!this.story_part3.paused) {
      this.story_part3.pause();
      this.story_part3.currentTime = 0;
     }
     this[storyAudio].play();
    }, 1000);
   } else if (storyPart === 'part2') {
    setTimeout(() => {
     this.part1 = false;
     this.part2 = true;
     this.part3 = false;
```

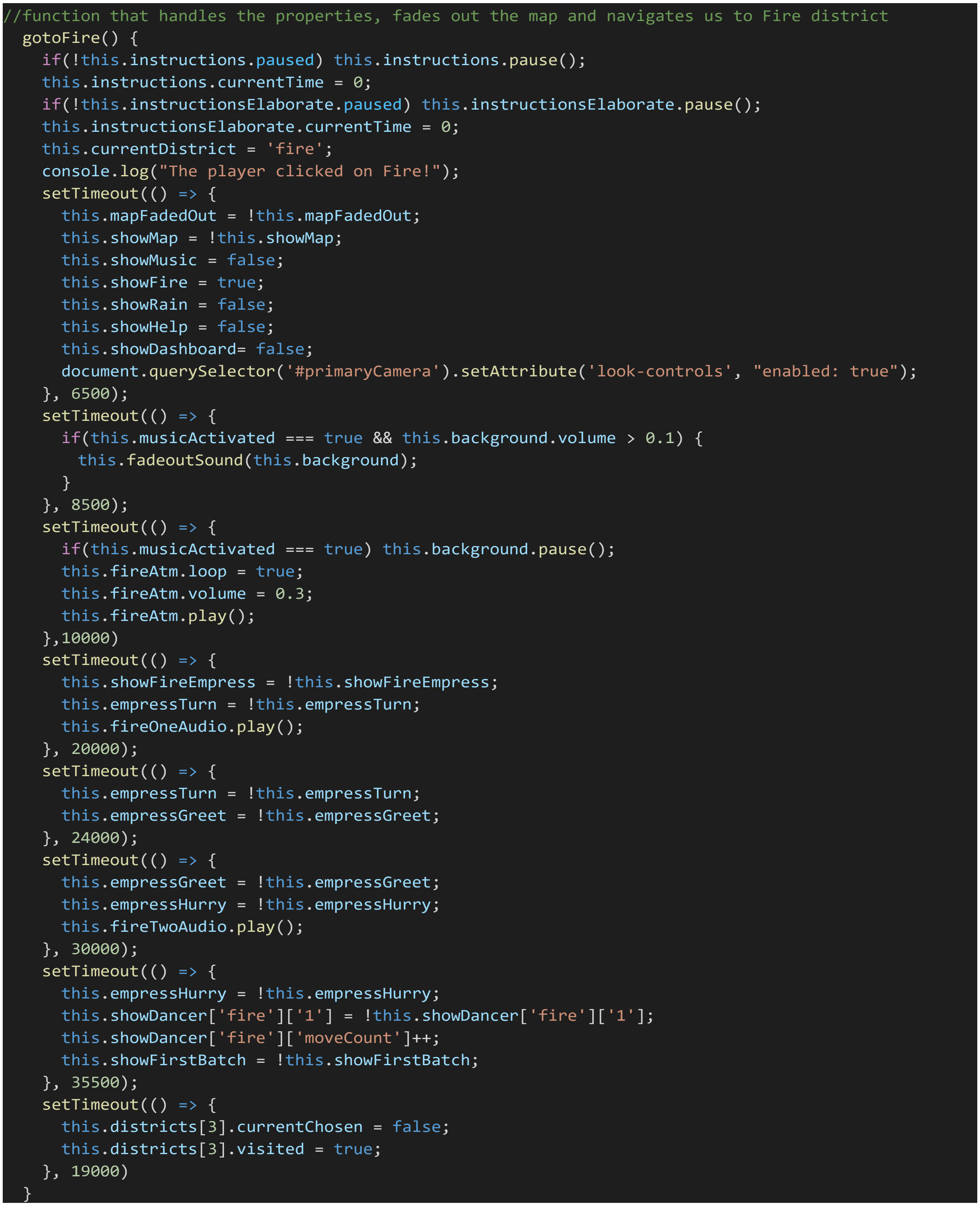


```
          this.part2 = true;
          this.part3 = false;
          if(this.musicActivated === true && this.background.volume > 0.1) {
            this.fadeoutSound(this.background);
          }
          if(!this.story_part1.paused) {
            this.story_part1.pause();
            this.story_part1.currentTime = 0;
          }
          if(!this.story_part3.paused) {
            this.story_part3.pause();
            this.story_part3.currentTime = 0;
          }
          this[storyAudio].play();

        }, 1000);

      } else if (storyPart === 'part3') {
        setTimeout(() => {
          this.part1 = false;
          this.part2 = false;
          this.part3 = true;
          if(this.musicActivated === true && this.background.volume > 0.1) {
            this.fadeoutSound(this.background);
          }
          if(!this.story_part2.paused) {
            this.story_part2.pause();
            this.story_part2.currentTime = 0;
          }
          if(!this.story_part1.paused) {
            this.story_part1.pause();
            this.story_part1.currentTime = 0;
          }
          this[storyAudio].play();
        }, 1000);
      } else if (storyPart === 'bookCover') {
        console.log("The player returned to the cover the book!");

        if(this.musicActivated === true && this.background.volume === 0.1) {
          this.fadeinSound(this.background);
        }
        console.log("story ", this.background.volume);
        this.part1 = false;
        this.part2 = false;
        this.part3 = false;
        this.story_part1.pause();
        this.story_part1.currentTime = 0;
        this.story_part2.pause();
        this.story_part2.currentTime = 0;
        this.story_part3.pause();
        this.story_part3.currentTime = 0;
      }
    } else if (storyPart === '') {
      setTimeout(() => {
        console.log("The player finished the book!");
        if(this.musicActivated === true && Number(this.background.volume.toPrecision(2)) === 0.10) {
```



```
          this.fadeinSound(this.background);
          console.log(this.background.volume);
        }
        console.log("story ", this.background.volume)
        this.bookRead = true;
        this.part1 = false;
        this.part2 = false;
        this.part3 = false;
        this.story_part1.pause();
        this.story_part1.currentTime = 0;
        this.story_part2.pause();
        this.story_part2.currentTime = 0;
        this.story_part3.pause();
        this.story_part3.currentTime = 0;
      });
    }

  }
```

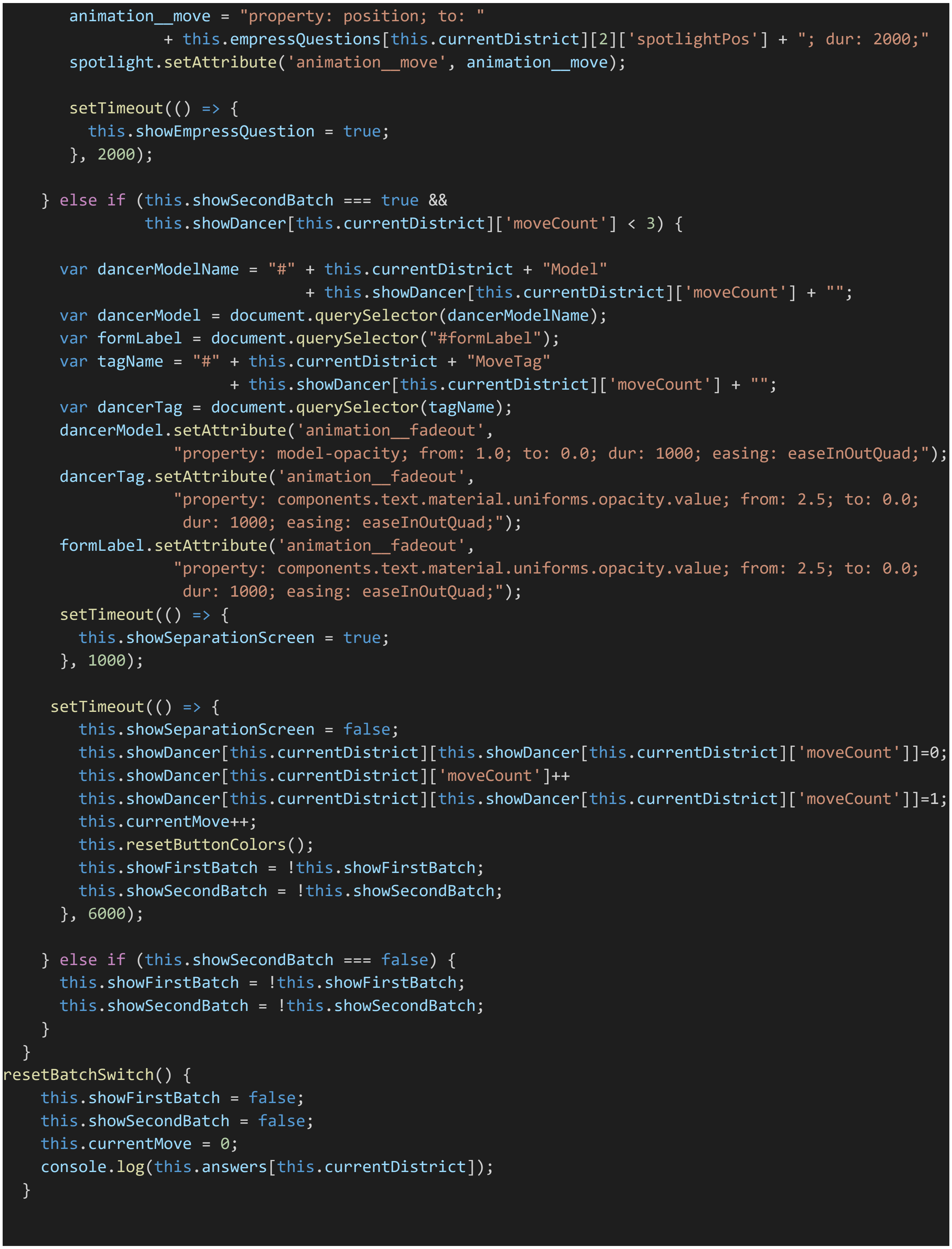
## Function called when the user chooses to travel to the Fire District



```
//function that handles the properties, fades out the map and navigates us to Fire district
  gotoFire() {
    if(!this.instructions.paused) this.instructions.pause();
    this.instructions.currentTime = 0;
    if(!this.instructionsElaborate.paused) this.instructionsElaborate.pause();
    this.instructionsElaborate.currentTime = 0;
    this.currentDistrict = 'fire';
    console.log("The player clicked on Fire!");
    setTimeout(() => {
      this.mapFadedOut = !this.mapFadedOut;
      this.showMap = !this.showMap;
      this.showMusic = false;
      this.showFire = true;
      this.showRain = false;
      this.showHelp = false;
      this.showDashboard= false;
      document.querySelector('#primaryCamera').setAttribute('look-controls', "enabled: true");
    }, 6500);
    setTimeout(() => {
      if(this.musicActivated === true && this.background.volume > 0.1) {
        this.fadeoutSound(this.background);
      }
    }, 8500);
    setTimeout(() => {
      if(this.musicActivated === true) this.background.pause();
      this.fireAtm.loop = true;
      this.fireAtm.volume = 0.3;
      this.fireAtm.play();
    },10000)
    setTimeout(() => {
      this.showFireEmpress = !this.showFireEmpress;
      this.empressTurn = !this.empressTurn;
      this.fireOneAudio.play();
    }, 20000);
    setTimeout(() => {
      this.empressTurn = !this.empressTurn;
      this.empressGreet = !this.empressGreet;
    }, 24000);
    setTimeout(() => {
      this.empressGreet = !this.empressGreet;
      this.empressHurry = !this.empressHurry;
      this.fireTwoAudio.play();
    }, 30000);
    setTimeout(() => {
      this.empressHurry = !this.empressHurry;
      this.showDancer['fire']['1'] = !this.showDancer['fire']['1'];
      this.showDancer['fire']['moveCount']++;
      this.showFirstBatch = !this.showFirstBatch;
    }, 35500);
    setTimeout(() => {
      this.districts[3].currentChosen = false;
      this.districts[3].visited = true;
    }, 19000)
  }
```

## Functions called when the user has requested to next or previous set of descriptions during the annotation process

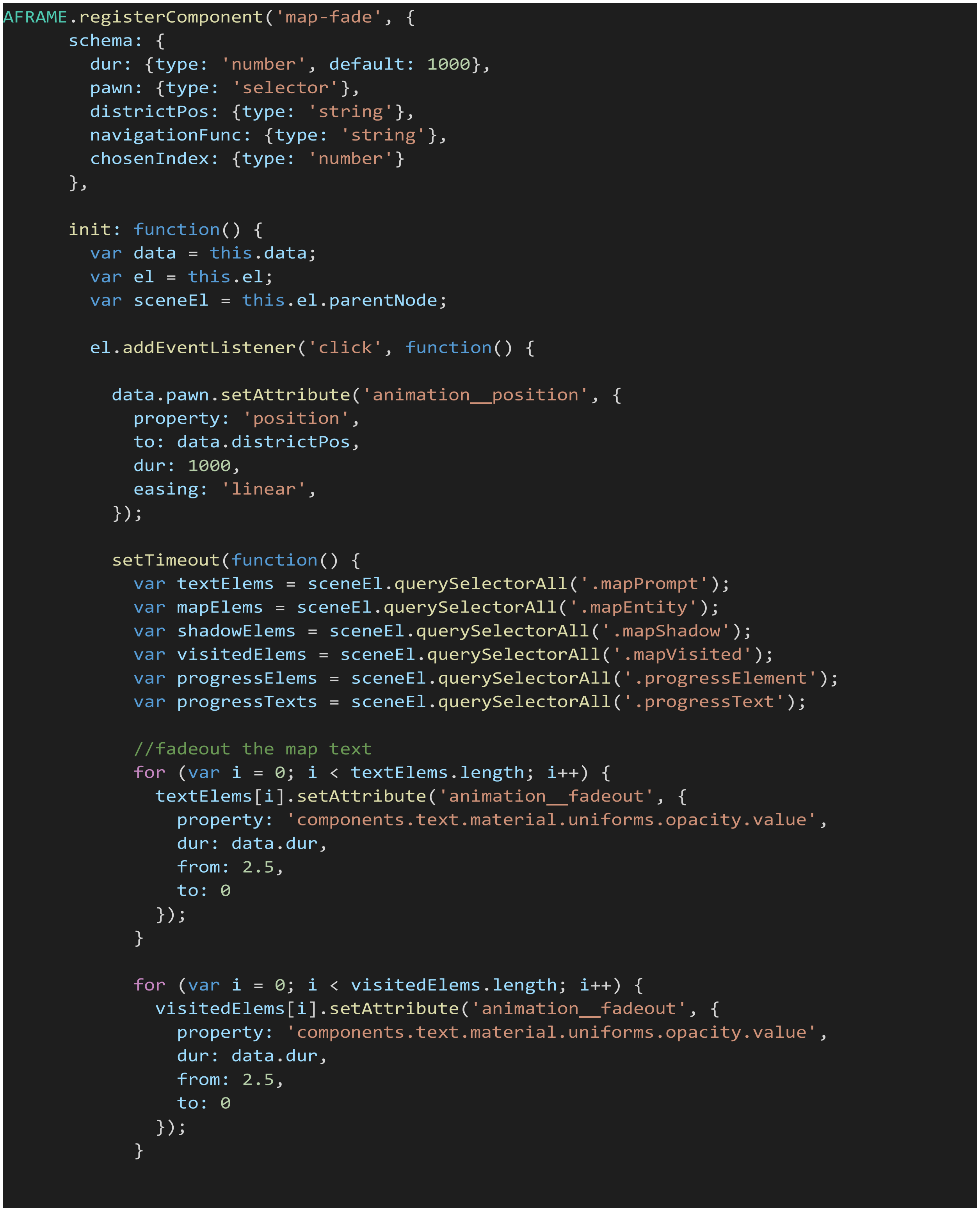
```
//Function that changes the batch of questions askes on a dance move and determines
  whether or not it's time to change the move shown
  switchBatch(back: boolean) {
    if (back === true) {
      this.showFirstBatch = !this.showFirstBatch;
      this.showSecondBatch = !this.showSecondBatch;
      return;
    }
    if(this.showDancer[this.currentDistrict]['moveCount'] === 3 &&
       this.showSecondBatch === true) {
       var doneButtonText = document.querySelector('#doneText');
       var doneButtonElem = document.querySelector('#questionsDoneButton');
       if (doneButtonText != null) doneButtonText.setAttribute('animation__fadeout',
          "property: components.text.material.uniforms.opacity.value; from: 2.5;
           to: 0; dur: 500;");
       if (doneButtonElem != null) doneButtonElem.setAttribute('animation__fadeout',
          "property: material.opacity; from: 1; to: 0; dur: 500;");
       var backButtonText = document.querySelector('#backText');
       var backButtonElem = document.querySelector('#questionsBackButton');
       if (backButtonText != null) backButtonText.setAttribute('animation__fadeout',
          "property: components.text.material.uniforms.opacity.value; from: 2.5;
           to: 0; dur: 500;");
       if (backButtonText != null) backButtonElem.setAttribute('animation__fadeout',
          "property: material.opacity; from: 1; to: 0; dur: 500;");
       var nextButtonText = document.querySelector('#nextText');
       var nextButtonElem = document.querySelector('#questionsNextButton');
       if (nextButtonText != null) nextButtonText.setAttribute('animation__fadeout',
          "property: components.text.material.uniforms.opacity.value; from: 2.5;
           to: 0; dur: 500;");
       if (nextButtonText != null) nextButtonText.setAttribute('animation__fadeout',
           "property: material.opacity; from: 1; to: 0; dur: 500;");

       //Reset the batch switches
       this.resetBatchSwitch();
       this.resetButtonColors();

       //move the dancer and the dance mat
       var dancerModelName = "#" + this.currentDistrict + "Model"
                               + this.showDancer[this.currentDistrict]['moveCount'] + "";
       var spotlightName = "#" + this.currentDistrict + "SpotLight";
       var dancerModel = document.querySelector(dancerModelName);
       var spotlight = document.querySelector(spotlightName);
       var tagName = "#" + this.currentDistrict + "MoveTag"
                               + this.showDancer[this.currentDistrict]['moveCount'] + "";
       var dancerTag = document.querySelector(tagName);
       dancerTag.setAttribute('animation__fadeout',
             "property: components.text.material.uniforms.opacity.value; from: 2.5;
              to: 0.0; dur: 1000; easing: easeInOutQuad;");
       var animation__move = "property: position; to: "
           + this.empressQuestions[this.currentDistrict][2]['modelPos'] + "; dur: 2000;"
       dancerModel.setAttribute('animation__move', animation__move);
```



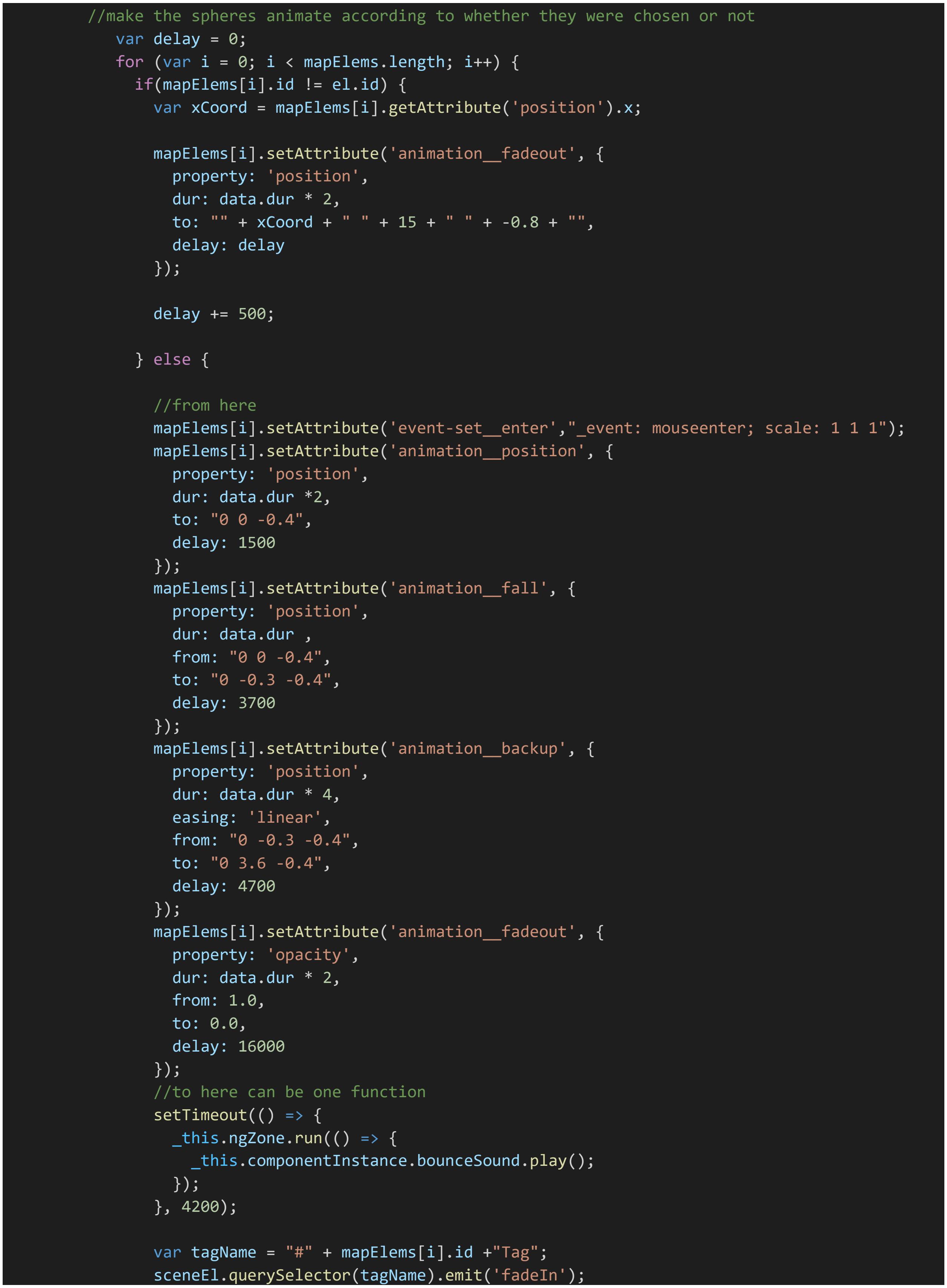
```
       animation__move = "property: position; to: "
                 + this.empressQuestions[this.currentDistrict][2]['spotlightPos'] + "; dur: 2000;"
       spotlight.setAttribute('animation__move', animation__move);

       setTimeout(() => {
         this.showEmpressQuestion = true;
       }, 2000);

    } else if (this.showSecondBatch === true &&
               this.showDancer[this.currentDistrict]['moveCount'] < 3) {

      var dancerModelName = "#" + this.currentDistrict + "Model"
                                + this.showDancer[this.currentDistrict]['moveCount'] + "";
      var dancerModel = document.querySelector(dancerModelName);
      var formLabel = document.querySelector("#formLabel");
      var tagName = "#" + this.currentDistrict + "MoveTag"
                        + this.showDancer[this.currentDistrict]['moveCount'] + "";
      var dancerTag = document.querySelector(tagName);
      dancerModel.setAttribute('animation__fadeout',
                  "property: model-opacity; from: 1.0; to: 0.0; dur: 1000; easing: easeInOutQuad;");
      dancerTag.setAttribute('animation__fadeout',
                  "property: components.text.material.uniforms.opacity.value; from: 2.5; to: 0.0;
                   dur: 1000; easing: easeInOutQuad;");
      formLabel.setAttribute('animation__fadeout',
                  "property: components.text.material.uniforms.opacity.value; from: 2.5; to: 0.0;
                   dur: 1000; easing: easeInOutQuad;");
      setTimeout(() => {
        this.showSeparationScreen = true;
      }, 1000);

     setTimeout(() => {
        this.showSeparationScreen = false;
        this.showDancer[this.currentDistrict][this.showDancer[this.currentDistrict]['moveCount']]=0;
        this.showDancer[this.currentDistrict]['moveCount']++
        this.showDancer[this.currentDistrict][this.showDancer[this.currentDistrict]['moveCount']]=1;
        this.currentMove++;
        this.resetButtonColors();
        this.showFirstBatch = !this.showFirstBatch;
        this.showSecondBatch = !this.showSecondBatch;
      }, 6000);

    } else if (this.showSecondBatch === false) {
      this.showFirstBatch = !this.showFirstBatch;
      this.showSecondBatch = !this.showSecondBatch;
    }
  }
resetBatchSwitch() {
    this.showFirstBatch = false;
    this.showSecondBatch = false;
    this.currentMove = 0;
    console.log(this.answers[this.currentDistrict]);
  }
```

## Sample code from the Motion Hollow project, file game.service.ts

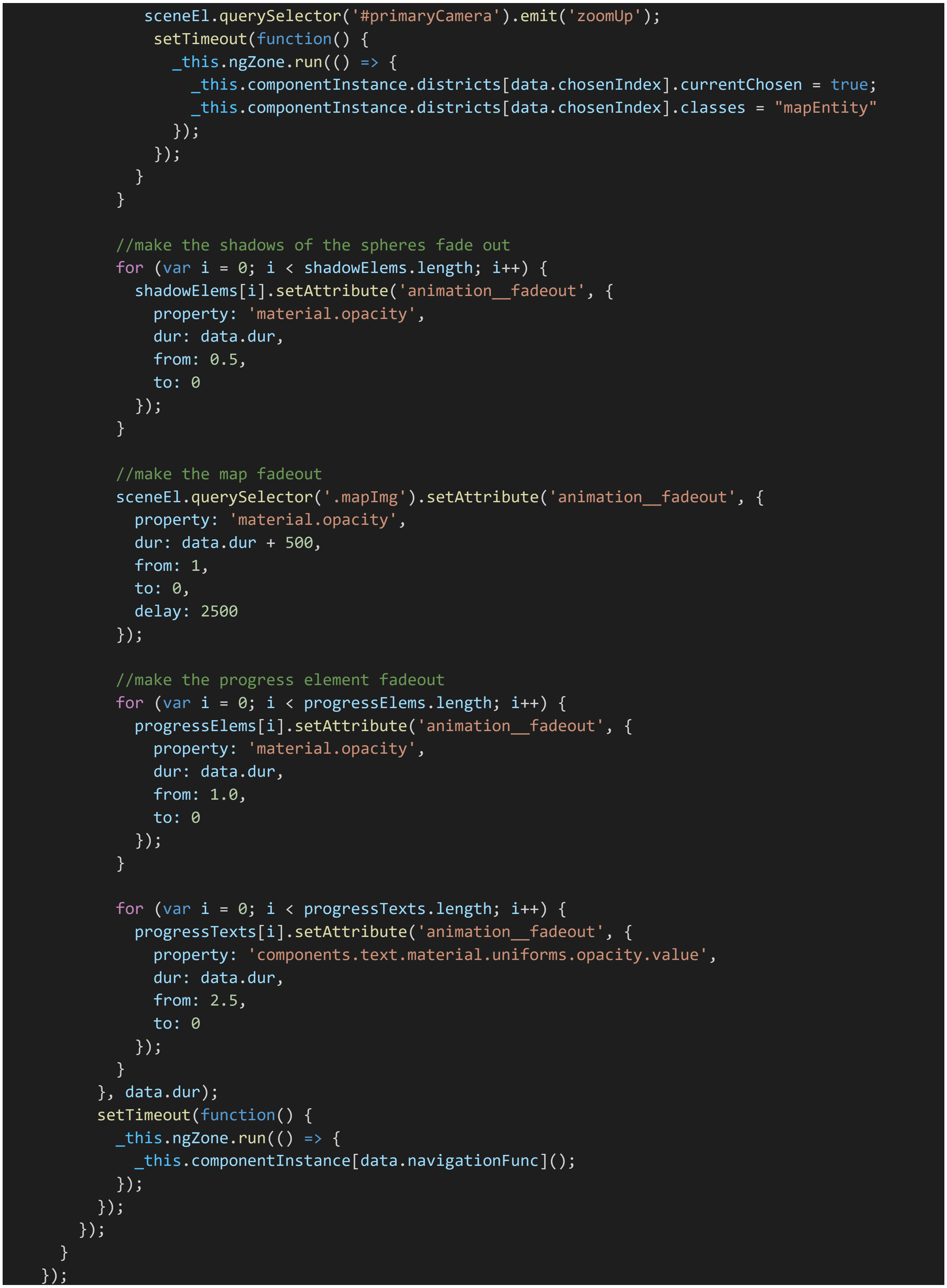
## A-Frame component that fades out the map entity when the user has chosen a district



```
AFRAME.registerComponent('map-fade', {
      schema: {
        dur: {type: 'number', default: 1000},
        pawn: {type: 'selector'},
        districtPos: {type: 'string'},
        navigationFunc: {type: 'string'},
        chosenIndex: {type: 'number'}
      },

      init: function() {
        var data = this.data;
        var el = this.el;
        var sceneEl = this.el.parentNode;

        el.addEventListener('click', function() {

          data.pawn.setAttribute('animation__position', {
            property: 'position',
            to: data.districtPos,
            dur: 1000,
            easing: 'linear',
          });

          setTimeout(function() {
            var textElems = sceneEl.querySelectorAll('.mapPrompt');
            var mapElems = sceneEl.querySelectorAll('.mapEntity');
            var shadowElems = sceneEl.querySelectorAll('.mapShadow');
            var visitedElems = sceneEl.querySelectorAll('.mapVisited');
            var progressElems = sceneEl.querySelectorAll('.progressElement');
            var progressTexts = sceneEl.querySelectorAll('.progressText');

            //fadeout the map text
            for (var i = 0; i < textElems.length; i++) {
              textElems[i].setAttribute('animation__fadeout', {
                property: 'components.text.material.uniforms.opacity.value',
                dur: data.dur,
                from: 2.5,
                to: 0
              });
            }

            for (var i = 0; i < visitedElems.length; i++) {
              visitedElems[i].setAttribute('animation__fadeout', {
                property: 'components.text.material.uniforms.opacity.value',
                dur: data.dur,
                from: 2.5,
                to: 0
              });
            }
```



```
//make the spheres animate according to whether they were chosen or not
   var delay = 0;
   for (var i = 0; i < mapElems.length; i++) {
     if(mapElems[i].id != el.id) {
       var xCoord = mapElems[i].getAttribute('position').x;

       mapElems[i].setAttribute('animation__fadeout', {
         property: 'position',
         dur: data.dur * 2,
         to: "" + xCoord + " " + 15 + " " + -0.8 + "",
         delay: delay
       });

       delay += 500;

     } else {

       //from here
       mapElems[i].setAttribute('event-set__enter',"_event: mouseenter; scale: 1 1 1");
       mapElems[i].setAttribute('animation__position', {
         property: 'position',
         dur: data.dur *2,
         to: "0 0 -0.4",
         delay: 1500
       });
       mapElems[i].setAttribute('animation__fall', {
         property: 'position',
         dur: data.dur ,
         from: "0 0 -0.4",
         to: "0 -0.3 -0.4",
         delay: 3700
       });
       mapElems[i].setAttribute('animation__backup', {
         property: 'position',
         dur: data.dur * 4,
         easing: 'linear',
         from: "0 -0.3 -0.4",
         to: "0 3.6 -0.4",
         delay: 4700
       });
       mapElems[i].setAttribute('animation__fadeout', {
         property: 'opacity',
         dur: data.dur * 2,
         from: 1.0,
         to: 0.0,
         delay: 16000
       });
       //to here can be one function
       setTimeout(() => {
         _this.ngZone.run(() => {
           _this.componentInstance.bounceSound.play();
         });
       }, 4200);

       var tagName = "#" + mapElems[i].id +"Tag";
       sceneEl.querySelector(tagName).emit('fadeIn');
```



```
           sceneEl.querySelector('#primaryCamera').emit('zoomUp');
            setTimeout(function() {
              _this.ngZone.run(() => {
                _this.componentInstance.districts[data.chosenIndex].currentChosen = true;
                _this.componentInstance.districts[data.chosenIndex].classes = "mapEntity"
              });
            });
          }
        }

        //make the shadows of the spheres fade out
        for (var i = 0; i < shadowElems.length; i++) {
          shadowElems[i].setAttribute('animation__fadeout', {
            property: 'material.opacity',
            dur: data.dur,
            from: 0.5,
            to: 0
          });
        }

        //make the map fadeout
        sceneEl.querySelector('.mapImg').setAttribute('animation__fadeout', {
          property: 'material.opacity',
          dur: data.dur + 500,
          from: 1,
          to: 0,
          delay: 2500
        });

        //make the progress element fadeout
        for (var i = 0; i < progressElems.length; i++) {
          progressElems[i].setAttribute('animation__fadeout', {
            property: 'material.opacity',
            dur: data.dur,
            from: 1.0,
            to: 0
          });
        }

        for (var i = 0; i < progressTexts.length; i++) {
          progressTexts[i].setAttribute('animation__fadeout', {
            property: 'components.text.material.uniforms.opacity.value',
            dur: data.dur,
            from: 2.5,
            to: 0
          });
        }
      }, data.dur);
      setTimeout(function() {
        _this.ngZone.run(() => {
          _this.componentInstance[data.navigationFunc]();
        });
      });
    });
  }
});
```

# REFERENCES

[1] A. Camurri *et al.*, “The dancer in the eye: Towards a multi-layered computational framework of qualities in movement,” in *ACM International Conference Proceeding Series*, 2016, vol. 05-06-July.

[2] G. Tsampounaris, K. El Raheb, V. Katifori, and Y. Ioannidis, “Exploring visualizations in real-time motion capture for dance education,” in *ACM International Conference Proceeding Series*, 2016.

[3] S. Fdili Alaoui, B. Caramiaux, M. Serrano, and F. Bevilacqua, “Movement Qualities as Interaction Modality,” 2012.

[4] Raheb, K. E., Tsampounaris, G., Katifori, A., & Ioannidis, Y. (2018, May). Choreomorphy: A whole-body interaction experience for dance improvisation and visual experimentation. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces* (pp. 1-9).

[5] K. El Raheb, S. Whatley, and A. Camurri, “A conceptual framework for creating and analyzing dance learning digital content,” in *ACM International Conference Proceeding Series*, 2018, pp. 1–8.

[6] J. Chadabe, “The Limitations of Mapping as a Structural Descriptive in Electronic Instruments mapping fly-by-wire algorithmic network in-teractivity instrument deterministic indetermin-istic.”

[7] L. Bishko, “ANIMATION PRINCIPLES AND LABAN MOVEMENT ANALYSIS: MOVEMENT FRAMEWORKS FOR CREATING EMPATHIC CHARACTER PERFORMANCES.”

[8] S. F. Alaoui, K. Carlson, S. Cuykendall, K. Bradley, K. Studd, and T. Schiphorst, “How do experts observe movement?,” in *ACM International Conference Proceeding Series*, 2015, vol. 14-15-Augu, pp. 84–91.

[9] R. Masu, N. N. Correia, S. Jurgens, I. Druzetic, and W. Primett, “How do Dancers Want to Use Interactive Technology? Appropriation and Layers of Meaning Beyond Traditional Movement Mapping ACM Reference format,” 2019.

[10] “The Eight Efforts: Laban Movement.” [Online]. Available: https://www.theatrefolk.com/blog/the-eight-efforts-laban-movement/. [Accessed: 26-Feb-2020].

[11] “Laban Movement Types | Sin Aesthetics.” [Online]. Available: http://games.spaceanddeath.com/sin_aesthetics/102. [Accessed: 26-Feb-2020].

[12] P. Subyen, K. Carlson, D. Maranan, and T. Schiphorst, “Recognizing Movement Qualities : Mapping LMA Effort Factors to Visualization of Movement,” in *Creativity and Cognition*, 2013.

[13] “Laban Examples Characteristics of the Eight Laban Effort-Actions.”

[14] A. Amory, K. Naicker, J. Vincent, and C. Adams, “The use of computer games as an educational tool: Identification of appropriate game types and game elements,” *Br. J. Educ. Technol.*, vol. 30, no. 4, pp. 311–321, Oct. 1999.

[15] A. Alexiou and M. C. Schippers, “Digital game elements, user experience and learning: A conceptual framework,” *Educ. Inf. Technol.*, vol. 23, no. 6, pp. 2545–2567, Nov. 2018.

[16] M. Trajkova and F. Cafaro, “E-Ballet: Designing for remote ballet learning,” in *UbiComp 2016 Adjunct - Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016.

[17] R. B. Dannenberg, E. L. M. Law, L. Von Ahn, and M. Crawford, “TagATune: A Game for Music and Sound Annotation. The Piano Tutor View project TAGATUNE: A GAME FOR MUSIC AND SOUND ANNOTATION,” 2007.

[18] “Laban Analysis Reviews.” [Online]. Available: http://www.laban-

analyses.org/laban_analysis_reviews/laban_analysis_notation/effort_dynamics_eukinetics/element_factor_state_drive.htm. [Accessed: 31-Aug-2020].

[19] “Space Harmony - Wikipedia.” [Online]. Available: https://en.wikipedia.org/wiki/Space_Harmony. [Accessed: 31-Aug-2020].

[20] C. Whittier, “Laban Movement Analysis Approach to Classical Ballet Pedagogy,” *J. Danc. Educ.*, vol. 6, no. 4, pp. 124–132, Oct. 2006.

[21] E. Groff, “Laban Movement Analysis: Charting the Ineffable Domain of human Movement,” *J. Phys. Educ. Recreat. Danc.*, vol. 66, no. 2, pp. 27–30, Feb. 1995.

[22] J. Newlove and J. Dalby, “Laban for all,” 2004.

[23] “Dance Imagery for Technique and Performance, Second Edition - Eric N. Franklin - Google Books.” [Accessed: 28-Sep-2020].

[24] “Theory - EUROLAB e.V.” [Online]. Available: https://www.laban-eurolab.org/lbms/theory/. [Accessed: 29-Sep-2020].

[25] “Drama Teacher AcAdemy Laban’s Eight Efforts,” 2015.

[26] “A-Frame – Make WebVR.” [Online]. Available: https://aframe.io/. [Accessed: 29-Sep-2020].

[27] A. Wodehouse and M. Sheridan, “Exploring emotional response to gesture in product interaction using Laban’s movement analysis,” *Interact. Stud. Stud. Soc. Behav. Commun. Biol. Artif. Syst.*, vol. 15, no. 2, pp. 321–342, 2014.

[28] “Body - Space - Expression: The Development of Rudolf Laban’s Movement and ... - Vera Maletic - Google Books.” [Accessed: 02-Oct-2020].

[29] “Kinesphere | Space and relationship.” [Online]. Available: https://thespaceintherelationship.wordpress.com/kinesphere/. [Accessed: 02-Oct-2020].

[30] M. I. Mandel and D. P. W. Ellis, “A Web-Based Game for Collecting Music Metadata,” *J. New Music Res.*, vol. 37, no. 2, pp. 151–165, Jun. 2008.

[31] D. Turnbull, R. Liu, L. Barrington, and G. Lanckriet, “A GAME-BASED APPROACH FOR COLLECTING SEMANTIC ANNOTATIONS OF MUSIC.”

[32] Y. E. Kim, E. Schmidt, and L. Emelle, “MOODSWINGS: A COLLABORATIVE GAME FOR MUSIC MOOD LABEL COLLECTION.”

[33] “Learning Game Design: Game Elements | Knowledge Guru.” [Online]. Available: http://www.theknowledgeguru.com/learning-game-design-game-elements/. [Accessed: 10-Oct-2020].

[34] S. Laato, S. M. Hyrynsalmi, and M. Paloheimo, “Online multiplayer games for crowdsourcing the development of digital assets: The case of ingress,” in *Lecture Notes in Business Information Processing*, 2019, vol. 370 LNBIP, pp. 387–401.

[35] “What is ‘Look and Feel’ of a website exactly? - Remember Creative.” [Online]. Available: https://remembercreative.com/2016/03/21/look-and-feel-of-a-website/. [Accessed: 10-Oct-2020].

[36] S. Garner and D. McDonagh- Philp, “Problem Interpretation and Resolution via Visual Stimuli: The Use of ‘Mood Boards’ in Design Education,” *J. Art Des. Educ.*, vol. 20, no. 1, pp. 57–64, Feb. 2001.

[37] “Angular.” [Online]. Available: https://angular.io/. [Accessed: 11-Oct-2020].

[38] “Express - Node.js web application framework.” [Online]. Available: https://expressjs.com/. [Accessed: 11-Oct-2020].

[39] “The most popular database for modern apps | MongoDB.” [Online]. Available: https://www.mongodb.com/. [Accessed: 11-Oct-2020].

[40] “Node.js.” [Online]. Available: https://nodejs.org/en/. [Accessed: 11-Oct-2020].

[41] “Autodesk Maya - Wikipedia.” [Online]. Available: https://en.wikipedia.org/wiki/Autodesk_Maya. [Accessed: 11-Oct-2020].

[42] “Video game walkthrough - Wikipedia.” [Online]. Available: https://en.wikipedia.org/wiki/Video_game_walkthrough. [Accessed: 12-Oct-2020].

[43] “Wholodance website – Official.” [Online]. Available: http://www.wholodance.eu/. [Accessed: 15-Oct-2020].

[44] “three.js – JavaScript 3D library.” [Online]. Available: https://threejs.org/. [Accessed: 16-Oct-2020].

[45] “Angular Material UI component library.” [Online]. Available: https://material.angular.io/. [Accessed: 16-Oct-2020].

[46] A. Katifori, K. El Raheb, A. Kasomoulis, M. Rezkalla, and Y. Ioannidis, “A Web-based system for annotation of dance multimodal recordings by dance practitioners and experts” 2018. [Accessed: 16-Oct-2020].