Visualization of Player Trajectory, Eye Movement and Emotional Response Data in the Context of Play

Daniel Kepplinger

MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

Interactive Media

in Hagenberg

im September 2019

Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere.

Hagenberg, September 23, 2019

Daniel Kepplinger

Contents

Abstract

An essential subject in studying and analyzing player data is to estimate what the player is feeling at a specific time in the game. Creating a visualization of gameplay data like player movement, and eye movement data and combining it with emotional response data is not an easy task. That is why this thesis is about creating different visualization strategies to visualize this data in an accurate and readable way.

Information on how those visualization methods were created, as well as the idea behind it, is one major part of this work. The four different visualization methods that have been developed using the Unity engine, can be viewed at in realtime while the game is running or in editor mode. One of them is a path visualization which shows the players' movement trajectories and can also aggregate them for a different view. The second one shows how long each player looked at specific game objects by outlining those objects with a particular color depending on how long the dwell time was. The third visualization is a graph, which shows the flow of a specific emotion, which can be chosen, during the playtime. The final but most inventive display of data consists of particle clouds that are spawning directly in the game level. Each particle system corresponds to the emotion data of a specific player and is located at the exact space where the emotion occurred. Those particle systems also glow and can therefore easily be detected on first glance. Since the emotions are color-coded in the color of the particles, it is straightforward to find spots where a particular player had a specific emotional response.

The other major part is the evaluation of the visualizations by having five game development experts analyze the player data of a dozen players visualized with those visualizations. The evaluation process included a short introduction to the demo game and the tool workflow. After that, the experts had to solve specific tasks and rate the visualizations based on each task. Then they rated the visualizations a second time based on specific metrics such as readability, aesthetics, informativeness, and so on. In the end they filled out a feedback questionnaire.

Kurzfassung

Ein wesentliches Thema beim Analysieren von Spielerdaten ist es einzuschätzen, wie sich die Spieler zu einem bestimmten Zeitpunkt im Spiel fühlen. Es ist keine leichte Aufgabe eine Visualisierung von Spieldaten wie Spielerbewegungs- und Augenbewegungsdaten zu erstellen und diese mit emotionalen Reaktionsdaten zu kombinieren. In dieser Arbeit geht es darum, verschiedene Visualisierungsstrategien zu entwickeln, um diese Daten präzise und lesbar darzustellen. Ein wesentlicher Bestandteil dieser Arbeit sind Information darüber, wie diese Visualisierungsmethoden erstellt wurden und deren Nutzen.

Die vier verschiedenen Visualisierungsmethoden, die mit der Unity-Engine entwickelt wurden, können während des Spiels oder im Editor-Modus in Echtzeit angezeigt werden. Eine davon ist eine Pfadvisualisierung, die die Bewegungen der Spieler anzeigt und die option besitzt, diese in einer aggregierten Form zu betrachten. Die Zweite zeigt, wie lange jeder Spieler auf bestimmte Spielobjekte geschaut hat, indem sie diese Objekte mit einer bestimmten Farbe umrandet darstellt. Die umrandete Farbe hängt davon ab, wie lange die Objekte von den Spielern angeschaut wurden. Die dritte Visualisierung ist eine Grafik, die den Verlauf einer auswählbaren Emotion während der gesammten Spielzeit zeigt. Die letzte, aber unkonventionellste Anzeige von Daten besteht aus Partikelsystemen, die direkt im Spiellevel erscheinen. Jedes Partikelsystem entspricht den Emotionsdaten eines bestimmten Spielers und befindet sich genau an dem Ort, an dem die Emotion aufgetreten ist. Diese Partikelsysteme stechen durch ihr leuchten auf den ersten Blick hervor. Da die Emotionen in der Farbe der Partikel farbcodiert sind, ist es leicht, Stellen zu finden, an denen ein bestimmter Spieler eine bestimmte emotionale Reaktion hatte.

Der andere wichtige Bestandteil dieser Arbeit ist die Auswertung der Visualisierungen, indem fünf Spieleentwicklungsexperten die Spielerdaten von einem Dutzend Spielern analysierten, die mit diesen Visualisierungen dargestellt wurden. Der Evaluierungsprozess beinhaltete eine kurze Einführung in das Demo-Spiel und den Tool-Workflow. Danach mussten die Experten bestimmte Aufgaben lösen und die Visualisierungen basierend auf jeder Aufgabe und bestimmten Metriken wie Lesbarkeit, Ästhetik, Informativität und so weiter bewerten. Am Ende füllten Sie noch einen Feedback-Fragebogen aus.

Chapter 1

Introduction

An important topic in Games User Research (GUR) is visualizing playtest data in an effective way [16]. GUR is a field in which tools and techniques that measure and study players' behaviour are developed. Those tools should provide game developers with the necessary help to better understand player behavior and therefore help them make important design decisions [7]. Traditional methods of game testing include interviews, observation of players and players thinking aloud. These methods are useful for collecting qualitative data and information. However, they are often limited, and the results might be biased because of personal opinions [19]. One of the problems in playtesting is the sheer amount of data it can produce [11]. This data can be hard to analyze thus requiring some kind of visualization for representing the data. Existing visualizations in the field of GUR include trajectory visualizations [10], heatmaps [33] and abstract state space visualizations [14]. However those visualizations mostly focus on a single metric or data source, and combinations of different data sources are rare. The goal of this thesis is to provide a better or at least different point of view on the players' experience of a game, by adding emotional response data and eye-tracking data to the mix.

1.1 Scope of the Project

This master thesis had a lot of requirements. Because to analyze data, first, the data has to be collected or imported from somewhere. Also, to obtain the gameplay data, a game with full access to the source code is needed. In the preproduction of this thesis, finding or creating a demo game was therefore necessary. After that, a method tracking and gathering of the data had to be developed and included in the game. However, those parts were a necessity and not the focus of the actual thesis. The thesis itself focuses on the visualization of data. That means only this part of the thesis will be evaluated. The goal is to create different kinds of visualization for different types of data that work standalone and in combinations. A visualization for the player trajectory using the recorded movement data. Another visualization that focuses on displaying eye-tracking data. Also at least one visualization that displays the emotional response data of players in any shape or form. The visualizations should be able to stand on their own but also allow a combined view with the other visualizations.

1.2 Research Question and Hypothesis

This thesis claims that having the additional emotion data information combined with other data like eye-tracking and player movement provides a different view on the players' experience and may, therefore, be of assistance in level design. The main research question is:

> Which visualization strategies can be employed for visualizing players' movement, eye-tracking and emotional response data in order to enhance the design decision-making process?

1.3 Thesis Outline and Methodology

This section gives an overview of what can be expected in the following chapters of this thesis:

- *State of the Art:* This chapter will go into detail on the subject of games user research and tools and literature similar to this project.
- *Visualization Strategies:* The ideas and methods behind the different visualizations will be explained here. This includes a trajectory visualization as well as an eye-tracking and two emotion visualizations. This chapter also contains some early concepts of some visualizations.
- *Thesis Project:* Contains all of the information on the development of the project. This includes the preproduction and planning stage as well as meetings and the technical architecture. It provides details on the horror first-person shooter game that was used for this study. It will also explain how the visualizations were developed in detail and list all of the features of the resulting prototype.
- *Collecting Playtesting Data:* Listed here are the methods of how the player data was collected. It includes details on the technical structure, the participants, and the environment where the playtesting took place. The participants of the study played a demo horror game and filled out an anonymous questionnaire.
- *Expert Evaluation:* The evaluation is done by five industry experts who try to analyze the recorded playtest data by using different visualizations. The evaluation progress is divided into several steps which will be explained in detail in this chapter. The major part consists of the experts trying to solve different predefined tasks and evaluate the visualizations based on those tasks. The results of this evaluation are also listed.

Chapter 2

State of the Art

The following Sections contain information on the current state in the field of gameplay visualizations with emphasis on player movement, eye movement, and emotional responses. These Sections also mention existing works and tools in this area. Hence, this should not only provide some basic knowledge in the field but also highlight the new and innovative part of this thesis. More so, this chapter will provide a claim why this thesis has a use in the field of games user research.

2.1 Games User Research

Games User Research (GUR) concerns itself with creating a set of tools and methods to measure and analyze players' behavior, for the sake of helping developers optimize the design decision making process [7]. To summarize GUR in one sentence, one could say that its purpose is to help game developers to deliver players the best gaming experience possible. That is why GUR tools' primary use is to gather data and to get an understanding of players' decision making [19]. However, a significant amount of data has to be analyzed, which can be confusing to decipher. Thus resulting in game developers spending too much time and effort to derive usable information out of those vast collections of playtesting data. Unquestionably, simplifying the data interpretation process is of excellent importance [11]. That is why data visualization tools are a necessity in the field of GUR.

Visualizations help in understanding and exploring large data sets and are a useful tool for analyzing data [21]. They are necessary to filter and display the relevant data in higher levels of detail, to get more insight into how a player progresses through a game [19].

First of all, for this thesis, the relevant target audience is game developers. Those are probably the group of people that will have the most need for understanding player behavior. That is not to say that this tool cannot be used elsewhere. For example, some mixed reality applications are not necessarily games but rather simulations or experiences. An example of this would be looking at a hotel in Virtual Reality (VR) before booking it. Another example [30] makes it possible to look around in the interior of a yacht inside VR. However, they can be quite similar in design and on a technical basis. In those cases, this tool could provide helpful insights on how the player perceives

Figure 2.1: In this Figure of a survival shooter in *Unity* [27], game events display as 2D icons on the paths, and small arrows show the direction the player faced at a particular place [11].

those simulations and experiences. Even though there exist multiple different gameplay visualization techniques, essential to look at are the ones relevant for game developers. The focus of this thesis lies in gameplay data that occurs during the developing process, such as during playtests. For that reason, the following segment covers a state-of-the-art overview of relevant gameplay visualizations.

To further illustrate this, some examples of different gameplay visualizations of existing works are listed below.

Figure 2.1, a snippet out of a tool named *Vixen* [29], features a system to visualize player trajectories using a path visualization. This visualization also displays directional information by drawing small arrows on the lines. These arrows orient in a way as to represent the direction the in-game character was facing at each timestamp. In-game events are also displayed as 2D Icons at the corresponding location. Often a rendered path is used for displaying the visualization of the player's movement inside the game world. For example, Hoobler et al. [12] varied the thickness of the paths to depict the elapsed time. Other related works use color cycling to represent the flow of time [10]. In Figure 2.2 from Mirza-Babaei et al. [21] the color of the path the player moved, is used to show the players arousal.

2.2 Visual Analytics of Movement

An essential information of a playtest run is to know where the player was at a particular time. That information can, for example, be used to combine the movement data

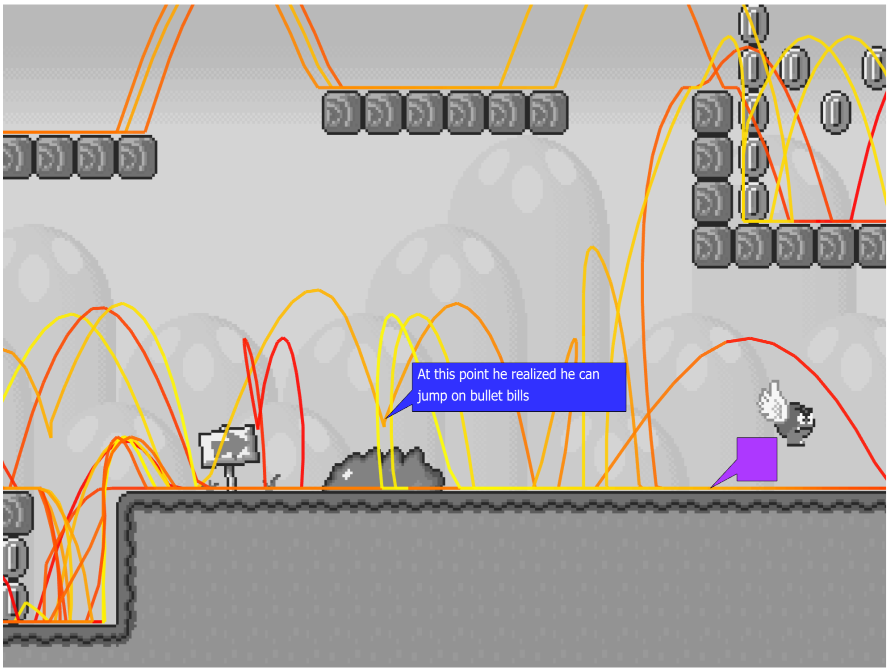


Figure 2.2: Screenshot of *Infinite Mario* a popular public domain version of *Super Mario Bros*. The color of the paths reflects the recorded galvanic skin response values. The text bubbles show player comments [21].

in the right sequence, and thus to result in the trajectory of the player's movement. Additionally, this trajectory can then be visualized as a path.

Visual analytics research is about transforming data to knowledge and information [4]. This complex topic will be narrowed down by a small margin, and only relevant issues will be mentioned, to not exceed the scope of this thesis. Two key topics in visual analytics are space and time [3].

2.2.1 Movement Data

Andrienko et al. [4] defines following properties of movement data, that are essential for analyzing:

- Temporal properties:
  - temporal resolution: depicts the lengths of the time differences between the measured positions;
  - temporal regularity: depicts if the temporal resolution is constant or variable;
  - temporal coverage: stands for the time span of the possibly multiple measurements.
- Spatial properties:
  - spatial resolution: the minimum difference in position of a particular object;
  - spatial precision: defines if the positions are points or are areas with spatial extend;

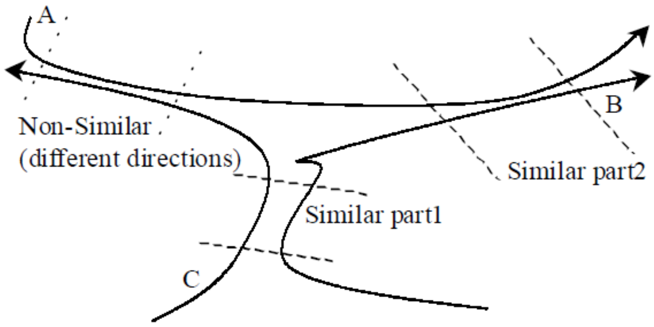


Figure 2.3: One problem with aggregation based on distance threshold are trajectories with different directions [15].

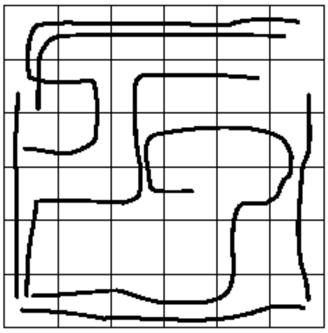
- Mover set properties:
  - number of movers: describes how many movers were measured.
- Data collection properties:
  - position exactness: are the measurement methods such as movement sensors exact?;
  - position accuracy: shows the amount of error that may be in the measurements;
  - missing positions: are there gaps in the data?

Movement data can be quite complex; however, some of those properties Andrienko et al. [4] listed, like time cycle coverage, were not deemed relevant in digital games. Thus they were not listed and will not be engulfed in deeper.

2.2.2 Trajectory Aggregation

Trajectories are hard to visualize because they have a lot of intersections and trajectories are overlapping one another. As a result, displaying them looks heavily cluttered and illegible. It is essential to abstract the data through appropriate aggregation methods [1].

The goal of aggregation is to have a single trajectory represent comparable trajectories. There exist several methods on how to handle this. It can be accomplished by aggregation based on a distance threshold, where the shortest distances between re-discretized points of two trajectories are compared. However, this method has some problems, as explained in Figure 2.3. Also, an alternative method would be spatial unit aggregation. Here the data is divided into cells, and each cell is assigned the number of times an object passes through. Figure 2.4 shows some more insight into this method. The third method is called spatio-temporal unit aggregation, and it uses the spatial data together with time. In comparison to the previous process, the number of each trajectory passing through a cell is computed in a particular time interval [15]. Therefore each path is seen as an array of moves between cells. A move is an object that contains



2	2	2	2	2	2
3	1	1	1	1	1
2	1	2	1	1	2
2	1	2	1	1	2
2	0	0	0	1	1
4	2	2	2	2	2

Figure 2.4: Aggregation based on spatial unit [15].

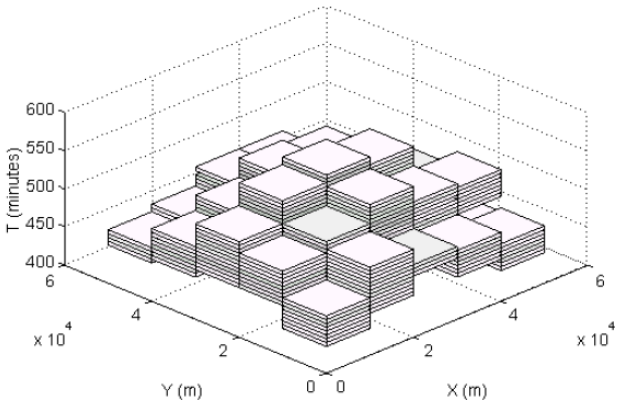


Figure 2.5: Aggregation based on spatio-temporal unit [15].

information about the place and time of the beginning and likewise the end of a move [2]. Data aggregated with this method can be visualized, as shown in Figure 2.5.

In Figure 2.6 an example of an applied aggregation is displayed.

2.3 Visual Analytics of Eye-Movement

Eye-movement is a very different issue compared to standard movement data. Most eye-tracking tools allow one to get the screen coordinates that one or both of a player's eyes are looking at. To better use this data in a 3D environment, the data has to be transformed into 3D space. This can be done by shooting rays starting at the screen position forward and checking for collisions with 3D objects. Those collision points are most likely the points in the game world, where the player did look at. Gennady Andrienko et al. [5] describe eye-tracking as . . . "The process of measuring and recording gaze positions and eye-movements of an individual." In human-computer interaction sciences, this is quite relevant for the evaluation of user interfaces and visual display.

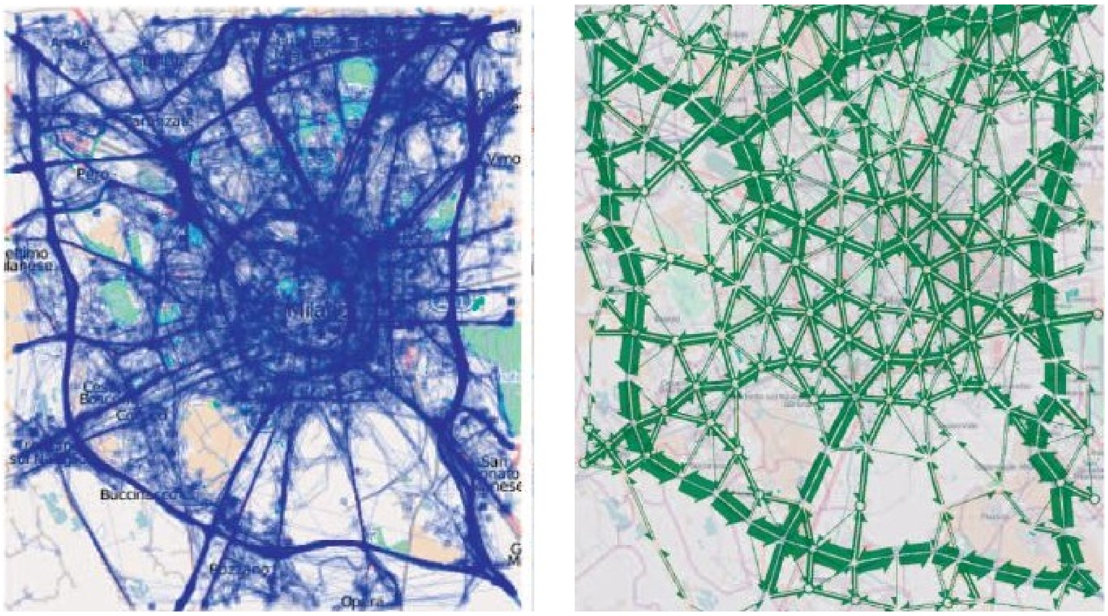


Figure 2.6: One the left are the original trajectories. On the right is an aggregated version [1].

Eye-tracking data is challenging to analyze, and the amount of data is quite high, which leads to the next point [5].

2.3.1 Visual Analytics

Nowadays, visual analysis of eye-movement data emerged to a significant field of research, which leads to a lot of unique new techniques on how to visualize said data. These new techniques allow glancing at more than what traditional methods like attention maps (Figure 2.8) and gaze plots (Figure 2.7) provide. Attention maps are heat maps, which conclusively show the weighting of all the viewing occurring. Gaze plots show the location of the fixations and the order in which they were made [31]. However, according to Michael Burch et al. [8] . . . "There is no single all-in-one visualization to solve all possible analysis tasks. The appropriate choice of a visualization technique depends on the type of data and analysis task."

2.3.2 Methods

The following list states some methods that are of use in eye movement analysis:

- Map display of trajectories,
- space-time cube display of trajectories,
- flow map of summarized eye movements,
- summary map showing users' attention by spatial distribution,
- clustering of time intervals of attention distribution by the similarity of the spatial patterns,

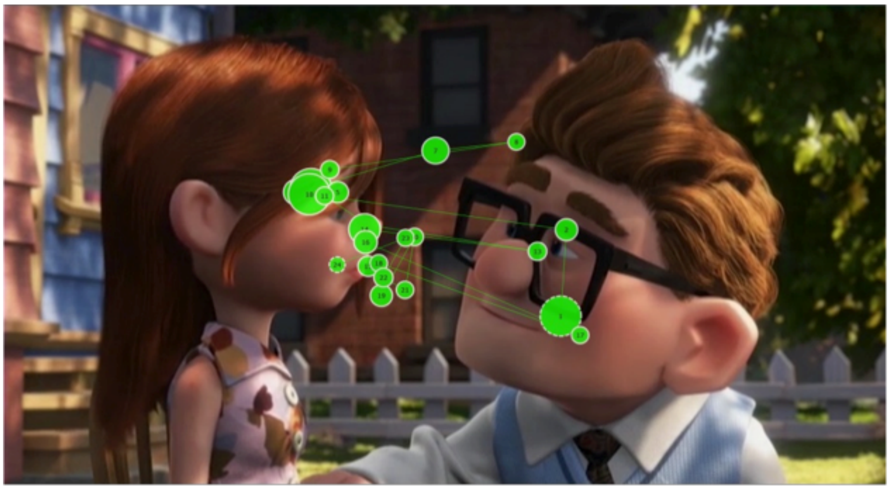


Figure 2.7: This Figure shows a gaze plot visualization [31].

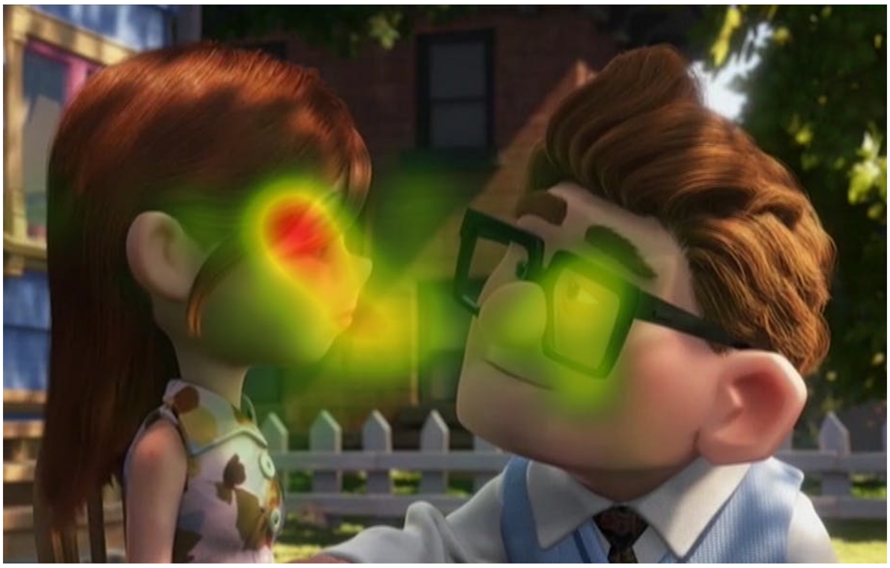


Figure 2.8: Visualized here is an attention map [31].

- filtering of trajectory segments,
- extraction of events from trajectories,
- discovery of reoccurring sequences of area visits [5].

The method that is used in this study uses the dwell time on specific objects in the game to analyse and visualize the data. These are mostly objects the player can interact with, or are important to complete the level. The objects can be chosen by the designer.

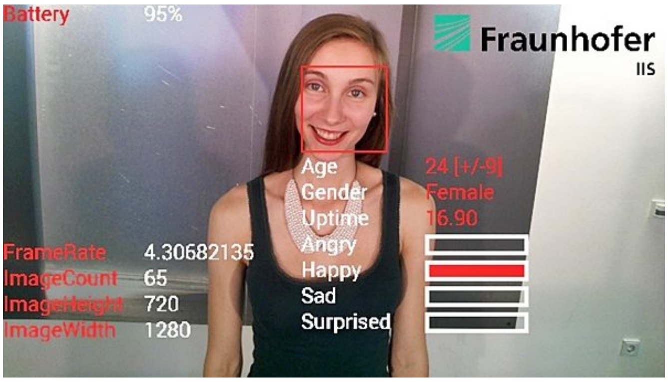


Figure 2.9: Age, gender and emotion detection of an extracted face using Shore [26].

2.4 Emotion Visualization

The process of how emotion data should be visualized has, to this day, not yet been standardized, and is therefore very experimental. The visualization method that this thesis presents is a experimental attempt of visualizing player emotions. In this section, other visualization methods, that have already been tried out will be shown and explained.

2.4.1 Extracting Emotions from Images

The origin of different facial expressions is often a result of experiencing a certain emotion. These facial expressions usually have similar patterns and can be extracted and analyzed from a picture of a person's face. So, it is also possible to conclude emotions related to that particular person. Also, the persons' gender and approximate age can be extracted, see Figure 2.9 as an example [6].

2.4.2 Visualization in Movies

Alexandre Denis et al. [9] attempted to extract emotions from movies by analyzing their scripts. They visualized the Affect of 750 movie scripts with an emotion visualization (Figure 2.10).

Those visualizations attempt to enable anyone to see the whole movie space on first sight or examine each movie individually. However, this was only preliminary work, and no evaluation has been conducted at the time of writing.

2.4.3 Plutchik's Emotion Wheel

The standard for visualizing emotions by colors is by using *Plutchik's Emotion Wheel* (Figure 2.11). Robert Plutchik [17] states that there are eight primary emotions: joy, trust, fear, surprise, sadness, anticipation, anger, and disgust. The wheel can be used to see various ways of how emotions relate to another and which emotions are opposites of each other. As a result, this helps bring clarity to emotions. Main Categories

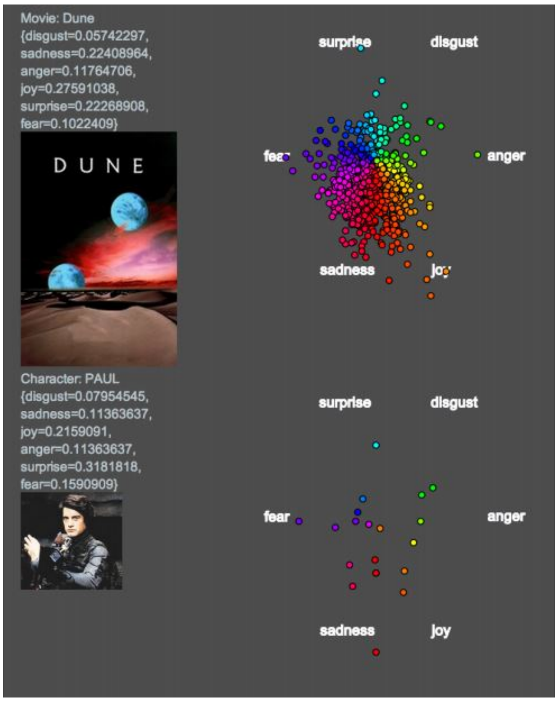


Figure 2.10: Valence visualization for 750 Movies by timeline [9].

of grouped emotions have a corresponding color assigned. This method provides an excellent standard to use in all kind of different emotion visualizations.

2.4.4 Social Media Emotions

Another way of obtaining sentiment data is by using Tweets. Siddarth Shankar Ramaswamy and Christopher G. Healey [32] made a project called Tweet Viz, that visualizes Twitter sentiment. They use word recognition and categorize those words by sentiment and topic. Then the data is visualized in a psychology diagram, clustered together by themes. The color code used is green for positive and high confidence tweets and blue for negative and low confidence tweets. Figure 2.12 provides a closer look at the topic.

2.5 Similar Tools

In Figure 2.1, an example of a GUR-Tool has already been shown. This specific tool is called *Vixen* and is probably the most similar tool when compared to the work of this thesis. *Vixen* is also using *Unity*; however, it focuses only on standard gameplay metrics like player movement and game events. The emphasis on this thesis lies on combining gameplay metrics with eye-tracking and emotional response data. Having this additional source of information on how a player feels at a particular time or while he is looking at a specific object in the game may provide game developers with new insights on player behavior. For example, just by seeing a players trajectory in a level, one might conclude

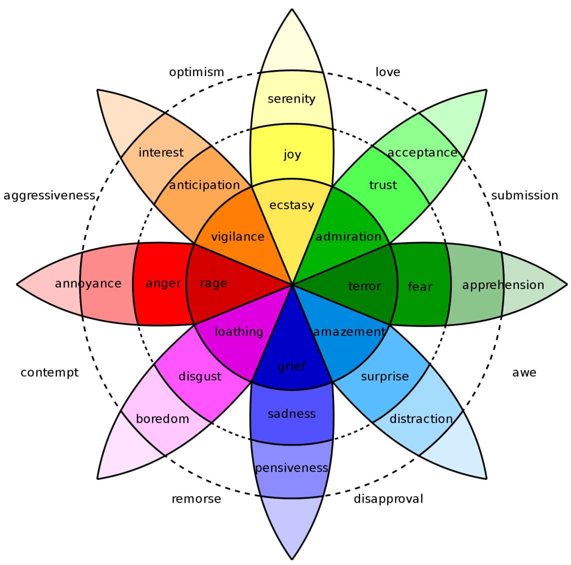


Figure 2.11: Plutchik's Emotion Wheel defined by Robert Plutchik [17].

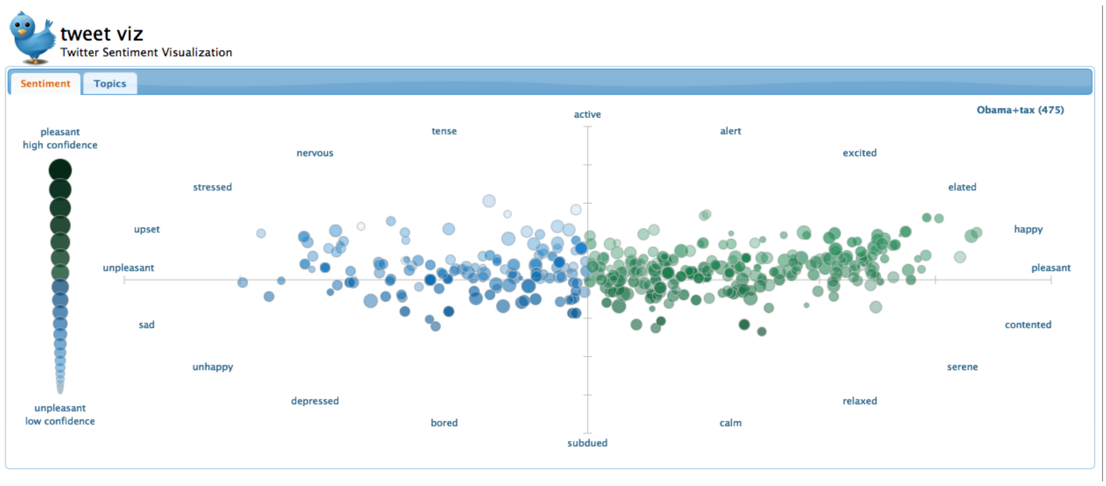


Figure 2.12: Tweet Viz query "Obama"+"tax". Copyright Siddarth Shankar Ramaswamy and Christopher G. Healey [32].

that this player went in circles a lot at a specific are and was probably having a hard time. But by having emotions as an additional information source, one might be able to detect that the player was feeling happy while doing so, and this might indicate that the player wanted to explore this part of the level more often.

Chapter 3

Visualization Strategies

In this chapter, different kind of visualization methods are presented and described in detail. The technical details on how those visualizations were made can be found in chapter 4.

3.1 Research Question

The main research question for this thesis is:

> Which visualization strategies can be employed for visualizing players' movement, eye-tracking and emotional response data in order to enhance the design decision-making process?

3.2 Visualization Methods

This section describes all the major types of visualization that were used in this project.

3.2.1 Path Visualization

Probably the first idea that comes to mind when thinking how to visualize a players movement in the game is a path visualization. Since this part of the project is nothing new, and there are already several other tools that provide this kind of data visualization, it was decided to go with the flow and not experiment too much on that topic. Still, those type of visualizations can be quite different aside from the main similarity that the path shows the players trajectory.

Path Properties

- *Color coding:* The path's color at different segments can be colorized depending on other aspects of the game like current health points of the player. During the early stages of this project, the idea was to color-code emotions on the path, but this idea was discarded later because of the particle cloud visualization that is explained later. This method was used in a *Super Mario* game by Mirza-Babaei et al. [21], where the color of the trajectory represented the players arousal.

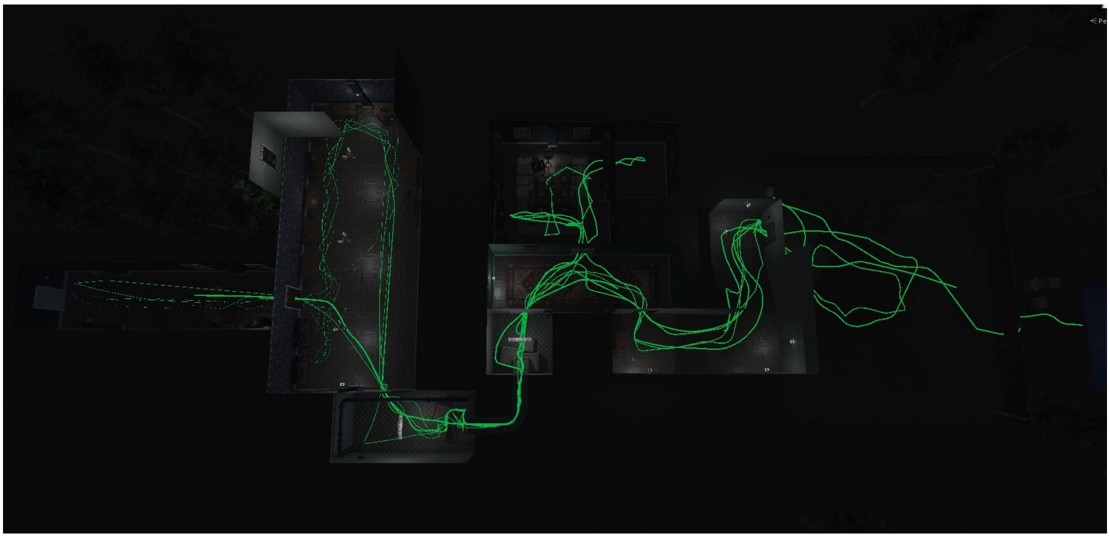


Figure 3.1: The more time flows by the thicker the green path becomes.

- *Time:* The flow of time can be visualized on a path with numbers or even color-coded. This project used the line thickness of the path to show the elapsed time as it can be seen in Figure 3.1. This method was used in a visualization on the game *Return to Castle Wolfenstein: Enemy Territory*, created by the Lithium system [13]. Since there were plans to have a time scrubbing mechanic that was not finished during the project, the visualization of time with line thickness was taken out in the latest version.

First Visualization

The first visualization that was made was a simple line path. As shown in Figure 3.2, the green lines show the player's movement and the purple lines indicate where the player's eyes were looking. As it can be seen in the Figure, the purple eye path is very chaotic and does not provide actual useful information. This is why another visualization method was chosen for the eye-tracking data.

Aggregated Visualization

When rendering multiple paths of different players at once, a lot of clustering and overlapping is taking place, and it gets harder to analyze. To solve this problem, the visualization has to be aggregated like in Figure 2.6. Aggregation means that the data will be generalized, and a single path will represent multiple similar paths. This makes the visualization a lot easier to comprehend and better to analyze the general flow of the trajectories. Smaller details, however, will be lost. There are multiple ways to aggregate trajectory data and also various ways to visualize them. The method used for this thesis will be explained later on.

Figure 3.2: The green line shows the movement of the player character. The purple lines show the trajectory of where the actual player looked.

3.2.2 Heatmap Visualization

Since another method of visualizing the eye-tracking data was required, the next idea was to use a heatmap. In heatmaps, the values of the data are represented as colors. By using a gradient that goes from green to yellow to red, so basically from cold to hot, the information on how long a player has looked at something can be coded in those colors. Mostly we wanted to know how long players looked at key game objects. That is why only those objects are rendered with a special shader that renders the object with an outline colored accordingly to the “looked at” values. The longer a player looked at an object, the “hotter” (Red) the color becomes. The effect can be seen in action in Figure 4.9.

3.2.3 Graph Visualization

There is not much new to say when it comes to graph visualizations. The one used is a basic line graph that shows the fluctuation of a specific emotion over time. Depending on the values of the emotions, the line graph also has an auto-size option to automatically chose the maximum x- and y-axis values to have the best view on the line graph. The line is also colored in the particular emotion’s color. See the visualization in Figure 6.4.

3.2.4 Particle Cloud Visualization

Probably the most interesting visualization, the particle visualization is used to display emotional response data of the players. Particle systems will spawn at the exact location where players had a strong response in a specific emotion. Those particle systems properties are also influenced heavily by the emotions data. The emotion itself decides the color and shape of the particle, and the emotion values influence the spawn rate, size, and lifetime. The stronger an emotional response occurs, the more and bigger particles will spawn. The particles are also glowing, which makes them very easy to detect. The main strength of this visualization is how fast a particular emotion can be found, thanks to the easily recognizable hot spots that are the glowing particle systems. This is shown in Figure 2.10. Also, in this Figure 4.11 it can be seen how the different particle systems for each emotion look. In the case that multiple particle systems are located on the same spot, the emitters will change the spawn angle accordingly as to not overlap with the other particles as seen in Figure 4.12. A maximum of three particle systems are allowed on the same spot.

Chapter 4

Thesis Project

The primary focus of this Chapter is to describe the development of the visualization tool. This includes all technical details as wells as meetings presentations and issues that happened during the development time. Also, the resulting prototype and all its features will be described in further detail.

4.1 Preproduction

There were a lot of prerequisites necessary for my thesis project to work. To visualize the data, it has to be collected first. Therefore it was necessary to have a demo game, which people could play while it collects their data. This demo game also had to be equipped with the ability to track emotions, eye movement, and player movement. In the end, it would be necessary to have a data collection of a lot of players, at the beginning however fake data, or data collected by the developer and other available testers were used to test the visualization.

4.1.1 Demo Game

Finding a suitable game that has access to its source code and can be upgraded by tracking tools was not an easy task. Creating the game from scratch was not an option since it would take to much time and resources and would not fit the scope of this thesis project. In terms of genre, an fps horror game was the preferred one. Players tend to be very concentrated in horror games and tend to have a variety of emotions when playing it. At least that was the assumption. *Resident Evil 4* [23] was the first significant consideration to go with, but to extract data like player position, a tool like the *Cheat Engine* [25] would be required. After debating, it was clear that a game that was made in *Unity* would be much easier to handle and also has the upside of using the tool directly in the game engine. A *Horror FPS Game Kit* [24] has been found in the *Unity Asset Store* after searching for a bit. It also already provided a demo level, which has been adjusted a little to our needs.

The demo level starts in a hallway and upon going through the first door, a big room, containing lots of different items, is entered. This room is more or less a showcase for the different items and objects this game asset provides. The next door has a different interaction method, then the first one. The player has to press and hold the E key and

Figure 4.1: When entering this room a surprising jumpscare happens. The screenshot is taken from the demo game of the Horror FPS Game Kit [24].

move the mouse to open the door. The first door opens automatically by just pressing E one time. In the next room, the player has to jump onto some boxes to go through a ventilation shaft. On the other side of the shaft, a door barricaded by wooden planks can be entered after removing the planks from the door. When going inside the jump-scare happens. At that moment, a ghost that appeared to be sitting on a couch screams and flys towards the player. This location can be seen in Figure 4.1.

The room the player just entered has a secret mechanism to unlock a secret place. This is done by interacting with one of the books on the shelf; however, it is not necessary to complete the level. Also on the table is a note that can be turned around while inspecting it. On the flip side, a secret code is displayed which has to be memorized by the player. The code shows the location and order of the right numbers on the input terminal and not the code itself. Outside of the room is a hallway that leads downstairs and ends with a closed door and a code input terminal next to it. On the way, a gun and ammunition can be picked up from a table. The player needs to enter the right code in the terminal, and then the door opens. This scenario is pictured in Figure 4.2. When going through the door, the player is finally outside in the garden. This is where the last obstacle is charging at the player. A zombie patrols the area and when noticing the player will charge at him. If the player manages to survive and kill the zombie with the gun, the game ends. If the player dies, the game is over but can be restarted.

Following adjustments have been made to the demo level:

- *End Game Condition:* The demo level had an open end with no clear goal. This is why a condition was added to end the game. The condition is killing the zombie in the garden.

Figure 4.2: After opening the door by entering the code on the input terminal the door opens up and a zombie awaits the player outside. The screenshot is taken from the demo game of the Horror FPS Game Kit [24].

- *Tag GameObjects:* The objects that the designer wants to include in the eye-tracking data have to be tagged. This is done by adding a *GazeAware* script of the eye-tracker SDK on the game object that also has a collider component attached.
- *Change Object Scale:* The outline shader used for the eye-tracking visualization needs all of the objects to be around the same local scale factor. Some of the objects in the demo scene had a scale factor of 100 and some a factor of 0.01. This is because the scale factor used for importing the mesh was already 100 or 0.01 and not consistent among the objects. By adapting the import scale factor of all objects used for the eye-tracking visualization, I was able to get the local scale of the objects to be around the same level without altering the size of their appearance.
- *Developer Hud:* More information to the ingame heads up display was added to ensure easier testing. This hud was taken out after finishing except for one aspect. This was the feature that notifies players if their face can no longer be detected through the webcam. The message disappears after the player has adjusted their head position, and the camera can detect the face again.

Finally, the demo game was available, and the focus of the project could shift to the next step.

4.1.2 Tracking Data

Collecting emotional response data is the focus of this thesis. However, it is also important to track other data like player and eye movement. That way, the combination of an emotion that occurred on a certain location when looking at a specific object can be analyzed. The tool that handles tracking and serializing data has been developed from scratch. The tracking rate has been set to ten times per second. This rate provides a reasonable accurateness while also not slowing down the performance too much. It also keeps the size of the data files low. In each of those ticks, the specific data and the current time since the level loaded are stored in a data collection. When exiting the game level, this data is then serialized into JSON files. Each group of data is stored in a different file. A player id is added to save the information of which data belongs to which player.

Movement-Tracking Module

This module handles the tracking of the player transform values. It uses the previously mentioned tracking tool to serialize the data after the player finished playing.

Eye-Tracking Module

Working in the same way as the movement tracking module, the eye-tracking module handles information of which objects have been looked at which time. The *Tobii* Eye-tracker [28] was used to measure the gaze position of the player. It includes an SDK for *Unity*, which makes it easy to access this data. The SDK provides information on which part of the screen the player is looking in pixel coordinates. With the use of raycasts that were shot straight forward into the 3D world with those positions as the origin, it can be detected if the ray intersects with an object in the game world. If that is the case, a data collection storing time values that represent how long a player looked at specific objects is updated. Raycasting is part of *Unity's* physics engine and easily accessible by user-created scripts. The items that are inspected for detection with the raycast can be chosen by the designer and is done by putting a *GazeAware* component onto the game object. At the end of the game, this data is then as well serialized in its own JSON file.

Face-Tracking Module

The face-tracking module handles information of the players' emotions at all times during the game. With the help of the *Affectiva* SDK [22] for *Unity*, it was possible to track facial expressions and gestures of the player at runtime with a webcam. The SDK also calculates basic emotions out of those facial features and provides this information for further use. The emotion data is also stored in a data collection and serialized in a separate file after the game ends.

4.2 Tools

The tool is made in *Unity* and programmed in C♯. Programming was done in Visual Studio 2017. Other helpful tools include the *Tobii* eye-tracker as well as the eye-tracking

SDK and the *Affectiva* SDK for emotion recognition with the webcam.

Most of the meetings with my project coach and other contributing individuals were held via *Skype*. We used *Google Drive* and *Dropbox* for document sharing.

4.3 Architecture

Unity is a component-based system. This is why the tool was also programmed in a component-oriented way to make the combination easier. To give functionality to a certain *GameObject*, one has to attach different components to it. To use this system to its full potential, one should be able to think in terms of components.

The tool itself is modular and not depending on the game. Since the idea behind this tool is to use it in any game that works with *Unity*, it is necessary to be easily integrated into a new project. The visualization part of the tool is separated from the tracking module. Because in the future, the methods of tracking data might change, or the user of this tool may want to switch to a different data tracking provider. As long as the data is formatted in a way the visualizer can understand it, it should be able to visualize it no matter where, when, and how the data was being tracked.

4.4 Visualization Overview

Before going into detail on the development process of the visualizations, it may be helpful to get an overview of all of them by looking at Figure 4.3. The rendering of player movement data is done with paths, which can also be transformed and looked at in an aggregated more generalized view. The eye-tracking data is visualized by outlining objects based on how long they have been looked at. The other two visualizations show information on the players' emotions. The graph shows the flow and fluctuation of a particular emotion over time. The particle clouds are rendered on the spots where certain emotions were strongly present. They are color-coded, and the properties and general look of the particles are influenced by the emotion data. Those visualizations are explained in detail in Section 4.7.

4.5 Visualization Implementation

This Section provides technical insight into how the different visualizations were implemented.

4.5.1 Trajectory Aggregation

The reason why it is often needed to aggregate trajectory data is that the more data is shown at once, the more it begins to overlap and be too cluttered to analyze. That means aggregation is important since it deals with overlapping and generalizes the data to make it easier to look at, therefore resulting in a better experience for the analyzer. Figure 4.4 shows the original paths as well as the resulting aggregation.

The term trajectory stands for the path of an object in motion. In the digital world, it can be represented as an array of positions at a specific time. The term aggregation

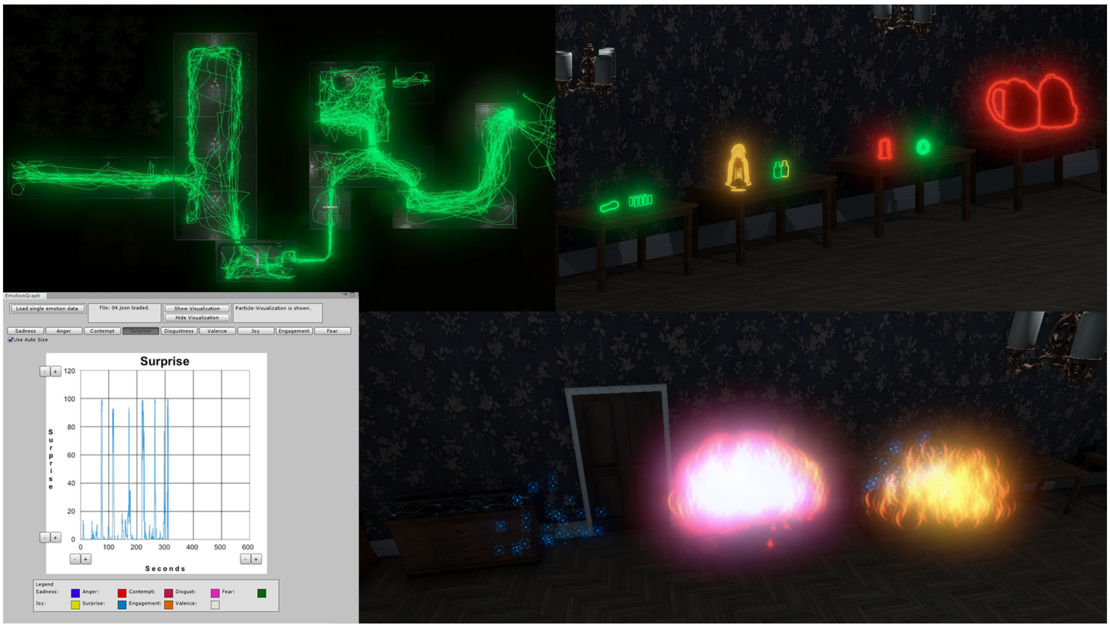


Figure 4.3: An overview of the different visualizations. Top left: Trajectory Visualiazation. Top right: Eye-tracking Visualization. Bottom left: Emotion graph. Bottom left: Emotion particle visualization.

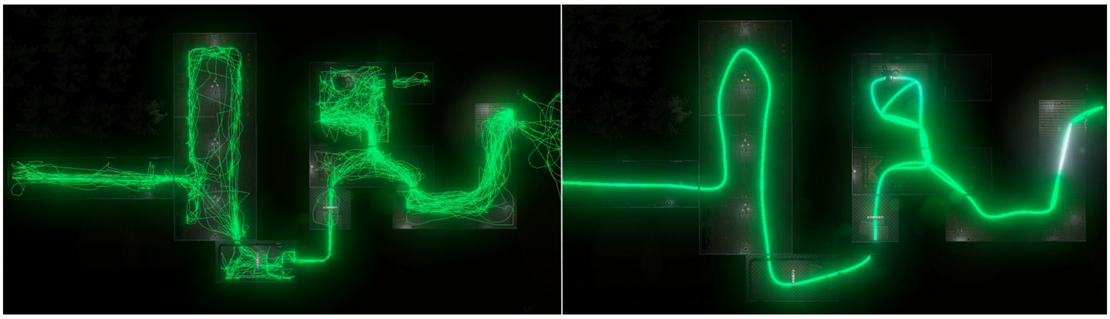


Figure 4.4: Left: Original Paths of a dozen players. Right: Resulting path after aggregation

comes from Latin origin and means cluster. It is a determination of equivalent classes of data. So basically a generalization of data. The combined term trajectory aggregation refers to a single trajectory representing multiple comparable trajectories. The aggregation method used in this work is based on the approach by N. Andrienko and G. Andrienko [1] and consists of the following steps:

1. Extract characteristic points from the trajectories,
2. generate a grid,
3. group the extracted points with the help of the grid by spatial proximity,

Figure 4.5: Left: Yellow circles resemble the paths point characteristics. Right: Characteristic points of the graph are grouped by spatial proximity. Color resembles which points belong to the same group. The big circle represents the centroid of a group.

4. extract the centroids,
5. create a sequence of cell moves and cell visits,
6. visualize the sequence with a *Catmull Rom Spline*.

Step 1: Extract Point Characteristics

Identifying the most important points of a path to recreate it with only a limited amount of points is the first step. Two of the most important points are the start and endpoint, but how are the other found? They can be extracted by searching for points of significant turns and significant stops, as well as points that exceed a certain distance. That means the algorithm needs a minimum angle, a minimum stop time, and a max distance. The minimum angle defines if two adjacent points have a higher angle between them, then the second point is marked as a point of significant turn and will be extracted. The minimum stop time defines how long a player had to stand still to make that point important enough to be extracted. The max distance regulates how much difference in the distance there can be between the last extracted point and the next one. Otherwise in long straight segments, where no significant turn happens too few points would be extracted. There is a minimum distance ensuring that two points that are to close to each other are counted as one single point. Different values for those variables result in different point characteristic, so it is important to trial and error a bit until the result seems good enough. The Figure 4.5 shows how those extracted points look when visualized.

Step 2: Generate Grid

Like in Figure 4.6, the next step is to create a grid that encapsulates all characteristic points that were extracted in step one. The width used for the grid cells can be adjusted to the developer's needs. A smaller width leads to more grid cells, and that will lead to a more accurate aggregation. With the grid, each extracted point can be represented with the grid cell that encapsulates it, which will be a great help in the next steps.

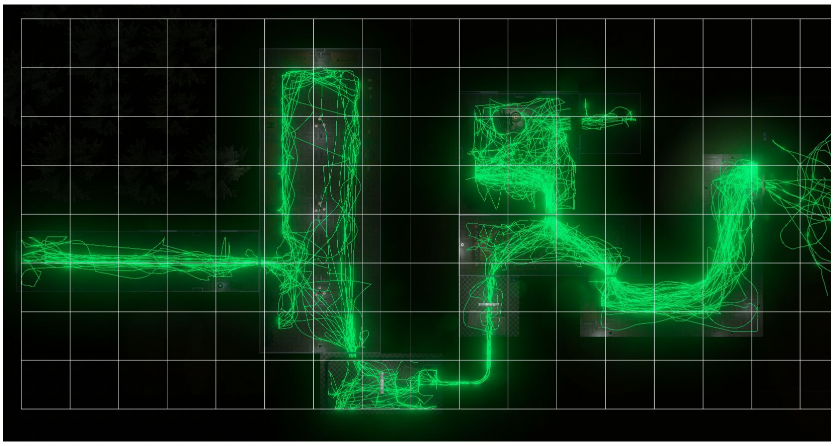


Figure 4.6: A grid is layered onto the trajectories. This helps certain algorithms be more efficient by having more spatial information (the cell id) about each point.

Step 3: Group Points

The goal is to create one group per grid cell. The groups can be generetaed by using a clustering algorithm and using the width of the grid cells as a maximum radius for a group. To find the right group for a point, the distance to each centroid in the grid cell and its neighbor grid cells needs to be calculated. The lowest distance to a centroid is the best group to put the point in. This means a point can be located inside a grid cell but belong to a group which's centroid is inside another grid cell. A centroid is a groups average of all points and is constantly updating when another point is added. The right side of Figure 4.6 shows the visualization of the groups.

Step 4: Extract Centroids

The centroid of a group is the average point of all the points in the group. When looking at 4.5 the big colored circles represent the centroids.

Step 5: Represent Trajectories as Sequence

As a result, the trajectories shall be represented as a sequence of cells $(c_1, c_2, \ldots c_n)$. They can also be represented by other sequences such as a sequence of moves and a sequence of visits. A move is a transition between cells, and a visit holds information on the start and end time of the player being inside a grid cell. A lot of additional data can be computed for each move and visit like the duration length and average speed of the player inside a grid cell.

Step 6: Visualize with Spline

The right side of Figure 4.6 shows the resulting visualization when using a spline algorithm on the cell sequence. For the representation of the sequence, this project uses a *Catmull-Rom-Spline*, as shown in Figure 4.7. One of the features of the *Catmull-Rom-*

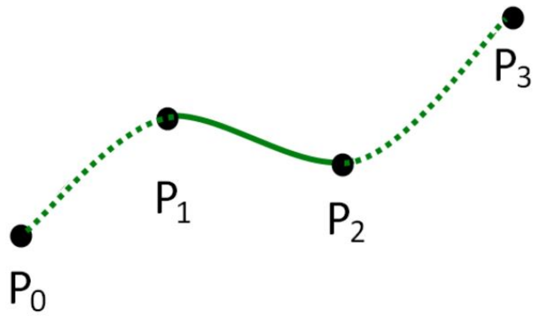


Figure 4.7: Image of a *Catmull-Rom-Spline* [34].

Spline is that the specified curve will pass through all of the control points. Two control points on each side are required to calculate a specific point.

4.5.2 Trajectory Tool

The window that controls the trajectory tool and providing an interface to the most important functions can be viewed in Figure 4.8. The windows offer buttons to load data of one or multiple players and render or hide them with a single click. By clicking on the other available options, additional information like point characteristics of the paths and the aggregated view can be shown. This tool was created using *Unity's* editor UI system. This system limits the aesthetic aspects of the tool but its enough for a prototype.

4.5.3 Eye-Tracking Shader

The Figure 4.9 of the eye-tracking data visualization, shows the outlines of tagged objects in different colors. This is the result of an outline shader. Basically, the shader works by rendering the object a second time but scaling the object a little bigger and presenting it in a given color. Then the standard rendering of the object will be rendered on top of that. The result is an outline with a specific width depending on the scale that was used and a particular color that can be specified in the material of the object.

The color of the outline will then be dynamically colored with a heatmap approach depending on how long players looked at an object. To do this, a gradient that goes from green to yellow to red is provided. The time the players looked at an object is interpolated on that gradient. The resulting color is then given to the object's material.

In Reality, two different materials are used for this effect. Depending on if the effect is used in editor mode or in real-time, the thickness of the outline needs to be adjusted. A first-person perspective camera in real-time mode and an isometric third-person camera in editor mode are rendering the objects quite a bit different. The outlines look a lot bigger in editor mode than they look in real-time, that is why a script automatically swapes out the width of the materials depending on the mode.

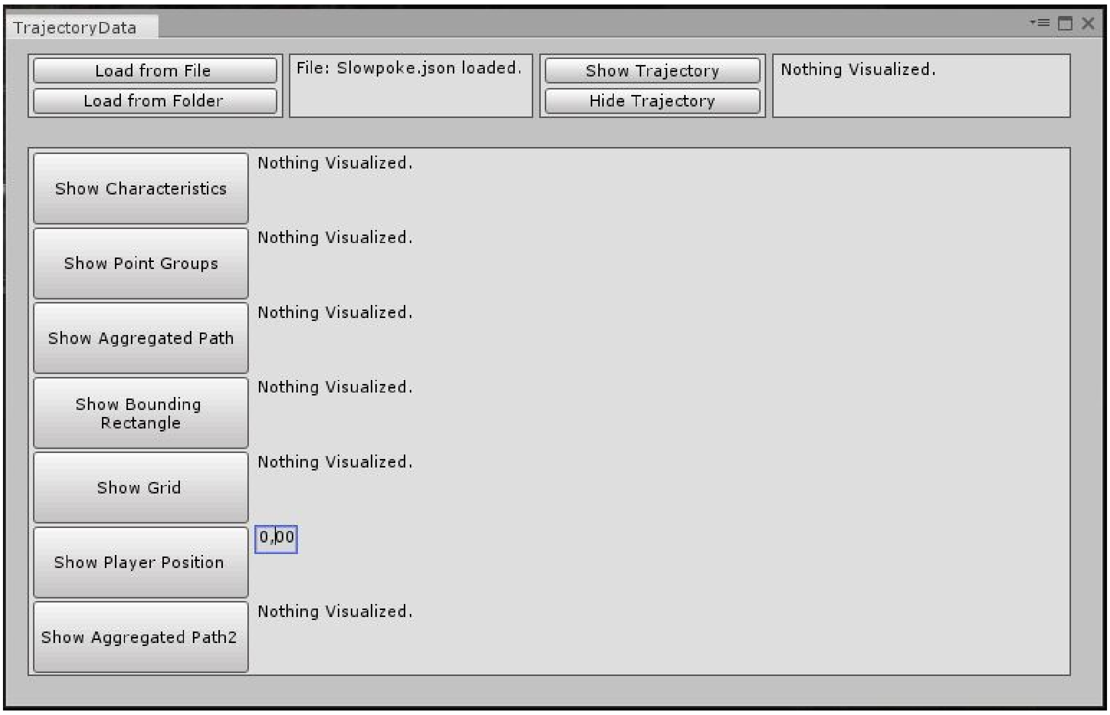


Figure 4.8: The Trajectory tool window created with *Unity's* editor UI provides access to many different views of the trajectory data.

Figure 4.9: The color of an objects outline resembles the time a player looked at the object. The color green means not looked at, while red means looked at a lot.

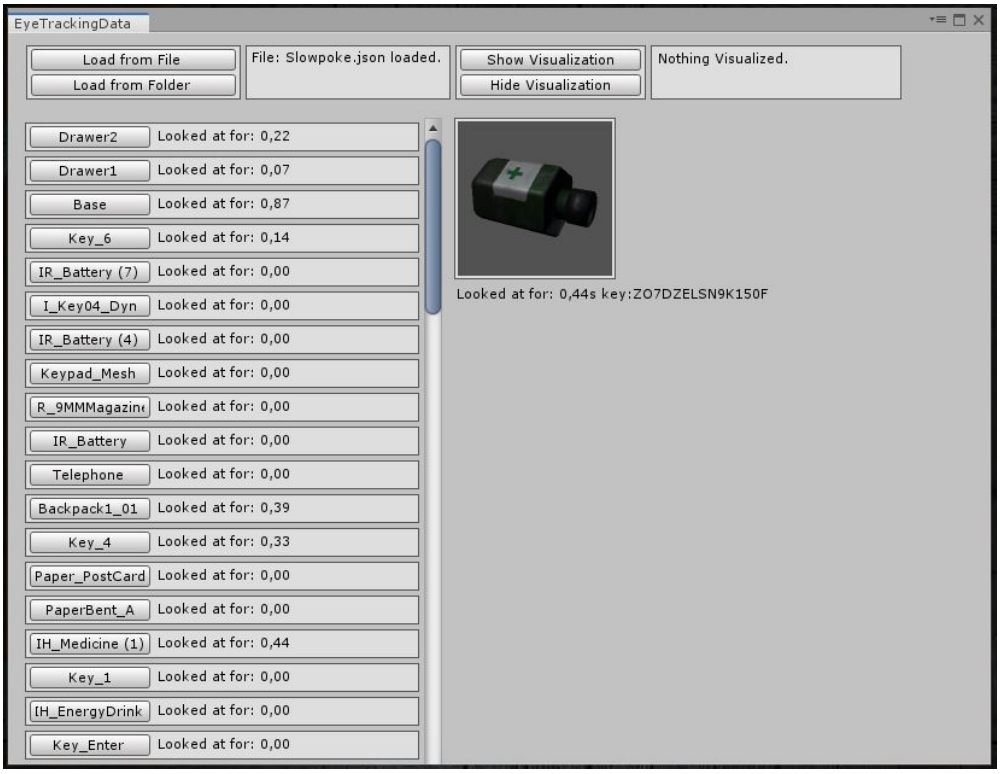


Figure 4.10: The eye-tracking window provides a list of all tagged game objects. By clicking on them the camera zooms in on them and more information on them is provided.

4.5.4 Eyetracking Tool

As already mentioned in the trajectory tool section, this tool is also created the same way with *Unity's* editor UI. The loading of the player data works the same way, but the specific eye-tracking data files have to be loaded instead. The tool also features a list of all game objects, that are being tagged as eye-tracking targets. By clicking on one of them in the list, the window will show more details on it and focuses the camera on the object. Figure 4.10 shows how the window looks.

4.5.5 Emotion Particle Clouds

As previously mentioned, the emotions are color-coded with Plutchik's emotion wheel. Additionally, every emotion also has a unique particle sprite to separate it even more from other emotions. The Figure 4.11 shows the most common emotions with their own particle system.

The particle systems itself are created with *Unity's* built-in particle engine called Shuriken. Particle size and spawn rate are controlled dynamically via a script and change depending on the emotional value. The trajectory and the emotion data of the player have to be loaded to spawn those particle systems at the right location. This ensures the tool has the needed data to know at which time which emotion occurred on which

Figure 4.11: Emotion particles from top left to bottom right: Anger(red), surprise(light blue), joy(yellow), sadness(dark blue), fear(green), engagement(orange), disgust(purple) and contempt.(lighter shade of purple)

area and can then spawn the corresponding particle system on said location. However since too many particle systems will be quite expensive on the rendering engine as well as produce a lot of overlapping, the emotions have to be filtered, and only the most prominent emotions should get a particle system representing them. For each emotion, the twelve most significant values from each player are extracted. This sounds a lot. However, most of those values happen nearly at the same time or at the same location. This leads to the next step. Another problem might occur if more than one emotion is prominent at the same location as another. In this case, the particle systems are rendered onto each other, and the result is a mess. To solve this problem, the algorithm is restricted to a maximum of three emotions per location. Of course, the three strongest emotions are picked in this case. Then the particles will only spawn at an angle from zero to 60 degrees in the first system, from 60 to 120 degrees in the second one and from 120 to 180 degrees in the third one. This is shown in action on Figure 4.12.

4.5.6 Emotion Tool

This tool works similar to the other tools but requires the emotion data and the movement data of one or many players to be loaded. Without the movement data, the particle systems can not be spawned at the right locations where the player was at. When selecting one or multiple particle systems, the window shows a list of them. By clicking on a particle in the list, it will move the view to the object and show more information on the data. The tool is pictured in Figure 4.13.

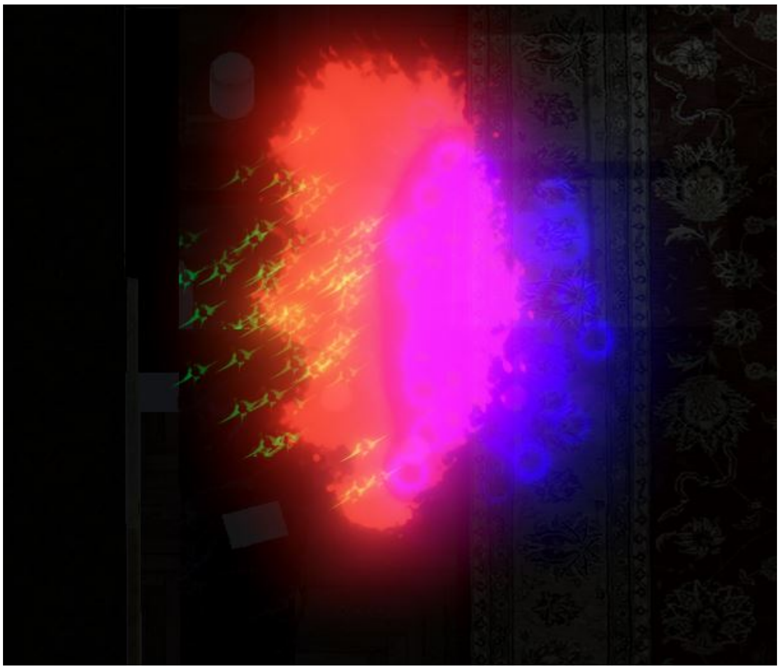


Figure 4.12: This particle system consists of the emotions fear anger and sadness. Each emotion spawns particles at their own angle in to limit overlapping

4.6 Issues

As in any project, several problems and difficulties showed up over time. Most of them could be solved or worked around, but they still consumed a lot of time and effort. The most time-consuming part was getting the trajectory aggregation to work. The literature and guide I followed were using the aggregation for 2D visualizations, and it was hard at times to convert those algorithms to 3D. For example, creating a Voronoi diagram with 3D data was something that just did not work and crashed *Unity* a lot of times. That is why I skipped that part and used a normal grid as a replacement for the Voronoi cells, which worked fine. Another technical difficulty was the shader that is used to visualize object outlines. The width and color of those outlines changes depending on the eye-tracking data. The problem is that when looking at the effect in editor mode, it looks a lot different than in realtime first-person view. The problem was solved by switching the shader depending on if the engine is running in play or edit mode. Another issue was the black box that is the Affectiva SDK which provides interface methods to get emotional response data, but provides no information on how those emotions are calculated. Most of the experts that evaluated this project asked about how the emotions are tracked exactly, and I could not give a detailed answer about how the SDK does it. By the way, finding and persuading the experts to help evaluate this game was not an easy and a very time-consuming task. Also finding enough playtesters was an issue, and how much are enough testers? In the end, a dozen of playtesters and five experts were found.

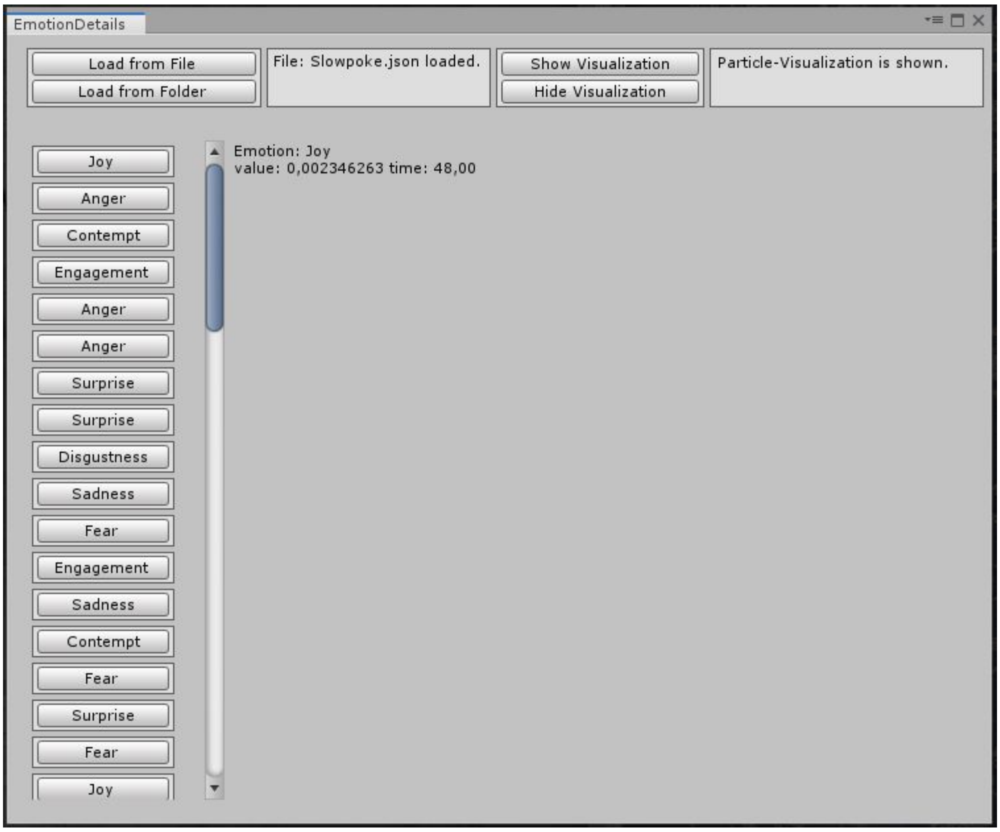


Figure 4.13: When selecting emotion particles in the scene view, A list of all selected emotions is shown on the editor window. The items in the list can be clicked to zoom in on them and show additional information.

4.7 Prototype

The resulting prototype or rather proof of concept and all of its features is talked about in the next sections.

One of the more significant properties of this tool is that anyone is able to view the visualizations in editor and in play mode. This means the view of the visualizations can be looked at in first-person while experiencing the game. This provides an even closer understanding of how one or many players reacted in different areas of the game.

4.7.1 Data Tracking

This part of the tool is separated to change or switch it in the feature. However, this was still a bigger part of this project, and I would like to dive into its features.

This tool allows one to track essential player data such as position and orientation of the player character at all times. The only requirement is to link the player character transform to the DataTracker script by dragging it onto the inspector of the script. The same goes for eye-tracking. The tool tracks if the player looked at specific tagged

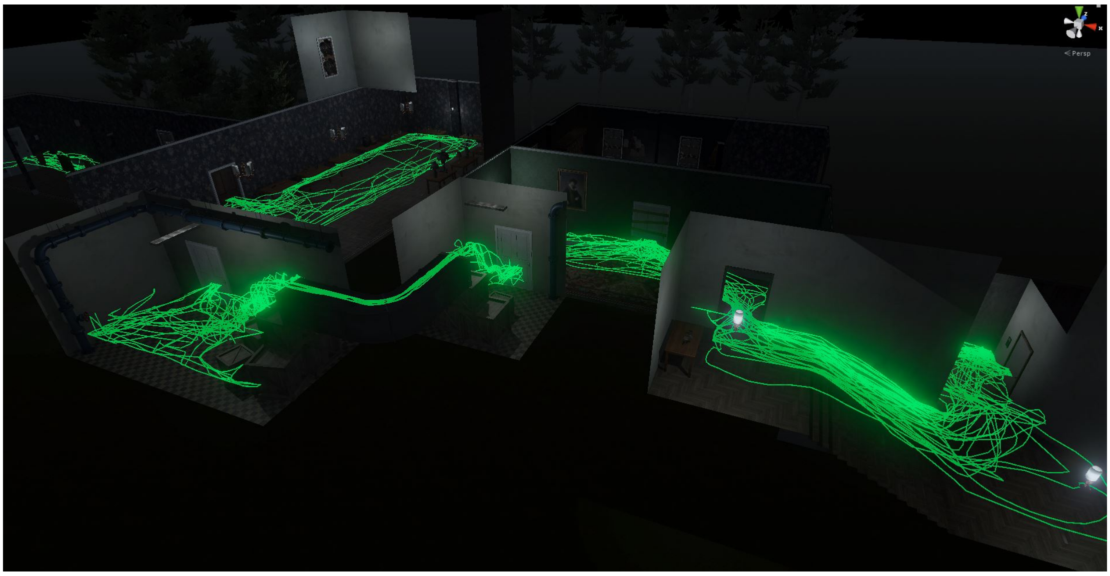


Figure 4.14: The trajectoy visualization can be looked at at any angle.

objects. The objects that are needed to be tracked can be just be tagged, and the only other requirement is that it or any of its children objects uses a collider component. Of course this tool currently only works if a compatible eye-tracker is connected to the computer. Any Tobii eye-tracker of the current generation as of the time of writing should work fine.

By connecting a webcam to the computer, the tool is able to track the emotions of a person who sits in front of the webcam. Make sure the person's face is clearly visible by the camera. When playing the game and the face is not recognized, a warning shows up on the screen telling the player that the face was lost. This warning vanishes as soon as the face is recognized again.

4.7.2 Trajectory Visualization

The trajectory visualization can show the movement path of one or many players rendered by green lines, as it can be seen in Figure 4.14. If the player wants to see the general game flow, he can aggregate the currently showed trajectories with just one click and look at the result. It is also able to show the characteristic points of the trajectories and group those points by spatial proximity. Those views, like in Figure 4.15 provide more insight into how the aggregated version of the visualization came to be.

Figure 4.16 shows how the visualization looks in realtime first person view.

4.7.3 Eye-Tracking Visualization

The eye-tracking visualization provides an interesting view of human psychology. Looking at Figure 4.17 for reference it allows us to see which objects in a game a looked at more often than others and which objects are not recognized at all. This way, it can

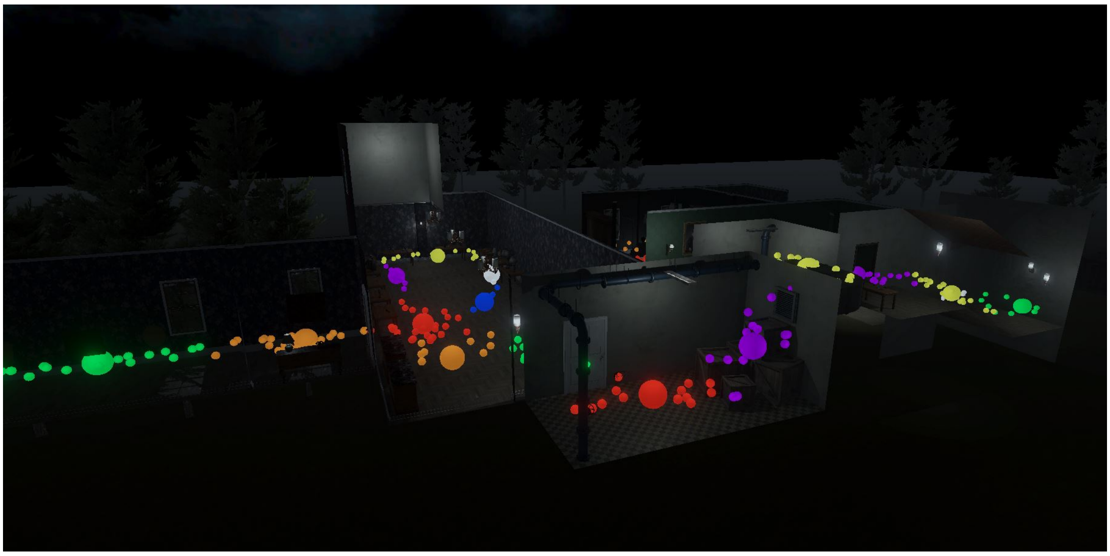


Figure 4.15: This Figure shows the point groups used for aggregation of Player1's data.

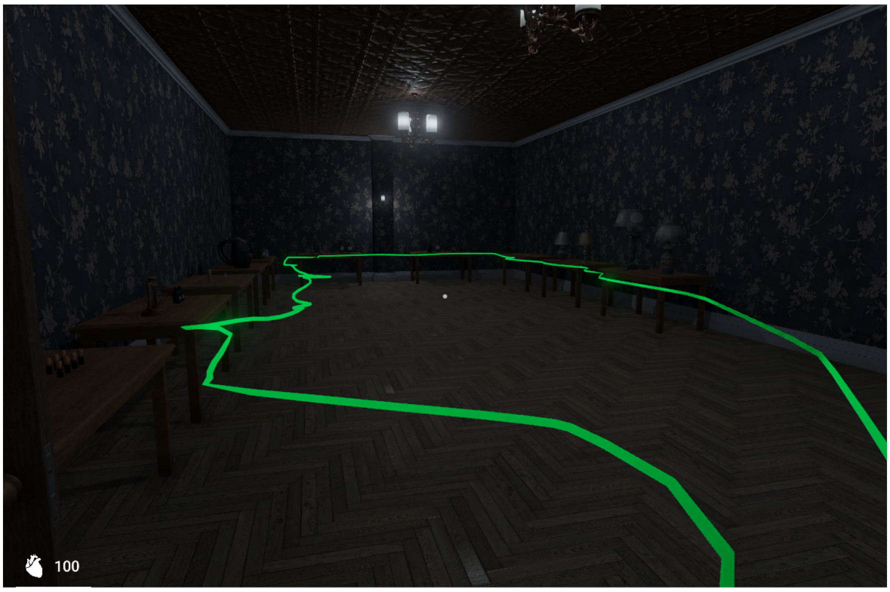


Figure 4.16: The trajectoy visualization from a first-person perspective.

also be tested to put the same object on different locations and compare how much it changes how players look at them. This visualization looks a bit different in play mode as it requires a different material, as seen in Figure 4.18. However, it works the same way, and since the player can look at them from up close, the outlines do not need to be as thick as in the editor view, to be recognized. The tool also provides a list with all tagged objects and when clicking on one of these shows even more information on

Figure 4.17: Average eye-tracking data of a dozen players.

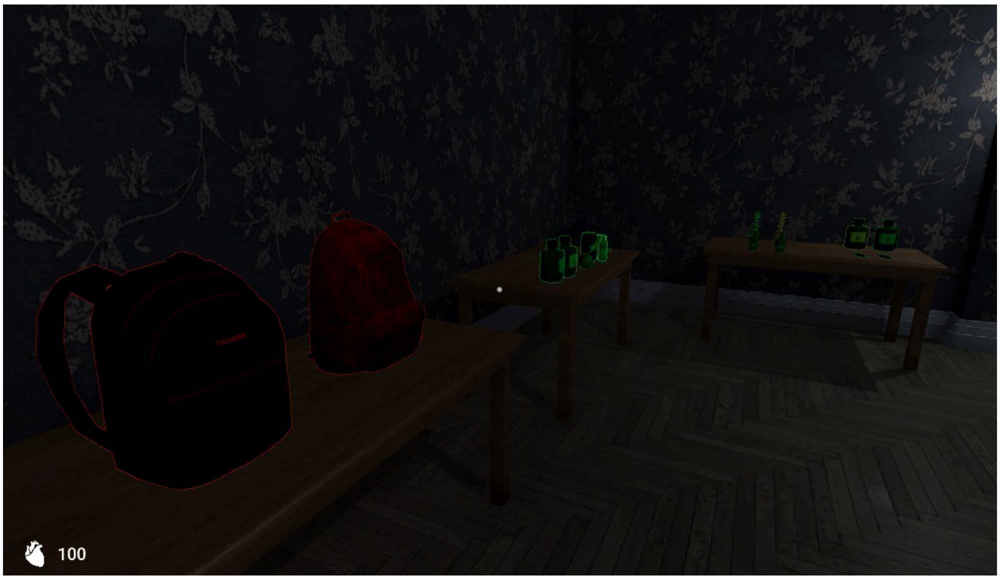


Figure 4.18: The same data as in Figure 4.17 but shown in first-person view. In comparison the outlines of the objects are much thinner.

them like how long the player looked at them exactly. It also automatically focuses the camera on the clicked object.

4.7.4 Emotion Graph Visualization

This feature is only a smaller, less significant part of the tool. It is nice to have but did not get as much effort put into it compared to the other visualizations. This graph visualization shows how emotion has changed throughout the playtime. The visualized emotion can be changed, but only one emotion can be looked at at a time. The visualization automatically adjusts the size and the value of the x- and y-axis to fit in the resulting line graph optimally. The intended use of this feature is for game developers to see which emotions resonate the strongest in players during their game and how the emotion fluctuates. To get an idea of how this visualization looks, see Figure 6.4.

4.7.5 Emotion Particle Visualization

The probably most unique visualization of this project is the emotion particle visualization. The goal of this visualization is to provide hot spots of emotions that can be seen at first glance to identify problem areas or areas of high reactions quickly. Please have a look at Figure 6.5 to get a better impression. This feature can also provide a more detailed view when selecting one or multiple of those particle systems. The detail view shows the exact values that the emotion recognition calculated and allows the user to focus the camera on the selected emotion particles. In the case of overlapping, just selecting the bunch and looking at the detail view can provide more insight. Needless to say, overlapping should be minimal at least when looking at a single players emotions because the emotions are first of all filtered and at most three emotion particle systems at a time can spawn in the same area. If that is the case and three different emotions occur at the same location, the particle systems adapt their spawn angle to not overlap with the others. For more detail see Figure 4.12.

4.7.6 Combination of Visualizations

A property of the prototype that allows each visualization to work on its own and combine those different visualizations at will. One can have the emotion particles of player1 active while looking at the average eye-tracking data of a dozen players, while also having the trajectory visualization for player1 and the aggregated trajectory of all players visualized. The combined visualizations can be seen in Figure 6.6.

Chapter 5

Collecting Playtesting Data

5.1 Environment

At the University of Applied Sciences Hagenberg on the 6th of June 2019 the first Gamers night event, where most of the playtesting sessions happened, took place. Visitors and exhibitors alike were able to play and test a variety of games, including the horror game for the thesis project. The players mostly played for themselves and were only being helped if they asked specific questions. The Environment consisted of two rooms with about twenty PCs each. On each of those computers, a game was ready to be played. On one of those PCs, the demo horror game this project created was playable.

5.2 Procedure

Any player could start the level by himself by pressing start in the main menu and play as long as he wanted or until finishing the first level. After that, the player was asked to do a short survey.

5.3 Participants

The participants were random voluntaries, but all of them had a gaming background. Some of them are even game developers themselves. Most of the participant were on average 24 years old with a standard deviation of 1.7.

5.4 Technical Setup

All of the hardware and software used during the playtests and how they were set up will be explained in this section.

5.4.1 Hardware

A mouse and headphones was provided for the participants. No external keyboard was necessary, since the test was run on a notebook. The playtest was performed on a Lenovo gaming notebook with the following specs:

- Intel Core i5-4200 CPU,
- 8 GB RAM,
- NVIDIA GeForce GTX 860M,
- 500 GB SSD,
- Windows 10 x64.

5.4.2 Data Tracking

The game tracks the movement of the player's controlled character in the background. Every 100 ms, the current position of the player is saved into a collection. After finishing or closing the game, the collecting is serialized in a JSON file. To track emotions with the Affectiva SDK [22], a camera is required to film the player's face during the playtesting session. The notebook's integrated camera was sufficient. The SDK can then detect facial gestures and grimaces. If the face is not recognized because the player is not visible by the camera or is too far away, a small info textbox shows up reminding the player to position his head in front of the camera. Tobii EyeX2 Controller [28] was set up and installed on the notebook. Using the included software, each participant was able to create a personal eye-tracking profile. By creating such a profile, the tracker gets customized to the position of the person's eyes. This process includes looking at points at different positions on the screen until they explode. This will usually take less than five minutes. The JSON file containing the tracked data is saved inside the streaming assets folder of the build. This folder contains subfolders called MovementData, EmotionData, and EyeTrackingData. Those folders store the corresponding JSON files. Each file also has a session id to identify which file corresponds to which play session.

5.5 Survey

The players were asked to do a short anonymous online survey after playing the game. The exact questions of the survey can be found in Appendix B.1. There were a dozen players who did the survey, and the most common opinions were that the code puzzle was a little too difficult, and it was their least favorite part of the game. Their favorite part of the game was the atmosphere. One out of those twelve participants was female. The average playtime was about ten minutes. One of the participants did not finish the level and played only through two-thirds of the game.

Chapter 6

Expert Evaluation

This work aims to prepare and visualize subjects emotional responses, as well as pupil size and position, at a certain time in a gaming experience. This amounts to more accessible data for game designers and researchers. They can use this to improve their game or application. In this chapter, the methods and participants of this study are introduced. The Discussion of the evaluation and the results will happen in the end.

6.1 Research Question

In this Chapter, the previously stated research question will be answered. As explained in the previous Chapter, a few different types of visualization methods have been developed in the thesis project. One visualization is for the trajectory data, which shows information about how the players traveled through the level. Another visualization focuses on eye-tracking data and can be used to see which objects in the game have been looked at. Two emotion visualizations were developed. One as a simple graph showing the flow of each emotion during the whole game. The other one visualized with particle clouds in the level itself. Those particle clouds change color shape and spawn rate based on which emotion was present and how strong the emotion was. By evaluating each of them, it is possible to conclude whether or not those methods are usable and efficient for analyzing player data. And also if they are helping developers to enhance the design decision-making process.

6.2 Methods

The following methods were used for this evaluation:

- *Personal Data Form:* First of all, the participants filled out a form about their personal data and game development experience. This form can be looked at in Appendix B on page 64.
- *Short Introduction:* The participants were able to play the game one time for themselves to get an experience to what time of game they will be analyzing. After that, I gave them a short introduction to the project and how to use the tool inside the *Unity Editor*.

- *Task Survey:* After that, the main part of the evaluation was completing given tasks and writing feedback on each individual task on the survey. This helps to analyze specific use cases on the tool by having different opinions on each task. The Tasks can be found in Appendix B on page 69.
- *Questionaire Metrics:* Afterwards, each participant rated the visualizations based on specific metrics. The questionnaire can be looked at in Appendix B on page 70.
- *Questionaire Feedback:* At last they filled out a feedback questionnaire that can also be found in Appendix B on page 72.

6.3 Approach

The participants could decide if they want to be recorded with a dictaphone for the whole session. The other option for them would be to write their feedback on each individual task down on the sheets. The recorded sessions have been transcribed later. After all experts finished their evaluation all of the transcribed information has been worked through and the number of opinions and statements that occurred more often has been noted.

6.4 Participants

Following participants took part in the evaluation: Expert1's job description is called "Unity developer" and he has five years of game development experience at the age of 28. His specialty is 2D platformer games, action games, and hybrid board games. For testing, he mainly uses playtests and game data analysis. He comes with little experience in analyzing play data; however, mentions that in his opinion analyzing player data is important (four points out of five).

Expert2 is employed as a "Unity developer" at the age of 32 with a whole seven years of game development experience. He mostly worked on free-to-play multiplayer online battle arena games, farming games, and hybrid board games. In terms of analyzing player or game data, he has used proprietary tracking and playtests before. He mentions that player analyzing is very important; however, this part is usually done by his colleges.

Expert3 works as a game artist and educator and has been working in the game branch for ten years already. Her experience in the game industry range from mobile and serious games to augmented and virtual reality games. She also has a lot of experience with *Unity*. In the past, she used *Google Analytics*, *Unity Analytics*, as well as custom tools and third-party plugins for analyzing games. She states that analyzing player data is very important for user experience and optimization as well as to find out if business and customer goals have been reached.

Packing four years of game development experience is Expert4 at the age of 31. His current job description is gameplay programmer. He worked on a Diablo-Esque mobile game, a mobile tactics game an on a three vs. three sports brawler game. For testing, he uses custom events logged to a database. In his opinion analyzing player, data is pretty much the most reliable feedback to get. Hearing players opinions is important but limited.

Expert5 is currently the CEO of a small startup, expertising in customized digital media systems. In his 41 years of live experience, he did not have a long history with game development except playing them himself. However, he is an expert in the field of virtual and augmented reality experiences. With over eight years of experience of working with mixed reality technologies, including two years teaching mixed reality at the University of Applied Sciences in Hagenberg, he has worked on several different mixed reality projects. The most notable one is *Dominator Experience*, where the user was able to experience the whole out and inside of a *Dominator* yacht in virtual reality. His preferred method of analyzing and testing users and applications is observation.

6.5 Procedures

This Section describes everything from getting in contact with the participants to giving them an introduction to the project and the demo game, letting them test and play around with the tool and discussing the results with them.

6.5.1 Contact

To start, a lot of game companies in Vienna and Upper Austria have been sent emails, requesting their assistance with the evaluation of this project. The thesis supervisors provided most of the contact data, and some contacts have been searched and found on the internet. In the initial mail, the project was explained shortly, and their help in testing and evaluating the project was requested. More information on the project has been given to them on-demand until some of them approved and offered me their help. Soon after that, the first appointments were made. Those evaluations happened in a room at the University of Applied Arts in Vienna provided by one of my supervisors, and sometimes directly at the corresponding companies game studio.

6.5.2 Introduction

After the initial meeting and greeting the participants which I will refer to as experts from now on, were given a short introduction to what the project is about and was offered to play the demo game for ten minutes. This was found necessary because analyzing data of a game that they know nothing about is not helpful, and this way, they experienced the game level for themselves. Before continuing, they signed a consent form and a personal data sheet which can be seen in Appendix B on page 64. The next step was to give them a short introduction on how to navigate the tool in the *Unity Engine*. Since all experts were already familiar with the *Unity Engine* itself, this part was understood by all of them very fast.

6.5.3 Technical Setup

The tool was running on a notebook with *Unity* installed and the demo project running. A demo build of the game was also installed and worked without problems. Aside from the usual input peripheries like a mouse and the built-in keyboard, nothing else was necessary. Figure 6.1 shows Expert5 working with the tool.

Figure 6.1: In this picture Expert5 is currently using the project working out a given task.

Figure 6.2: Original trajectory of Player1 in an orthographic topdown view.

6.5.4 Tasks

The main part of the evaluation was about the experts trying to do all of the tasks on the task sheets (Appendix B on page 69). They also evaluate how helpful the tool was for achieving each specific task. They were able to rate each task by giving one to five points, one meaning not helpful and five meaning very helpful. This whole process was mostly done without outside help unless they asked for it. Task1 is about exploring the capabilities of the trajectory visualizations of the tool. This task asked the experts to find areas where players struggled and what they think could have caused that. Figure 6.2 shows the trajectory visualization that is intended to be used for this task.

Figure 6.3: This Figure shows Player1's eyetracking-data in from an orthographic top-down view. Red outlined objects indicate which objects have been looked at the most.

Task2 focuses on interpreting the eye-tracking visualizations. The participants were asked to find the objects players looked at the most and least and why that was the case. See Figure 6.3 for a preview of the used visualization for this specific task.

In Task3, the experts were expected to analyze the emotion data based on a graph visualization and on Task4 they again tried to analyze the emotions of the player but based on particle visualizations. That task asked the experts to find locations where players were surprised or frustrated and why they think the players felt that way. The graph visualization for Task3 can be looked at in Figure 6.4 and the visualization for Task4 is shown in Figure 6.5.

In Task5, the experts had to use a combination of all visualizations to finish the task. The purpose is to test how well the visualizations go together. Take a look at the combined visualization in Figure 6.6.

6.5.5 Metrics

The participants have evaluated the following metrics by giving one to five points. A single point means poor resemblance of the metric and five points stand for excellent. Those metrics are taken from [20] where they have been introduced and reduced to six in [18]. In Figure 6.16, the average ratings of the experts for each of the following six metrics can be seen:

- *Clarity:* Is the displayed data clearly interpretable or ambiguous?
- *Readability:* Are the visual elements easily legible and distinguishable?
- *Informativeness:* Does it provide interesting or new information?
- *Aesthetics* Is it visually appealing?
- *Accurateness:* Is the displayed data accurate enough?
- *Usefulness:* For which tasks is it useful?

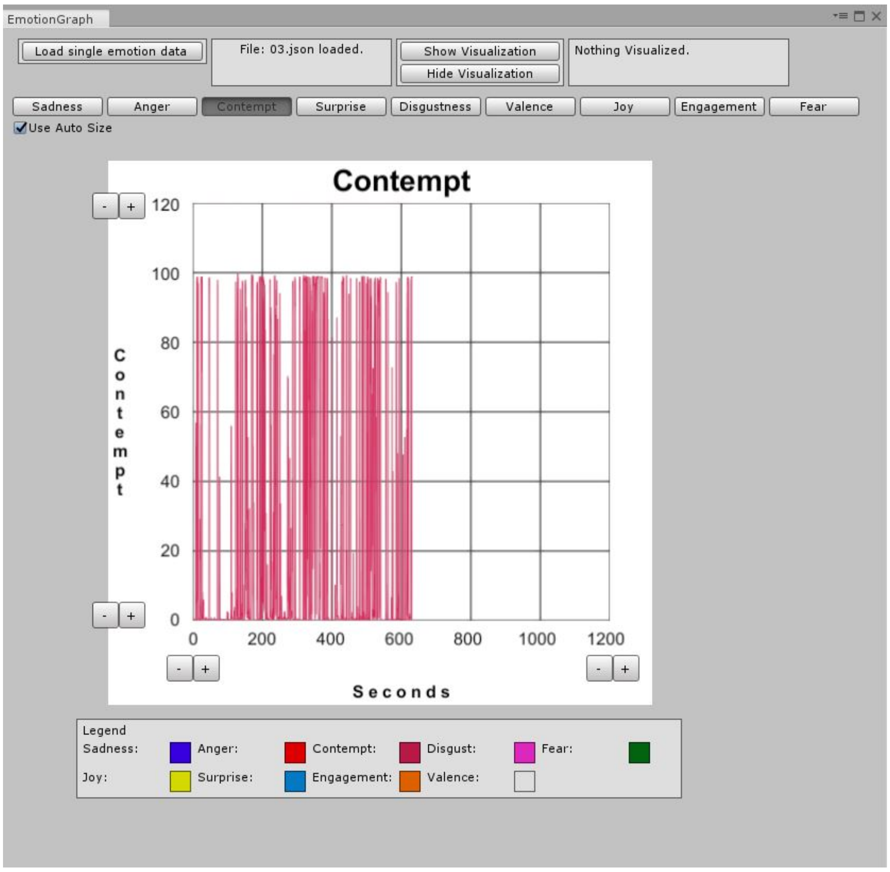


Figure 6.4: In this graph Player3's contempt emotion, that was present from start to finish of the level, is shown.

6.5.6 Feedback

The participants filled out two pages that were focused on their feedback on the tool and how to improve it. The specific questions they have been asked can be looked at in Appendix B on page 72.

Trajectory Visualization

In general, the most liked things about the whole tool was the trajectory visualization. Being able to view the original path data for any player and also aggregate it to see the general game flow for multiple players provides a lot of important information.

Eyetracking Visualization

This visualization was liked because it makes it easy to identify objects that were looked at a lot on first glance. However, Expert5's opinion on this matter is that objects that were looked at the least and objects that were not looked at at all are not easily distinguishable.

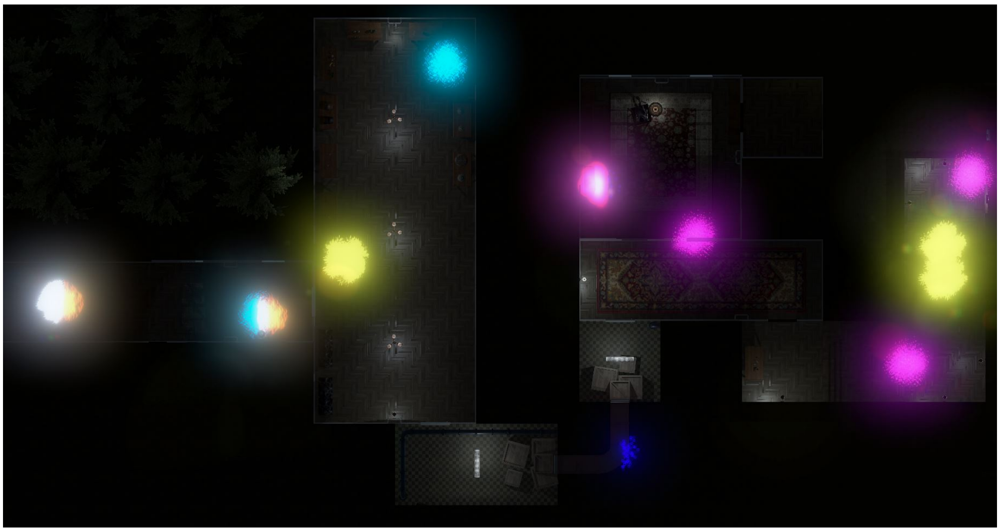


Figure 6.5: This topdown view of Player1's emotion visualization is what was used by the experts during Task4.

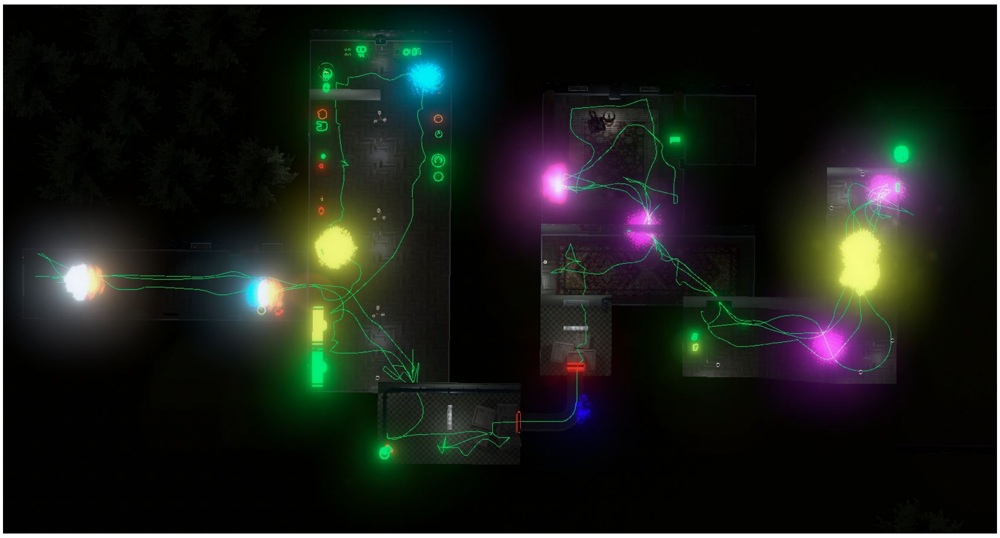


Figure 6.6: Combined visualization of Player1's trajectory, eyetracking and emotion data.

Emotion Visualization

The idea of seeing emotions as particle clouds on the game level did also impress the participants, and they thought it looks nice and points out right away the most important areas in the level. Emotions tailored toward game design was another mention by Expert4. Regarding this filtering, the visualizations for specific emotions like fear in a horror game would be helpful.

General Feedback

The integration into the *Unity Engine* was also a highlight for all of the participants because they could combine development and analyzation and have no need for working with separate tools.

In terms of how to improve the visualizations, most experts (E1, E2, E4, E5) mentioned that it needs to be a bit clearer and the time flow needs to be integrated better. They want to see what happened at a specific time, for example, by scrubbing a time bar. Also, Expert2 and Expert4 mentioned the UI inspector part needs to be responsive to different display sizes and better spacing in general for most of the UI.

Demo Game

Improvement on the game level itself was also a question on the sheet. Here most of the experts agreed that the note with the code should be able to be picked up and looked at in the inventory. Also, the door interaction needs to be consistent and some boxes which the player had to jump over need size adaption.

Would they use it?

When asked if they would use an improved an polished version of the same tool most of the experts (E3, E4, E5) said yes. They said it is beneficial to use it with a smaller amount of playtesters to recognized game design problems very fast. Expert5 stated the following:

> "Yes, I would use it to try to comprehend the experience for different types of players."

The ones (E1, E2) who did not mention that is is not suited for the projects they are doing, which currently is hybrid board games. However, they would use it if they were to make a first or third person game.

6.6 Results

In this section, research questions are formed and then answered by having the opinions of the experts in mind. Specific opinions of them will also be stated to reinforce those arguments. Also the result of the task and metrics survey can be seen in Figures 6.15 and 6.16.

6.6.1 Trajectory Visualization

To evaluate the results of this part of the visualization, the following research question was tried to be answered.

How helpful was the trajectory visualization to find areas where players struggled? Also, what kind of problems can be identified when looking at multiple or only a single players data? How can those problems be fixed?

Figure 6.7: This shows Player1's trajectory data. When looking at it in detail it can be seen that he went back and forth in front of the door for a few times, indicating that he was having trouble opening it.

Task1b: Player Movement – General Problems

Task1b was about finding areas where player struggled with the help of the trajectory visualization. The experts (E1, E2, E3, E4, E5) did find backtracking problems in the level. Figure 6.9 shows how the backtracking is visualized. To fix the mentioned problems the most mentioned solutions by Expert1, Expert3 and Expert4 were able to pick the code sheet up, and the player can view the code in his inventory, rather than having to backtrack to the room with the sheet.

> "The tool makes it easy to identify where a lot of backtracking took place."

. . . is what Expert2 states, after looking at the aggregated trajectory data of many players. In the ratings of the metrics, this is resembled by a five out of five rating from three experts (E1, E2, E4) and a four out of five from the other experts.

Task1c: Player Movement – Single Player's Problems

Task1c required the experts to look for problems that only show up when looking at a single player's data. Two experts (E3, E5) found signs of a player having trouble opening a door. Figure 6.7 shows how the door problem was detected. Their opinion on how to solve this was changing the interaction method that is used on the door. It should be the same as the other doors to make them consistent throughout all doors of the game. Another problem was detected by Expert1 who found traces of a player having trouble to jump onto the boxes. He identified this error by seeing lots of chaotic trajectories in front of the boxes. Those trajectories looked like the player was repeatedly jumping on the spot as seen in Figure 6.8. He proposed to solve this problem by adjusting the

Figure 6.8: Player10 probably had a hard time jumping onto the boxes. The trajectory data in this visualization indicates several tries to jump onto the boxes.

height of the boxes. This quote of Expert1 describes quite well what most of the experts thought about this feature in general:

> "Finding rough errors is very easy when looking at a single players trajectory while looking at the aggregated trajectories of many players makes it easy to get an idea of the general game flow."

Expert5 mentioned when looking only at a single players trajectory, it is easy to see if the player is a first time or revisiting player, as well as if he is a gamer in general or not.

6.6.2 Eye-Tracking Visualization

In the same way as before, evaluation of the eye-tracking visualization is done by answering the following question.

How well are you able to identify the objects players looked at the most or the least? How are these objects recognizable and what could be the reason behind players looking or not looking at specific objects?

Task2b: Eye Movement – Most looked at Objects

Finding the most looked at objects using the eye-tracking data visualization was the goal of this task. Some experts (E1, E2) used the list in the inspector to look for the items with the highest "Looked at" value, others (E3, E4, E5) immediately recognized the red outlined items in the level as the items that were looked at the most. The most looked items were required to continue the level or had to be looked at to pass through.

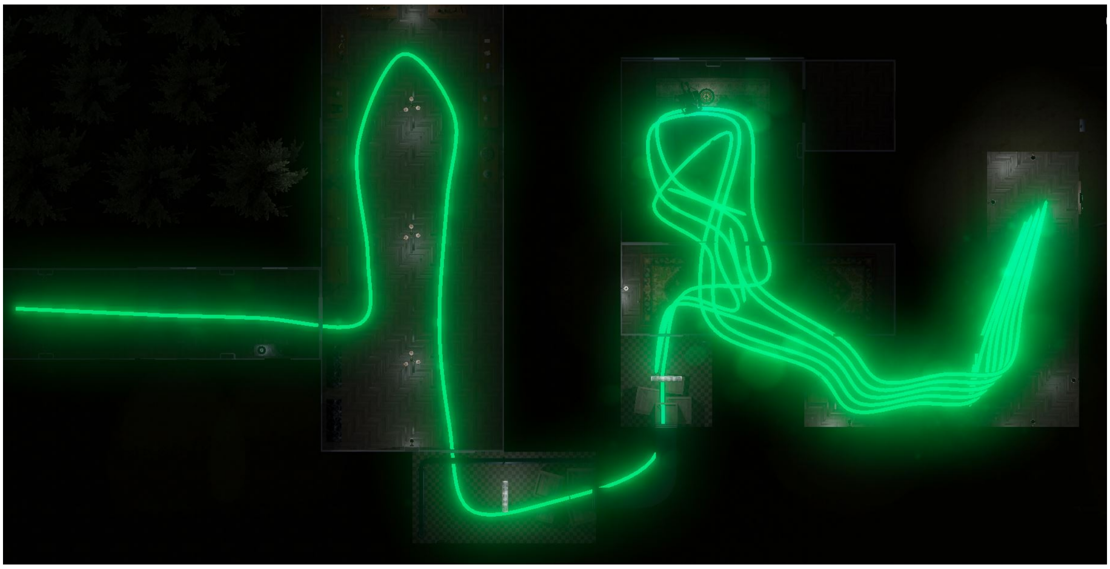


Figure 6.9: This aggregated path shows on first sight, where backtracking took place. The small offset on the green paths is there to not make the paths overlap. On Average players went through the hallway between the code input terminal and the room with the code sheet fives times.

In the second room of the game where a lot of items are lying around on tables, the experts could see that some objects were more often looked at than others. Why that is the case can sometimes be explained and other times just guessed. In general bigger items such as backpacks were looked at more often than small things like canned fish. Expert5 believes that both backpacks were looked at a lot because players wanted to choose one or the other and therefore, were comparing them before picking them up.

Task2c: : Eye Movement – Least looked at Objects

The goal here was to find the least looked at objects. Most of the above also applies to this task. Instead of red outlines, the green outlines show which items are not being looked at or only looked at for a short time. Almost all the experts except Expert5 gave this task the same rating because it is so similar. The reason Expert5 rated this task only a four out of five and therefore one point less then he rated the task for finding the most looked at values, is because he could not see the difference between an object that was not looked at at all and an object that was looked at for a short time. He suggested making the non looked at objects an entirely different color. Another interesting theory that was mentioned by Expert4 is:

> “Some items seem to be so accepted as important (like a gun and ammo), that the players pick them up immediately, not looking at them for long.”

However, the opinions united in the fact that this visualization is a great help to ensure important objects are getting recognized. This way if an object gets not enough attention, the solution would be to place it somewhere more visible.

6.6.3 Emotion Graph Visualization

The following question will get answered by the results the experts found when working on this task.

How and how well can the emotion of a player, that was present the most during a certain part of the game, be identified? Also is it possible to find out why the emotion was present?

Task3b: Most present Emotion

In this task, the experts had to use the graph visualization to find the most present emotion of a specific player. Reading the data on the graph visualizations proved not difficult for the experts, and everyone came to the same conclusion about what is the most common emotion. The graph shows how strong each emotion is at a time. The fluctuation of the emotion is also straightforward to recognize with the help of the graph. Most player data had contempt show up as their most present emotion. Some opinions (E1, E2, E3) about this stated that this was most likely due to the player being uncomfortable, which might be the case since it is a horror game. However, the experts had no insight into how the emotions are getting measured. They only know it had to do with facial expressions that were tracked by the webcam. When looking at a single player's data, Expert4 mentioned the following:

> "I'd say Contempt and Engagement are the most prominent. Without knowing how contempt is measured, it is hard to speculate about why it is so prevalent."

Expert5's opinion, however, states that the high contempt and engagement value derives from the players being highly concentrated. In a game genre like the demo game with a lot of suspense and having to search for items and clues, a high concentration is required. Expert4 also mentioned that the emotions used in this visualization could be more tailored towards game design. Emotions like contempt do not give as much relevant information for a game designer as for example, frustration. Expert5 complained about not being able to see all emotions at once on the line graph. This would make a good overview and save some time because he would not have to look at every emotions graph to find the most dominant one. This was the reason he was rating this visualization only a three out of five.

6.6.4 Emotion Particle Visualization

In Task4, the experts played around with the emotion particle visualization, and the following question will be answered using their results and feedback.

Is it possible to find locations in the game, where one or many players felt a specific emotion? How can those spots be recognized and are you able to deduce why those emotions were there?

The general impression of all the experts was that it was easy to find certain emotions based on the color-coded particle systems that stood out thanks to different shiny particles.

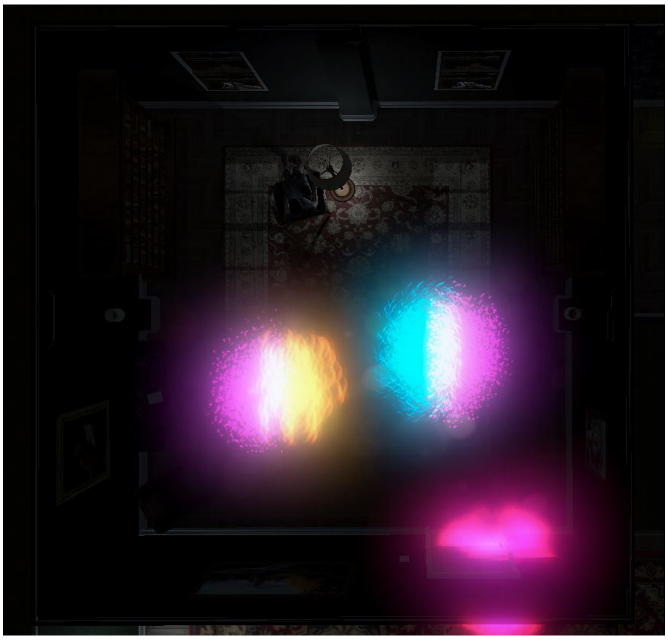


Figure 6.10: A location where Player3 felt surprised is at the spot where the demo games' only jump scare happened. The blue particle color indicates a surprise emotion.

Task4b: Emotions – Surprise Locations

Task4b required to look for areas where Player3 was surprised. All experts were able to find the two locations where Player3 was surprised. The Figures 6.10 and 6.11 show these locations and how the visualization looks. The blue sparkle particle system represents the surprise emotion. The experts knew which color to look for thanks to a color-code legend. The legend was visible on the editor window of the tool. The two locations are the area with the jumpscare and in front of the code input terminal. Surprise on the jumpscare spot is a no brainer. The general answer on why the player was surprised on the terminal was that he was surprised he entered the wrong code. He could also have been surprised after opening the door and seeing the zombie charging at him. The very high ratings for this task also reflect that the experts did not have problems in locating those areas.

Task4c: Emotions – Frustration Areas

Find locations, where Player4 was frustrated, was the goal of Task4c. The experts mostly looked for particles that resembled negative emotions, especially anger and sadness. They identified the spot right in front of the code sheet as the one with the most frustration for Player4. The Figure 6.12 shows the main spot of frustration for Player4. It shows that finding points of frustration was a bit harder than looking for specific emotions like surprise. The experts were unsure about which emotions exactly indicate frustration. This shows in the slightly lower rating of this task compared to the similar previous task.

Figure 6.11: Another location where Player3 was surprised is in front of the code input terminal.

Figure 6.12: Details on the emotion particle cloud of Player4 can be seen in the editor window when selecting the particles. Due to the combination of anger, fear and sadness this spot shows a possible point of frustration.

Task4d: Emotions – Similar Reactions

In this task, the experts had to check if other players felt similar emotions as Player3 and Player4 on those locations. Expert4 states:

> "Emotions seem to be somewhat similar. There seems to be more of a trend, which emotions correlate to which locations."

Mostly the same emotions showed up on specific locations, but on other areas players had completely different emotions. A lot of the evaluators (E1, E3, E4, E5) recognized the same emotions on the same players on locations, where for example a jump scare happened, or a problematic riddle had to be solved. The prominent emotion for the jumpscare was surprise. Many players had angry and sad emotions in front of the code-sheet and the code input terminal. To deduce why those emotions where there, is not hard. People tend to be surprised on jump scares and also get angry or sad if they cannot solve a riddle or enter the wrong code at the code input terminal.

However, on different locations like in the corridor, players had utterly different emotions or did not show any emotion at all. Some of these could possibly be due to error or external influences, like somebody entering the room and talking to the participant. This is why this data cannot be described as entirely reliable, especially if an emotion only shows on one specific player at a given location.

6.6.5 Combined Visualization

This section tries to answer the following research question:

When looking at a combination of all previously mentioned visualizations, does the shown information contradict or complement each other? Or do they show different independent parts of information? How clear and readable are the combined visualizations?

Task5: Visualization Combination

By using all of the visualizations at once in Task5, the experts tried to find out if the different displayed data can be useful in combination or not. The experts were unified in the opinion that these visualizations mostly show different kinds of data that can only be correlated so much. Because of this, the combination of those visualizations is even more important, to get a better understanding of a player's experience. Only looking at one of those visualizations might lead the analytics to believe something that might be the opposite. Expert5 mentioned that the combined display solidifies certain assumptions and interpretations that were made when looking only at one visualization. This quote of Expert3 describes this case very well:

> "The combination was very helpful, because for example when looking only at the trajectory data on a player, who was backtracking and running in circles a lot, one would assume he had problems progressing in the game. However, the emotion data showed that he was actually having a good time and was not feeling frustrated at all."

In all its usefulness, most experts (E1, E2, E4, E5) felt overwhelmed by looking at all visualizations at once, and smaller details are no longer visible. They agreed that the

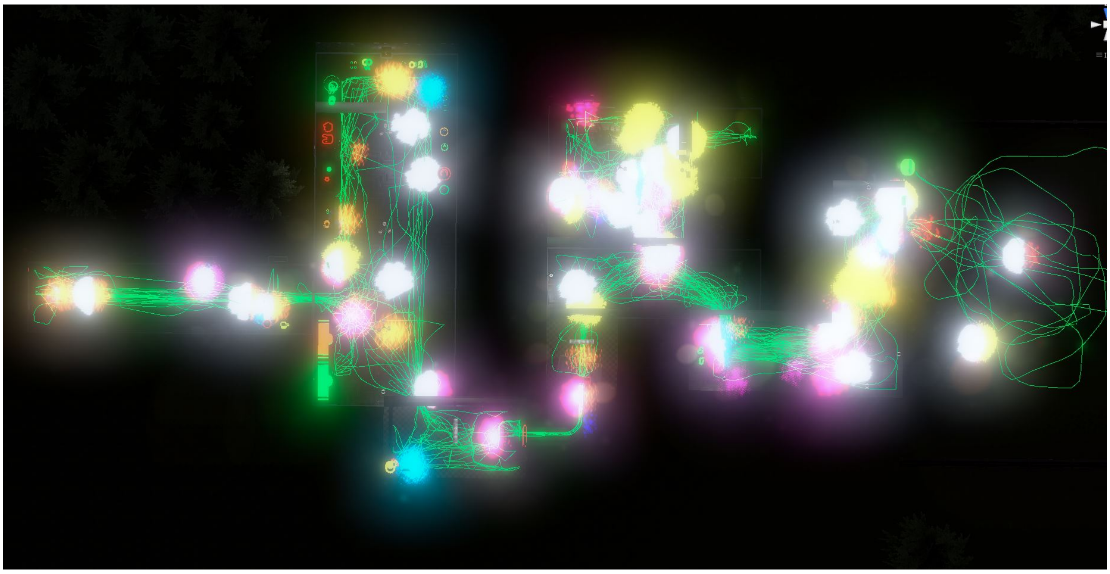


Figure 6.13: Combined visualization for all players. Without filtering the data there is a lot of overlapping involved.

combination was not very clear and needs a lot of optimization. They suggested features like filtering for specific emotions or a time-bar like on a video player to scrub around and only see data that took place at that specific time. Figure 6.13 shows how looking at all data visualization for all players at once can be overwhelming. Even though some of them were overwhelmed the rating of this visualization was still pretty high. This was probably because they were still able to complete the task without problems.

Task Ratings

The Figure 6.14 shows the rating of the participants on each task.

Metrics Rating

The individual ratings of the metrics can be seen in Figure 6.15. The lowest score of a two out of five was rated by Expert4 on the metric readability. He stated that this is because the exact emotion data is hard to read and understand. He would like to know how those emotions were calculated exactly, which is currently more of a black box. Other low scores like a bunch of three out of fives were rated on clarity (E3, E4, E5) and aesthetics (E1, E2, E4) can be related to the fact that the combined visualization had a lot of clutter and overlapping when looking at multiple players data. This view also had too much going on at the same time, ruining the aesthetic of the tool. The average ratings are shown in this Figure 6.16.

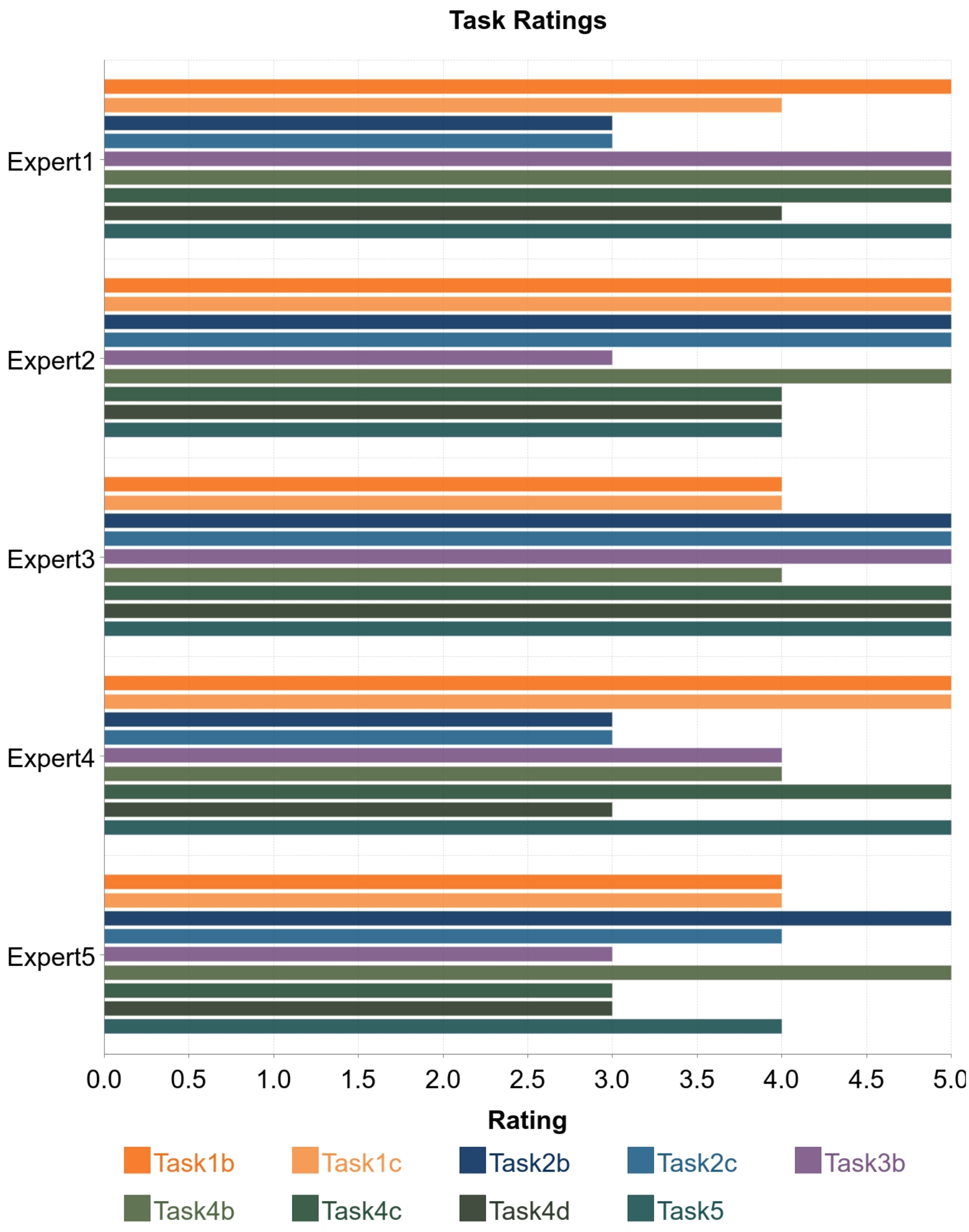


Figure 6.14: Rating of all experts based on specific tasks. Ratings from one (not helpful) to five (very helpful).

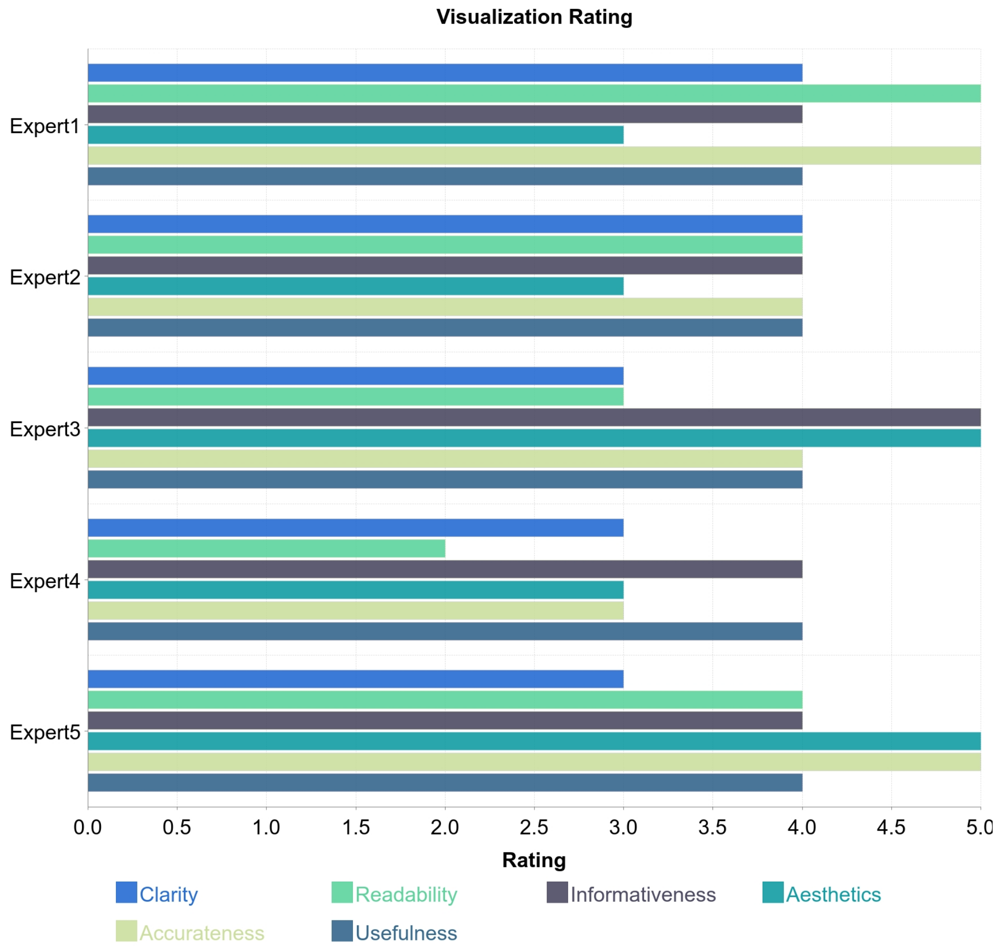


Figure 6.15: Rating of the visualizations by all experts based on specific metrics. Ratings from one (poor) to five (excellent).

6.7 Result Overview

In this section, the results are discussed by subject. When looking at the results of the expert evaluation and the survey the playtest participants did, it can be seen that the participants did not like the code puzzle and had problems with it. Since this is also what all experts found out, that can be seen as a success.

Features that worked out great:

- *Trajectory visualization:* The most liked feature of this tool provides a good overview of the game flow when using the aggregated visualization, as well as

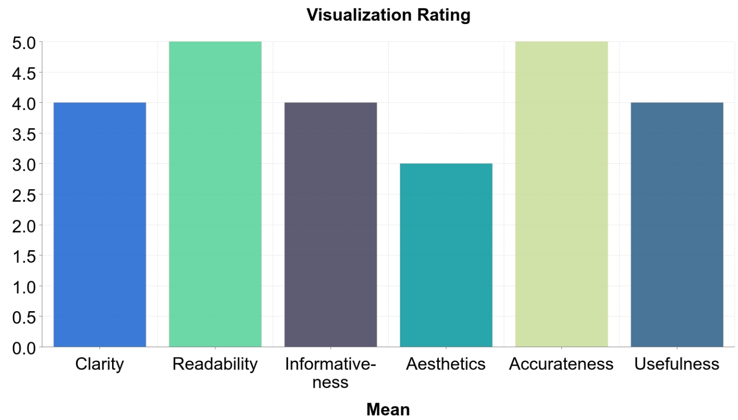


Figure 6.16: Average rating of the visualizations by all experts based on specific metrics. Ratings from one (poor) to five (excellent).

a detailed view of the original player trajectories.

- *Eye tracking visualization:* The heatmap approach on the outline color of the objects worked and was understood by most evaluators. This visualization provides an interesting view of human psychology by seeing which objects are looked at the most.
- *Emotion particle visualization:* This visualization shows hot spots in the game level on first glance and provides more detailed information when selecting a particular emotion. It has proven useful to find locations where players reacted with specific emotions efficiently. All Evaluators identified frustration or surprise areas by looking at the particles.
- *Emotions graph:* This feature gives a good overview of the emotion of each player. It should only provide an overview, and for detailed emotion data, the particle visualization can be used.
- *Similar emotions:*Specific hot spot areas show similar emotions on all players. This proves that it is possible to analyze a level based on player emotions since not every player reacts completely different. However, other less important game areas show completely different emotions on each player. This could be due to external influences, errors, or just each player reacting differently on minor things.

Some of the negative points that occurred during the evaluation were the following:

- *UI:* Every expert stated one or more times that to use this tool more effective and generally have a better experience using it, the UI needs to be improved. The Legend for the emotions color coding was difficult to read for most of them. The color that was closest to the emotion was not the actual color but the color of the next emotion.
- *Chosen emotions:* Expert2 mentioned that the chosen emotions are not tailored toward game design. He would like to have emotions like frustration and does not care about emotions like contempt, which he can't interpret very well.
- *Combined visualization:* The combined visualization had a lot going on at the

same time; that's why the experts had difficulty focusing on specific aspects of it. When looking at a single player's data, this was not a problem though.

- *Time visualization:* Experts had trouble knowing which emotion happened at what time since some emotions that happened in the same area were present during a different time in the game. Currently, when selecting an emotion, the time value shows up in the editor window, but that was not very helpful for them. They wished for a time scrubbing bar that shows or hides part of the visualizations depending on the time.
- *Emotion aggregation:* When looking at all players at once, each emotion is shown separately. Currently, there is no way of showing average emotions or another way of generalization. Expert2, Expert3, and Expert4 wished for an aggregated version of this.

6.8 Discussion

In this Section, the results are discussed by subject. Each of the experts found signs of backtracking using the trajectory tool and signs of frustration with the help of the particle visualization. This can be explained by looking at the results of the survey that the playtest participants did. Many of the participants did not like the code puzzle and had problems with it according to the anonymous survey. The survey states that ten out of twelve players found the code puzzle to be the most annoying part of the game. Since this is also what all experts found out, that can be seen as a success. The survey also states that their favorite moment of the game was the jump scare. This area has been found in Task4b by all experts when they were looking at locations, where Player3 was surprised. In Task4d, most of the experts (E1, E3, E4, E5) did find similar reactions on other players to this area. Being surprised does not correlate to the player enjoying that moment. However, it still undermines the fact that this could very well be the reason for their enjoyment. One player stated that he had trouble with the interaction controls. When looking at Player1's data, two experts found out that the player probably had trouble opening the second door in the game which had a different interaction method than the first door. The experts looked at the trajectory visualization, and it showed lines going back and forth in front of the door. Their interpretation was that the player was not able to fully open the door until trying for several times.

Chapter 7

Summary

The trajectory visualization shows the movement of players rendered by green lines. It also features an aggregated version to see the general game flow and reduces clustering and overlapping. Another feature is to show the characteristic points of the trajectories and group those points by spatial proximity. Like all other visualizations, it can be looked at any angle and in realtime while playing or in editor mode.

The eye-tracking visualization allows anyone to see which objects have been looked at for how long. This is because the outlines of objects have been colorized depending on how long they have been looked at. Green outlines symbolize that they have not been gazed at or only been looked at briefly while red ones resemble a high gaze dwell time. This feature provides a list with all of the important game objects. Those objects have been checked as to whether the player looked at them or not. Clicking on one of these shows even more information on them. It automatically focuses the camera on the clicked object.

The emotion graph is a simple graph visualization that provides insight on which emotion was present at which time during the game. By looking at all emotions after another the least and most present emotion can be found out.

The most unconventional part of this project is the emotion particle visualization. It provides hot spots of emotions that can be seen at first glance. This helps in quickly identifying problem areas, as seen in the results of the evaluation. This feature includes a more detailed view by selecting one or multiple particle systems. It will show the exact values of the emotion data and focus the camera on it. Overlapping is reduced by allowing at most three emotion particle systems to spawn at the same location. If that is the case, the particle systems adapt their spawn angle to not overlap with the others.

7.1 Playtest

The playtest session took place at the University of Applied Sciences Hagenberg. There a dozen players were able to experience the demo horror game and fill out an anonymous survey afterward. A webcam provided face tracking to record their emotional response values, and an eye-tracker recorded their eye movement. The game itself tracked the player movement. This data was later used for the expert evaluation.

7.2 Evaluation

Five experts have been involved with the evaluation of this thesis. Each one of them had the chance to experience the tool in a personal meeting. They started by playing through the demo game like the playtesters did, to get a feeling what kind of game they are analyzing. After that, they were given a short introduction to the workflow with this tool. The next step was for them to work out the tasks I have prepared for them. Each task was focused on a particular visualization of the tool, and the experts rated each task after finishing it. They rated the visualizations as a whole based on the following metrics: usability, informativeness, readability, usefulness, aesthetics, and clarity. Their last assignment was to fill out a feedback survey while talking about their opinions in paralell.

7.3 Results

Here is a short overview of the results of the evaluations:

- *Trajectory visualization:* This tool provides a good overview of the game flow when using the aggregated visualization, as well as a detailed view of the original player trajectories. The experts were able to spot backtracking spots and even spots where the players had trouble opening a door or jumping onto a box.
- *Eye-tracking visualization:* The experts understood the visualization. With its help, they were able to find the most and least looked at objects in the game just by looking at the outlines of the objects.
- *Emotion particle visualization:* It was easy to find spots where a player experienced a specific emotion like surprise. Emotions of players appeared to be similar in spots like jump scares and puzzles and different on locations like hallways were nothing extraordinary was happening.
- *Emotions graph:* This feature was was useful for finding out how much a particular emotion was present during the game. However not much else can be derived from the visualized data.
- *UI:* The experts had trouble understanding the UI. For example, the legend explaining which emotion corresponds to which color was understood wrong by them because the spacing of the label and the color was off-putting.
- *Combined visualization:* This was fine when looking at a single player's data, however looking at multiple players it was too overwhelming for the experts because of cluttering and overlapping.
- *Time visualization:* Experts had trouble knowing what happened at which time and wished for a time scrubbing feature.

Chapter 8

Conclusion

The research question that has been answered in this work is the following:

> Which visualization strategies can be employed for visualizing players' movement, eye-tracking and emotional response data in order to enhance the design decision-making process?

One major part of the evaluation was to analyze specific parts of the data related to specific tasks. Those tasks had to be completed by the experts with the help of the developed visualizations. For example Task4c was:

> Find locations where you think the player X was frustrated? How did you identify the locations, and what could be the reason for the frustration?

Since the tool contains various visualizations, those other questions purpose was to evaluate the visualizations individually. Those questions were mostly about finding areas where players struggled and finding items the players looked at the most and similar questions.

The five experts were able to answer those questions to an extent. For example, all experts were able to find a backtracking area, which was the path between the room where the code was found and the code input terminal in front of the locked door. The experts used the trajectory visualization for this task. With the aggregated view of all players, the backtracking area was easily recognizable by the number of green lines in that area. With the eye-tracking visualization, experts were able to find the most and least looked at items very fast by looking at the color of the game objects outlines. The red outlines stood out especially and resemble objects that have been gazed at the most.

The line graph visualization helped find the most present emotion in the game. The contempt and engagement emotion were deemed as the most present, by the experts. They gained this knowledge by looking at the graph and comparing each emotion.

The last visualization was used to look for locations in the game, where players were frustrated or surprised. The surprise locations the experts found by looking for blue particle systems as depicted in the legend of the tool were located where the jump scare happened and in front of the code input terminal. They assumed players were surprised either because they entered the wrong code and were expecting it to be right or the opened the door finally just to see the zombie outside running at them. The answer to the main research question would be that it proved to be useful to visualize the emotions with particle systems directly in the game level to create easily recognizable hot spots.

More detailed information can always be shown by selecting one of those hotspots, however, getting a clear overview of the emotions occurring in the whole level was an advantage for the experts to find problem areas sooner. Expert3's statement undermines this:

> "The glowing particle systems make it easy to identify areas of high emotional impact on the players. Also finding specific emotions by looking for the corresponding color like blue for surprise was outstandingly easy."

By using the data visualizations, the experts (E4, E3, E5) came up with design decisions like being able to pick up the codesheet in the inventory to reduce backtracking and player frustration. They would also make the interaction with all doors consistent because the visualization showed that Player1 was having trouble opening the second door. Another problem Expert3 found was that Player10 had trouble jumping on the boxes which had to be done to complete the level. That is why the expert would adapt the size of the boxes to make them easier to jump onto.

Even before getting expert feedback, it was clear that the user interface (UI) of the tool is not great. The UI was not a focus of this project, and it technically works. Since this is just a proof of concept, there is no need to improve the UI, unless something is understood wrong. In the future, if this tool is transforming into a product, it is without question necessary to rework the whole UI and Aesthetic. Some experts mentioned they want to know how exactly a particular emotion gets tracked. This, however, is currently a black box by the Affectiva SDK, in the future, this SDK can be replaced by any other emotion tracking tool. It was wished-for to have more interconnectivity behind the actual emotion particle visualization and the inspector UI. For example, clicking on the particle automatically selects the emotion from the inspector list and vise Versa. This sounds like a helpful change and should be considered to be included. The most wished-for feature was to include a time scrubbing system. Each visualization should have the option to show only data of a specific time, that can be scrubbed like on a video player. This is a fantastic idea and would help a lot in understanding how the player progressed through the game.

Appendix A

DVD+RW Contents

Format: DVD+RW, Single Layer, ISO9660-Format

A.1 PDF-Files

Path: /pdf_files

- thesis.pdf . . . . Digital version of this master thesis
- questionaire.pdf . . . . Digital version of all the questionaires and surveys

A.2 Project-Files

Path: /project

- unity_project . . . . . . Unity project
- data . . . . Playtesting data files
- screens . . . . Sceenshots and videos

A.3 Online Sources

Path: /online_sources

- literature . . . . Collection of online sources used in this thesis

Appendix B

Questionnaire

This chapter contains of all the questionnaires and surveys that the experts and participants of the playtest session filled out.

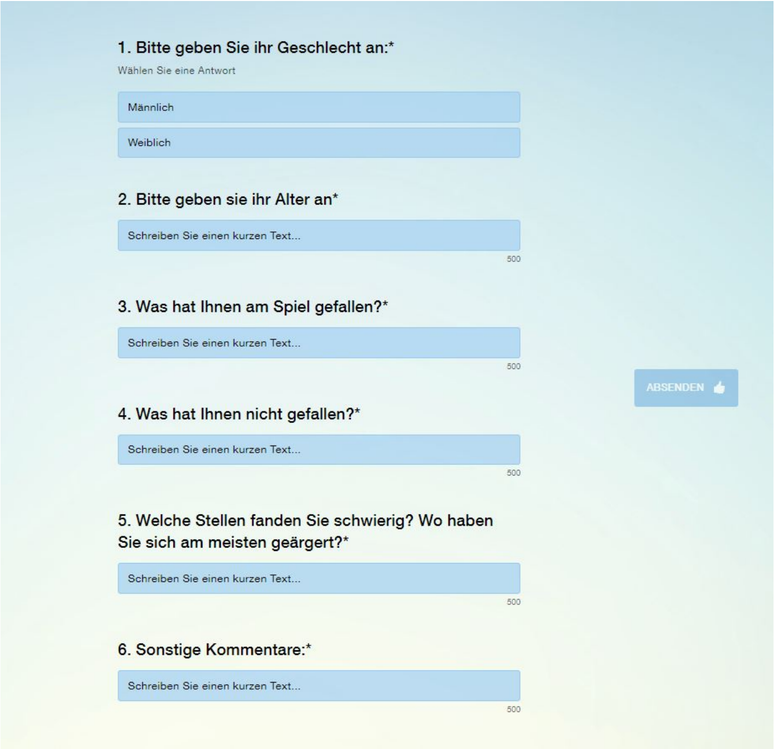


Figure B.1: This image shows the questions on the survey that was filled out by the playtesters.

Personal Data 1/2:

Sex:
☐ Female
☐ Male

Age:

Years of game development experience:

Job Description:

Experience in the Game Industry. On which kind of Games did you work?

Which test methods are you using to analyze your players/games?

Personal Data 2/2:

Do you already have experience in analyzing player data? If so, do you use any visualizations for that purpose?

How important is analyzing player data in your opinion and why?

1 not important	2	3	4	5 very important
☐	☐	☐	☐	☐

Tasksheet 1/5

T1a: Open the trajectory tool.

T1b: Find areas where players struggled. Based on the visualization, what do you think caused these problems? What changes would you suggest to fix these issues?

How helpful was the visualization for this task?

1 Not helpful	2	3	4	5 Very helpful
☐	☐	☐	☐	☐

T1c: Are there any other problems that show up when looking at only a single player's data? If so, why do you think those problems emerged?

How helpful was the visualization for this task?

1 Not helpful	2	3	4	5 Very helpful
☐	☐	☐	☐	☐

Tasksheet 2/5

T2a: Open the eye-tracking tool.

T2b: Find the objects all players looked at the most. How did you recognize those objects? Why do you think players looked at them the most?

How helpful was the visualization for this task?

1 Not helpful	2	3	4	5 Very helpful
☐	☐	☐	☐	☐

T2c: Find the objects players looked at the least / did not look at all. How did you recognize those objects? Why do you think that the player looked at them the least?

How helpful was the visualization for this task?

1 Not helpful	2	3	4	5 Very helpful
☐	☐	☐	☐	☐

.

Tasksheet 3/5

T3a: Open the emotion graph tool.

T3b: Find the emotion of player 03, that was present the most during the level. Why do you think that is the case?

How helpful was the visualization for this task?

1 Not helpful	2	3	4	5 Very helpful
☐	☐	☐	☐	☐

Tasksheet 4/5

T4a: Open the emotion visualization tool.

T4b: Find locations where you think the player 03 was surprised. How do you recognize them and what could be the reason for them to be surprised?

How helpful was the visualization for this task?

1 Not helpful	2	3	4	5 Very helpful
☐	☐	☐	☐	☐

T4c: Find locations where you think the player 04 was frustrated. How did you identify the locations, and what could be the reason for the frustration?

How helpful was the visualization for this task?

1 Not helpful	2	3	4	5 Very helpful
☐	☐	☐	☐	☐

Tasksheet 5/5

T4d: Check if other players felt similar emotions on those locations. If they are not similar, what could be the reason for this?

How helpful was the visualization for this task?

1 Not helpful	2	3	4	5 Very helpful
☐	☐	☐	☐	☐

T5: Look at all three visualizations at once (eyemovement, trajectory and emotion). Does the combination confirm problem areas, or do they show independent problems? Can you see any contradicting data?

How helpful was the visualization for this task?

1 Not helpful	2	3	4	5 Very helpful
☐	☐	☐	☐	☐

Visualization-Evaluation

Please use the following reply options in order to express your opinion on the visualizations. Please check the boxes that resemble your opinion the most.

A1: Clarity (Is the displayed data clearly interpretable or ambiguous?)

1 Poor	2	3	4	5 Excellent
☐	☐	☐	☐	☐

A2: Readability (Are the visual elements easily legible and distinguishable?)

1 Poor	2	3	4	5 Excellent
☐	☐	☐	☐	☐

A3: Informativeness (Does it provide interesting or new information?)

1 Poor	2	3	4	5 Excellent
☐	☐	☐	☐	☐

A4: Aesthetics (Is it visually appealing?)

1 Poor	2	3	4	5 Excellent
☐	☐	☐	☐	☐

A5: Accurateness (Is the displayed data accurate enough?)

1 Poor	2	3	4	5 Excellent
☐	☐	☐	☐	☐

A6: Usefulness (For which tasks is it useful?)

1 Poor	2	3	4	5 Excellent
☐	☐	☐	☐	☐

Feedback Sheet 1/2:

Which features did you like most and why? Please try to name at least three.

Which part of the tool needs the most improvement? Try to name at least three. Also, what kind of improvements do you suggest?

What problems on the game-level did you recognize with the help of this tool? Did you spot design errors on the game?

Was the tool helpful in analyzing the gameplay data? Which visualization was the most helpful and why? (Eyedata View, Trajectory View or Emotion Visualization)

Feedback Sheet 2/2:

Would you use the tool (in an improved and polished state) in the future? If yes, how would you use it and where do you see use in it? If not, why not?

How clear was it looking at all visualizations (Trajectory, Emotion, Eyetracking) at once? Where did you see problems? Do you see a possible solution to those problems?

Other Comments

Thank you!

References

Literature

[1] Gennady Adrienko and Natalia Adrienko. "Spatial Generalization and Aggregation of Massive Movement Data". *IEEE Transactions on Visualization and Computer Graphics* 17.2 (Feb. 2011), pp. 205–219 (cit. on pp. 6, 8, 22).

[2] Gennady Andrienko and Natalia Andrienko. "Spatio-temporal aggregation for visual analysis of movements". In: *2008 IEEE Symposium on Visual Analytics Science and Technology*. Nov. 2008, pp. 51–58 (cit. on p. 7).

[3] Gennady Andrienko et al. "Space, Time and Visual Analytics". *International Journal of Geographical Information Science* 24 (Oct. 2010), pp. 1577–1600 (cit. on p. 5).

[4] Gennady Andrienko et al. *Visual Analytics of Movement*. Springer, 2013 (cit. on pp. 5, 6).

[5] G. Andrienko et al. "Visual Analytics Methodology for Eye Movement Studies". *IEEE Transactions on Visualization and Computer Graphics* 18.12 (Dec. 2012), pp. 2889–2898 (cit. on pp. 7–9).

[6] Soheila Ashkezari-Toussi, Mohammad Kamel, and Hadi Sadoghi-Yazdi. "Emotional maps based on social networks data to analyze cities emotional structure and measure their emotional similarity". *Cities* 86 (2019), pp. 113–124 (cit. on p. 10).

[7] Regina Bernhaupt. *Evaluating User Experience in Games: Concepts and Methods*. 1st. Springer Publishing Company, Incorporated, 2010 (cit. on pp. 1, 3).

[8] Michael Burch et al. "A Task-Based View on the Visual Analysis of Eye-Tracking Data". In: *Eye Tracking and Visualization*. Springer, Sept. 2017, pp. 3–22 (cit. on p. 8).

[9] Alexandre Denis et al. "Visualization of affect in movie scripts". In: *Empatex, 1st International Workshop on Empathic Television Experiences at TVX 2014*. June 2014 (cit. on pp. 10, 11).

[10] Priyesh N Dixit and G Michael Youngblood. "Understanding playtest data through visual data mining in interactive 3D environments". In: *Proceedings of the 12th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia and Serious Games (CGAMES)*. 2008, pp. 34–42 (cit. on pp. 1, 4).

[11] Brandon Drenikow and Pejman Mirza-Babaei. “Vixen: Interactive Visualization of Gameplay Experiences”. In: *Proceedings of the International Conference on the Foundations of Digital Games.* (Hyannis, MA, USA,). Ed. by S. Deterding et al. ACM, Aug. 2017, pp. 1–10 (cit. on pp. 1, 3, 4).

[12] Humphreys G. Hoobler N. and Agrawala M. “Visualizing competitive behaviors in multi-user virtual environments”. *In Proc. VIS 2004, IEEE Computer Society* (2004), pp. 163–170 (cit. on p. 4).

[13] Nate Hoobler, Greg Humphreys, and Maneesh Agrawala. “Visualizing Competitive Behaviors in Multi-User Virtual Environments”. In: *Proceedings of the Conference on Visualization ’04*. VIS ’04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 163–170 (cit. on p. 14).

[14] Yun-En Liu et al. “Feature-based Projections for Effective Playtrace Analysis”. In: *Proceedings of the 6th International Conference on Foundations of Digital Games.* New York, NY, USA: ACM, 2011, pp. 69–76 (cit. on p. 1).

[15] Nirvana Meratnia and Rolf A. de By. “Aggregation and Comparison of Trajectories”. In: *GIS ’02: Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems.* GIS ’02. 605022. McLean, Virginia, USA: ACM, 2002, pp. 49–54 (cit. on pp. 6, 7).

[16] Pejman Mirza-Babaei. “Biometric storyboards: a games user research approach for improving qualitative evaluations of player experience”. PhD thesis. University of Sussex, Mar. 2014 (cit. on p. 1).

[17] Robert Plutchik. “The Nature of Emotions”. *American Scientist* 89 (Jan. 2001), p. 344 (cit. on pp. 10, 12).

[18] Günter Wallner, Nour Halabi, and Pejman Mirza-Babaei. “Aggregated Visualization of Playtesting Data”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems.* CHI ’19. Glasgow, Scotland Uk: ACM, 2019, 363:1–363:12 (cit. on p. 41).

[19] Günter Wallner and Simone Kriglstein. “Visualization-based analysis of gameplay data - A review of literature”. *Entertainment Computing* 4 (Aug. 2013), pp. 143–155 (cit. on pp. 1, 3).

[20] Günter Wallner and Simone Kriglstein. “Visualizations for Retrospective Analysis of Battles in Team-based Combat Games: A User Study”. In: *CHI PLAY*. Oct. 2016, pp. 22–32 (cit. on p. 41).

[21] Günter Wallner et al. “Unified Visualization of Quantitative and Qualitative Playtesting Data”. In: *CHI ’14: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* Toronto, Ontario, Canada: ACM, 2014 (cit. on pp. 3–5, 13).

Software

[22] Affectiva. *Affectiva SDK*. 2016. URL: https://www.affectiva.com (cit. on pp. 20, 36).

[23] Capcom. *Resident Evil 4*. 2005. URL: http://www.residentevil.com/4/us/ (cit. on p. 17).

[24] Thunderwire Games. *Horror FPS Kit*. 2017. URL: https://www.twgamesdev.com/pages/projects/horror-fps-kit (cit. on pp. 17–19).

[25] Eric Heijnen. *Cheat Engine*. 2012. URL: https://www.cheatengine.org/ (cit. on p. 17).

[26] Fraunhofer IIS. *Shore*. URL: https://www.iis.fraunhofer.de/de/ff/sse/ils/tech/shore-facedetection.html (cit. on p. 10).

[27] Unity Technologies. *Unity Engine*. 2005. URL: https://unity3d.com/de (cit. on p. 4).

[28] Tobii. *Tobii EyeTracker*. 2016. URL: https://gaming.tobii.com/products/ (cit. on pp. 20, 36).

[29] ToolbeltTechnology. *Vixen*. 2016. URL: http://vixen.launchrock.com/ (cit. on p. 4).

Online sources

[30] Responsive Spaces GmbH. *Dominator VR Experience*. 2015. URL: https://www.re-spaces.com/showroom/boot-dominator-experience (cit. on p. 3).

[31] Sita Jodi and Redmond Sean. *What eye tracking tells us about the way we watch films*. 1999. URL: http://theconversation.com/what-eye-tracking-tells-us-about-the-way-we-watch-films-19444 (visited on 04/02/2019) (cit. on pp. 8, 9).

[32] Lydia van der Spek. *Visualizing Emotions: Turning 'Dry' Data Into Stories*. 2013. URL: https://mastersofmedia.hum.uva.nl/blog/2013/03/06/visualizing-emotions-turning-dry-data-into-stories/ (visited on 04/02/2019) (cit. on pp. 11, 12).

[33] Clive Thompson. *Halo 3: How Microsoft Labs Invented a New Science of Play*. 2007. URL: https://www.wired.com/2007/08/ff-halo-2/ (visited on 08/02/2019) (cit. on p. 1).

[34] Wikipedia contributors. *Centripetal Catmull–Rom spline — Wikipedia, The Free Encyclopedia*. 2019. URL: https://en.wikipedia.org/w/index.php?title=Centripetal_Catmull%E2%80%93Rom_spline&oldid=914658410 (cit. on p. 25).