2022

MASTER'S THESIS

A 3D-Game for Detecting the Player Group With the Most Similar Playing Behaviors

ACADEMIC SUPERVISOR: Ruck Thawonmas

Graduate School of Information Science and Engineering
Ritsumeikan University

MASTER'S PROGRAM
MAJOR in Advanced Information Science and Engineering

STUDENT ID: 6612200093-2

NAME: Sangvanloy Tannop

Abstract

This thesis aims to present an alternative way to group players from the most similar playing behaviors. However the traditional means can be time-consuming while partially invasive means to assess human influence factor-like personality, our game requires lesser time, is more entertaining, and requires no human influence factor-like personality. The game was made based on a paper cognitive assessment test(Immediate Recall, Trail A, Stroop, Digit Symbol Substitution, and Delayed Recall) which were then gamified into five different levels. Then we separate each player by collecting players' game-play data such as the time used to complete the task or how the player chooses to shoot the target.

Three participants were chosen from eight different laboratories at Ritsumeikan University for a total of twenty-four to partake in the experiment. Each participant then played the game once to collect the data. The main purpose of this experiment is to find if we could separate the participants into each group based on the player's play data. The experiment shows that the record of participant's data allows for the prediction of the group significantly better than chance level.

Keywords— Enjoyment, Playing behaviors, Cognitive, Neuropsychological assessment

Acknowledgement

Firstly, I would like to express my deepest gratitude to my supervisor, Prof. Ruck Thawonmas. This thesis would have not been possible without all the support and guidance. He provided great opportunities for not just me but others in the laboratory to increase our knowledge such as hosting Deep learning seminars which also helped me in doing a lot of research and I came to know about so many new things.

I am very grateful for the advice and feedback offered by the senior researcher Pujana Paliyawan, who was also involved in my research project. He always helps me with many things, not just the research project.

I also would like to express special thanks to Prof. Kingkarn Sookhanaphibarn and Prof. Worawat Choensawat from Bangkok University in Thailand for sharing this great opportunity to study at Ritsumeikan University and assisting during my research thesis.

Finally, I would like to express my gratitude to the other seven laboratories for contributing to my experiment.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Research Background

It appears that human influence factors have a significant impact on how users perceive the quality of the system, and personality is one of the most important factors in understanding human behavior, which is why personality has stood out as an important predictor [1]. It can predict individuals' abilities in their learning approaches, academic success, intelligence, or ratings of subjective well-being [2]. Personality psychology is built on assumptions that each characteristic could be displayed in stable behavioral patterns which could be captured in different terms [1, 3].

Many companies also use pre-employment with cognitive assessments for screening of high potential employees[4–6]. Cognitive assessments are used to measure a candidate's cognitive skills, they helped identify better candidates and speed up the process of requirement. It doesn't measure individual knowledge about something but rather how they think. Most of the cognitive ability tests are designed to measure mental skills such as:

- Attention to details: Evaluates candidates' ability to pay attention to textual detail while processing information.
- Problem solving: Evaluates candidates' ability to define problems and analyze data and textual information to make correct decisions.
- Critical thinking: Evaluates candidates' skills in critical thinking through inductive and deductive reasoning problems.
- Numerical reasoning: Evaluates candidates' general aptitude with numbers and their skill in interpreting them for a wide range of applications.
- Reading comprehension: Evaluates candidates' ability to read a portion of text and comprehend its contents.
- Spatial reasoning: Evaluates a candidate's ability to analyze 2-dimensional and 3-dimensional objects and space.

The traditional methods are self-assessment questionnaires, although these methods achieved high accuracy, they required the use of personal information and took a long period to complete. In addition to being more entertaining, our method takes less time and is easier to obtain necessary information.

1.2 Research purpose

This thesis presents an alternative way to group players from the most similar playing behaviors by using the play data collect from the game. Similar to this paper [1], data were collected by the game we created based on

BrainCheck's cognitive assessment battery [7] with a three-dimensional environment (3D environment) which also be able to be effective in promoting player engagement [8].

Chapter 2

Related Work

2.1 Spell

Spell is a spelling game, that is part of the TU-Berlin developed PflegeTab application which seeks to improve the quality of life for people with dementia [9] usually played on a tablet device. The game task is to spell a word by dragging letters that were randomly scattered around the screen to the correct position below the screen. The player must tap and hold down then drag to the correct position. The game continues with a new random word after the initial word is complete and spelled correctly.

The author of the paper [1] used data collected from the Spell game. Collected data other than the player's performance such as the time difference between a finger down and a finger up event, the length and speed of swipes, the frequency of taps, and the touch accuracy. In sum, server records of performance and touch behavior are transformed into 110 variables. However, the author only uses the test for detecting players' personality based on their playing data and use it to predict Big-5 [10] personalities trait, and only had a 2D format.

(a) Spell (Buchstabieren)

Figure 2.1: Screenshots of the Spell game.

2.2 BrainCheck

BrainCheck test battery [7] is a set of cognitive assessments based on commonly including instruments in the neuropsychological screening test. The current version of BrainCheck is available on the website[1], which consists of five assessments: Immediate Recall, Trail Making Test, Stroop Test, Digit Symbol Substitution Task, and Delayed Recall. However, the test was only used to diagnose age-related cognitive impairment, and the tasks only had a 2D format.

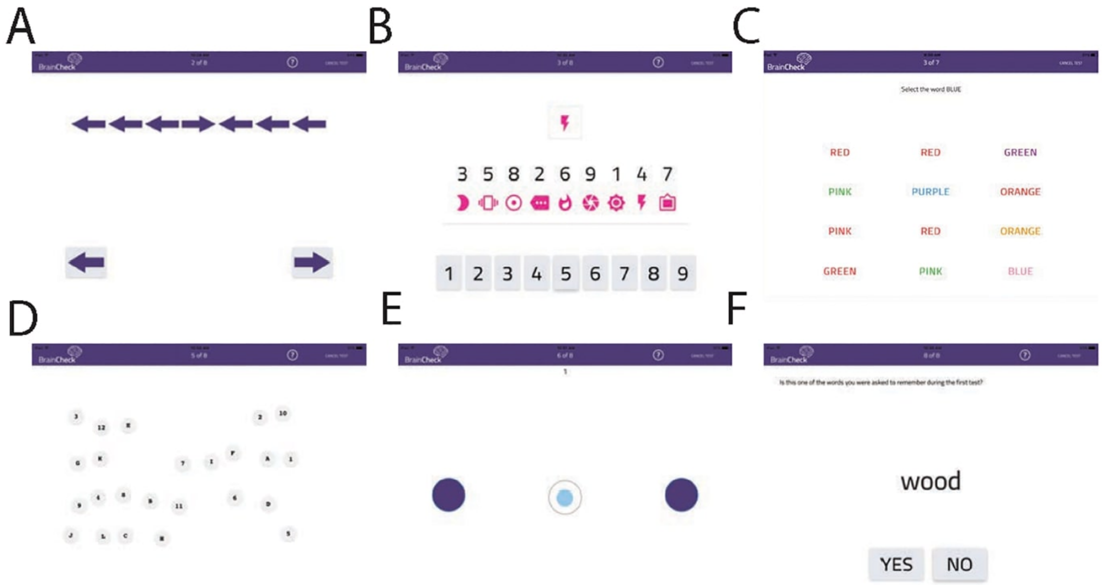

Figure 2.2: Screenshots of the 6 assessments in the BrainCheck battery.

[1]https://braincheck.com/

Chapter 3

Proposed Concept and Game

This chapter explains the game in detail. We believe that we could classify players into different groups based on their game data. The proposed game was created based on five different cognitive assessments similar to BrainCheck's battery test (Immediate Recall, Trail Making Test A, Stroop Test, Digit Symbol Substitution Task, and Delayed Recall) which were used to assess player's cognitive function. The game was developed in a 3D environment and there is a total of 5 different levels. The proposed game system is a game composed of five different levels, each level represents each cognitive test. Most of the games are the First-Person game, only level 2 is a Top-Down game.

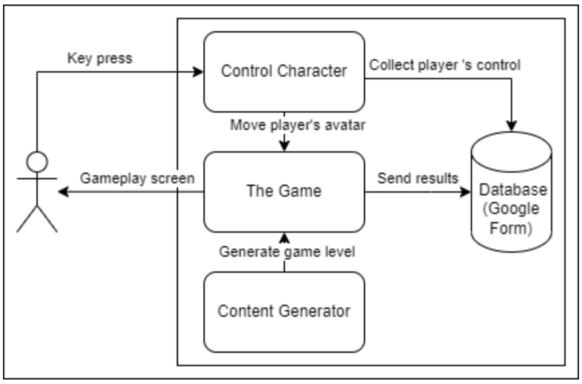

Figure 3.1: System overview.

3.1 Unity and WebGL

In this research, we used Unity and C# programming language to create the game. Unity or Unity3D is a game engine and integrated development environment (IDE) for creating interactive media, typically video games [11], and to make the game playable on web browsers, the need to be made with WebGL (Web Graphic library) build setting. Lastly, to collect all data, we connect the game with Google Form and export the data in CVS format to use with WEKA (Waikato Environment for Knowledge Analysis).

WebGL or Web Graphic library [12, 13] is a Javascript API that enables web-browse to display, draw, and interact without additional plugins. Almost all browsers support WebGL. With the WebGL package provided by unity, it allows publishing the game which is compatible with HTML5/JavaScript and able to run content in a web browser. The downside to using WebGL is that we cannot use a custom .dll file, making preprocessing inside the game with WEKA.dll tools not possible [14, 15]. The WEKA.dll file is a .dll file that was converted from a Java file with IKVM.NET [16]. The solution to this is to process the data outside the game, but building the game in normal Windows/Mac/Linux format doesn't have the said problem.

3.2 WEKA

Data preprocessing and machine learning algorithms are included in WEKA [17]. With it, users can experiment with and compare different machine learning methods on new data sets. The application includes algorithms for regression, classification, clustering, association rule mining, and also attribute selection with an interface that enables easy access to the underlying functionality.

3.3 Basic Control

3.3.1 First-Person

Games with the First-Person perspective rendered from the viewpoint of the player's character mean that the game displays what the player's character would see with the character's own eyes. Usually, the player will not see most parts of the character's body except for the character's limbs (arms, legs) and object in the hand [18]. Game Controls in the game are shown in Table 3.1.

Table 3.1: First-Person Controls

Input	Action
Key W	Move the player's character forward.
Key A	Move the player's character left.
Key S	Move the player's character backward.
Key D	Move the player's character right.
Arrow-Key Up:	Move the player's character forward.
Arrow-Key Left	Move the player's character left.
Arrow-Key Right:	Move the player's character right.
Arrow-Key Down:	Move the player's character backward.
Key Space-Bar	Jump to answer.
Mouse Left-Click	Shoot the gun at the aimed point in the middle of the screen.

3.3.2 Top-Down

Top-Down perspective, also sometimes referred to as Bird's-Eye view, usually refers to a camera angle that shows players and the areas around them from above. This camera angle will display the player's character and regions around them similar to a Third-Person view. Game Controls in the game are shown in Table 3.2.

Table 3.2: Top-Down Controls

Input	Action
Key W	Move the player's character forward.
Key A	Move the player's character left.
Key S	Move the player's character backward.
Key D	Move the player's character right.
Arrow-Key Up:	Move the player's character forward.
Arrow-Key Left	Move the player's character left.
Arrow-Key Right:	Move the player's character right.
Arrow-Key Down:	Move the player's character backward.

3.4 Game levels

There are five levels, each designed to be a different assessment following BrainCheck's battery assessment [19]. All of the game levels were designed to follow the procedure of the cognitive assessment task but were gamified by us.

3.4.1 Level 1: Immediate Recall

The Immediate Recall [19–21] measures the participant's ability to correctly recall seen words before and after a period of time. First Immediate Recall is measured by answering whether a word was just seen or not from a list of 20 words.

This level was designed to be a shooting game in a First-Person view. First, after instruction dialog Fig.3.2 and Fig.3.3 closed, player were presented with total of 10 words on the board to remember. After 15 seconds, the board will be hidden and a total of 20-word lanterns will appear in the sky, the words attached to the lanterns are a mix between the previous 10 words and another 10 random words. To answer, players are required to look around and use the mouse left-click button to shoot down lanterns with words previously shown on the board as many as possible in 20 seconds, the more the better.

To ensure that the correct answer word lanterns will be spread evenly thought out the sky, first, we group the lanterns in 4 different cluster zones seen in Fig.3.8 then use the Algorithm 1 to assign each lantern with the correct and wrong answer.

Algorithm 1 Algorithm of Level 1 words and lantern clusters

```
Require: N= 2, 2, 3, 3
if Wordlist.txt is not empty then
    Random 30 words from Wordlist.txt
    if Successfully random 30 words from Wordlist.txt then
        Random and remove 10 words from 30-word list
        if Successfully random and remove 10 words from 30-word list then
            Set all lanterns with leftover words from the word list
            Back up for 10 correct words
        end if
    end if
    for all 4 cluster do
        Randomly assign and delete the number of correct words from N list to each cluster
        Random choose correct answer position inside each cluster
        Replace and delete correct answer position lanterns with correct words from the correct word list
    end for
end if
```

Algorithm 2 Algorithm of Shooting

```
if Fire button is pressed then
    Start Shot-Effect co-routine to turn our laser line on and off
    Create a vector at the center of our camera's viewport
    Declare a raycast hit to store information about what our raycast has hit
    Set the start position for our visual effect for our laser to the position of gunEnd
    if Raycast has hit anything then
        Set the end position for our laser line
        Get a reference to a health script attached to the collider we hit
        Saved previous target shooted number for reference
        if The collider is the target then
            if The target contains correct word then
                Player gain 1 score
                Total shoot +1
            else
                Total shoot +1
            end if
        end if
        Save target number shot
        if Player shot to the left then
            Left +1
        else
            Right +1
        end if
    end if
end if
```

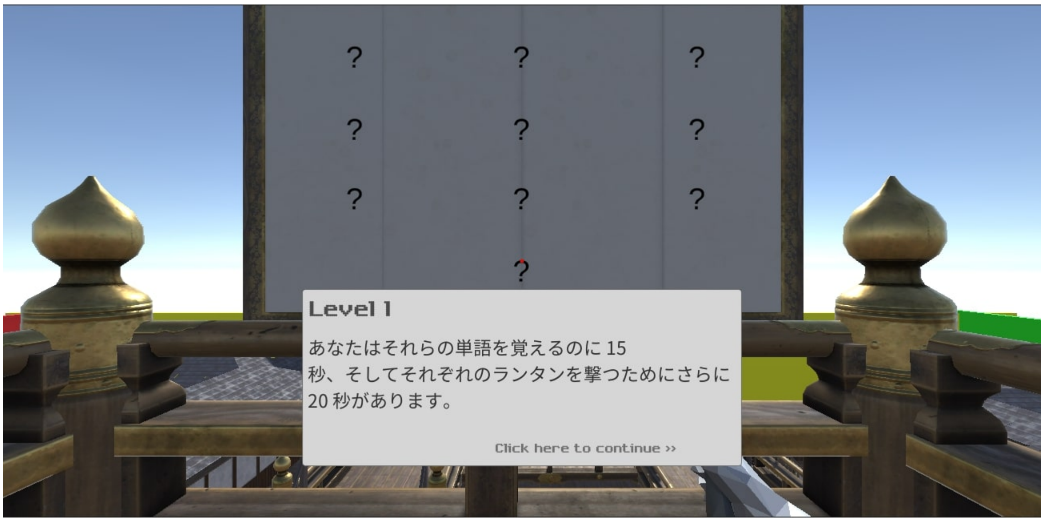

Figure 3.2: First instruction dialog from level 1.

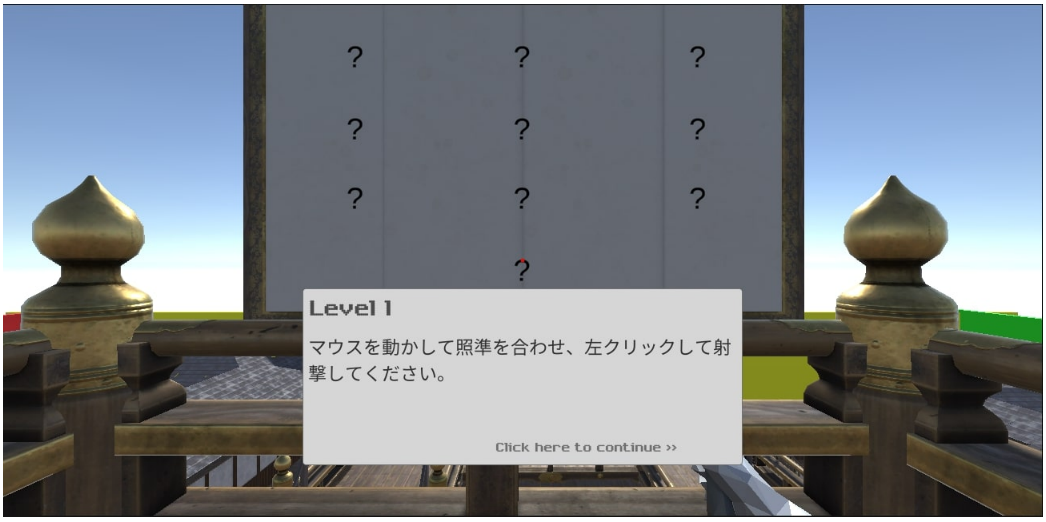

Figure 3.3: Second instruction dialog from level 1.

Figure 3.4: Ten words are shown after the dialog is completed.

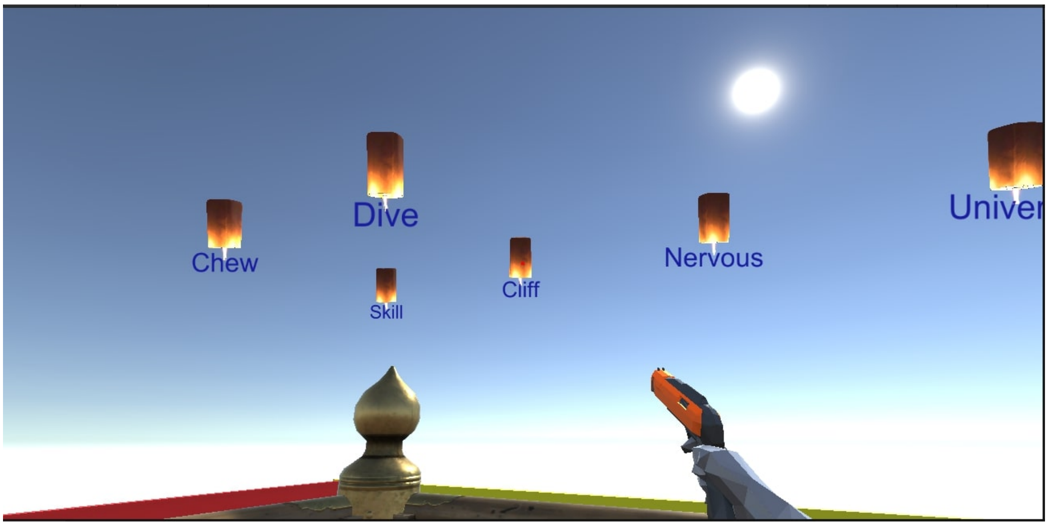

Figure 3.5: Word lantern targets.

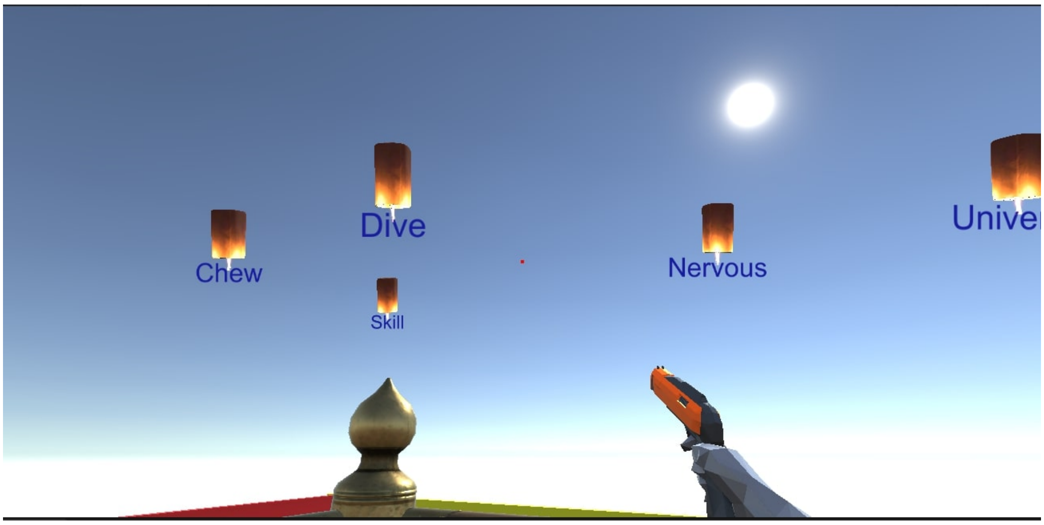

Figure 3.6: A lantern with the word "Cliff" was shot.

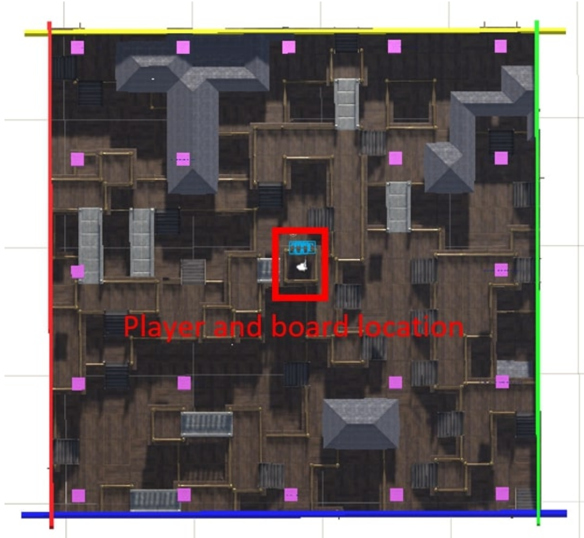

Figure 3.7: Location of the player and the board.

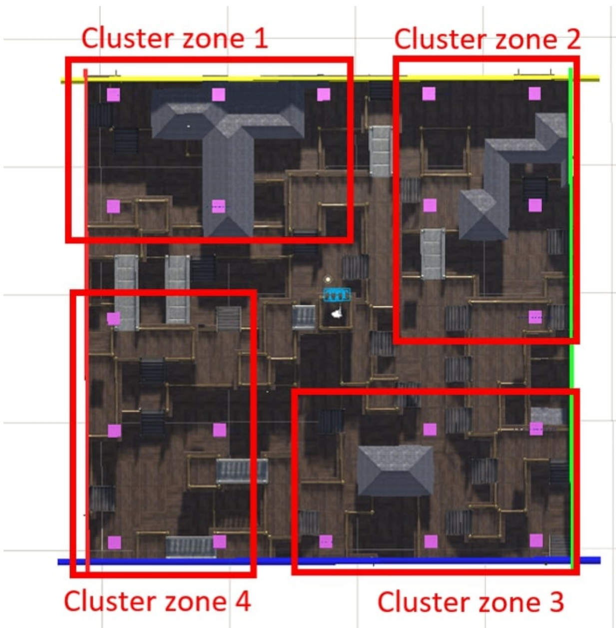

Figure 3.8: 4 Different cluster zone.

3.4.2 Level 2: Trail Making Test A

Trail Making Test [19, 22] is a neuropsychological test of visual attention and task switching. The participants are instructed to connect 25 numbers from 1 to 25 as fast as possible. The task measures visual search speed, scanning, speed of processing, mental flexibility, and executive functioning. Trail Making Test A uses only numbers (1 to 25) while Trail Making Test B uses letters and numbers (1-A-2-B-3-C-...).

At first, this level was made with a first-person view. After testing and playing, we discovered that this task was too challenging and took a long time to complete. The solution to this problem is to change the game view, we change from a First-person view to a top-down view, and reduce the number of the number targets. Changing to a Top-down view allows players to see the entire map of the numbers while in the First-person view, players have to remember the number's route and location which doesn't correspond to the paper cognitive test. several level, after instruction dialog Fig.3.9 and Fig.3.10 closed, player have to control player character which is the red cube shown in Fig. 3.11 by using control in the Table 3.2. Players have to collect numbers in order from 1 to 5 as fastest as possible. There's also an AI that was used for normalizing the score.

Wave Function Collapse

The code used for generating the level 2 map was made by marian42 [23]. The algorithm for generating the level map is called WFC (Wave Function Collapse) [24]. Modules are blocks with structures that the algorithm selects for each "slot" (Slot is a place in the 3D voxel grid that can contain a module or can be empty), so each slot has a set of possible modules it can contain. At first, the map starts in a completely unobserved state where every module could be in any slot then one by one, each slot is collapsed, which means, one module from the set of possible modules is selected at random. For each module, only several modules are allowed to be placed next to it, and a backtracking algorithm makes the WFC algorithm able to generate infinitely without a dead end. Later, the map was cropped into a fixed size to fit the game level.

Algorithm 3 Algorithm of Level 2 Random number to random location

```
for all spawnObject do
    Random a spawn position from a list of spawnPosition
    Create randomized spawnObject at randomized position
    Removed the spawned spawnObject from the list
    Add randomized spawnObject to a list
    Save destination of the spawnObject to a list in an order
    Removed the spawned spawnObject from the list
end for
```

Algorithm 4 Algorithm of Level 2 NavMesh AI

```
Require: Distance between AI and target position
Start AI timer
if AI reached target destination then
    if Not the last destination then
        Move to next destination
    else
        Stop AI timer
    end if
end if
```

Algorithm 5 Algorithm of Level 2 Player

```
Start Player timer
if Player reached target destination then
    if Correct then
        Hide the number
        Play a sound
        Move pointer to next destination
    else
        Stop AI timer
    end if
end if
```

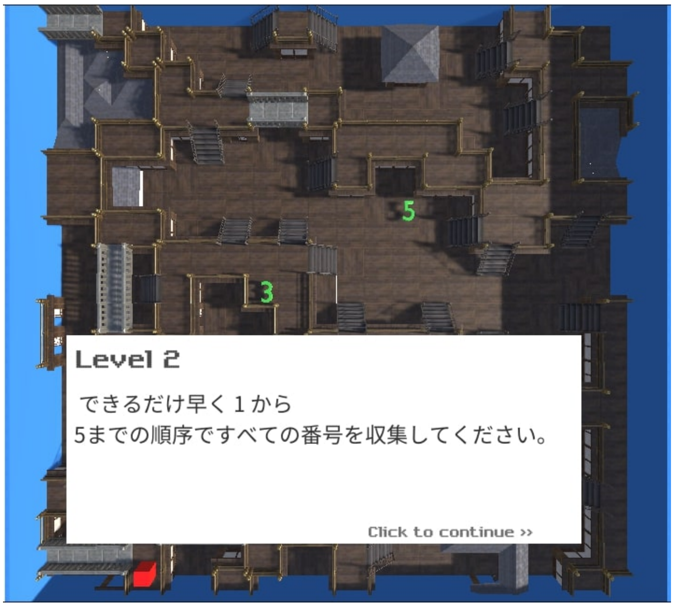

Figure 3.9: First instruction dialog from level 2.

Figure 3.10: Second instruction dialog from level 2.

Figure 3.11: Game start.

Figure 3.12: Player reaching the first number.

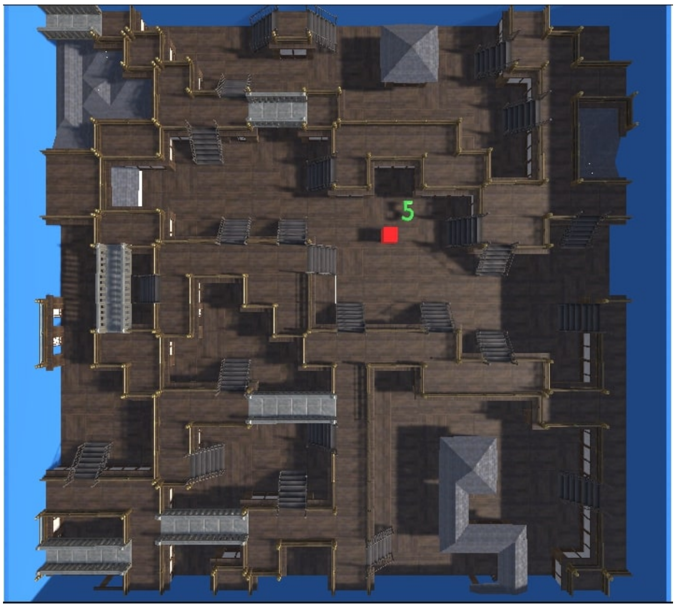

Figure 3.13: Player reaching the last number.

Figure 3.14: This figure show all possible spawn position of the number.

3.4.3 Level 3: Stroop Test

The Stroop Task [19, 25] measures the reaction time used to overcome cognitive interference. Naming the word with incongruent color takes longer and is more likely to make mistakes in contrast to when the word and color are congruent, commonly used as a measure of executive function.

In this level, after instruction dialog Fig.3.15 and Fig.3.16 closed, players will start at this position shown in Fig.3.17. Players have to search the word of the color on the floor Fig.3.18, after jumping over the correct tile, the players' screen will then move to the start location Fig.3.19. Each player will have 30 seconds to answer as much as possible.

Algorithm 6 Algorithm of Level 3 Random Color and Word

if Start level or player answered **then**
    Random and removed one answer color from the color list
    for all 12 Colors for tile floor **do**
        Random one color word from the color list
        Add the color word to the fake color array
    end for
    for all 12 Tile on the floor **do**
        Random one color from the fake color array
        Random color to the word on the floor
        Use switch case to assign a color to the word on the floor
    end for
    Random one tile for the answer location
    Get answer spawn location from the tile
    Set target position for AI to move to answer location
    Set the question board with answer color word
    Set the randomized tile with the answer color word
end if

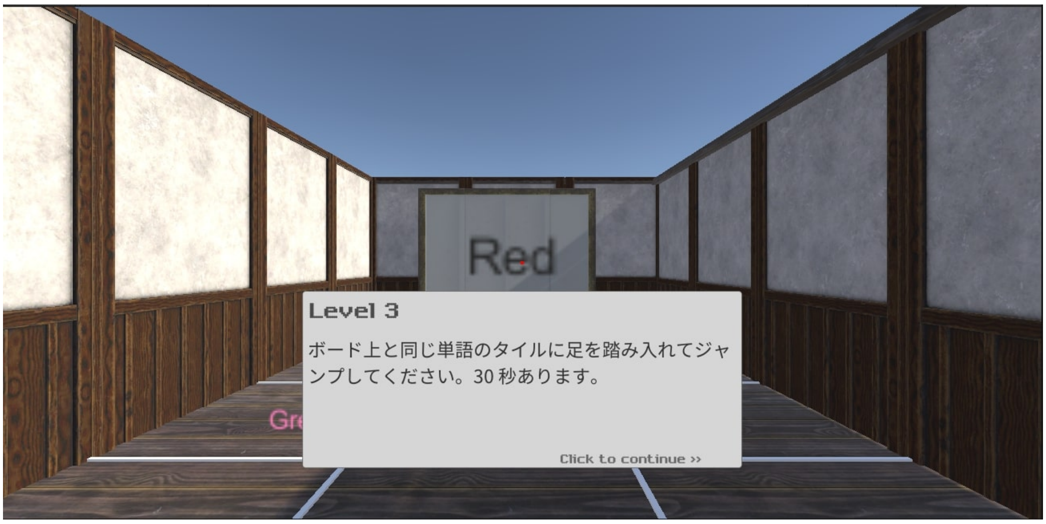

Figure 3.15: First instruction dialog from level 3.

Figure 3.16: Second instruction dialog from level 3.

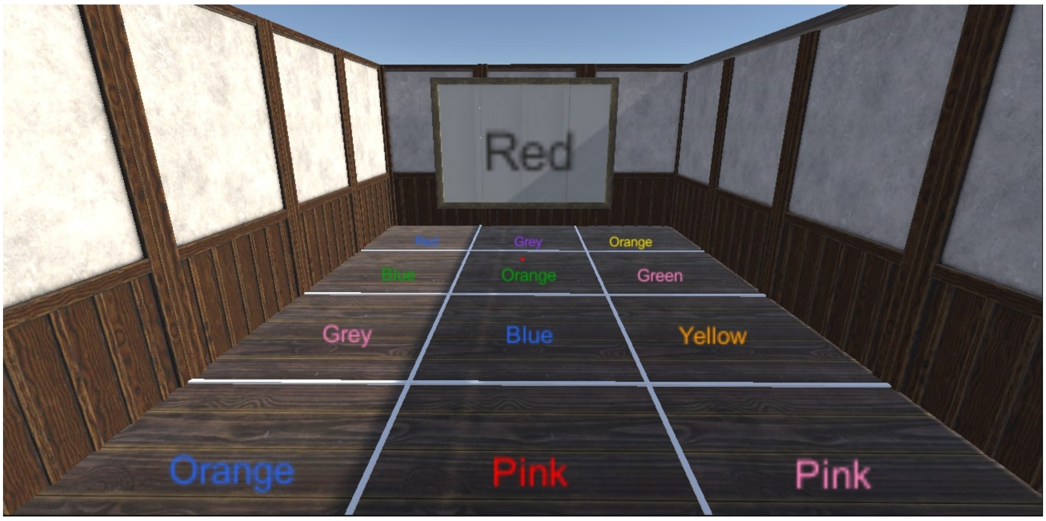

Figure 3.17: The game after the player finished the instruction dialog. The start location is where now the player is standing.

Figure 3.18: Answer tile with the corresponding word on the board.

Figure 3.19: The player has been returned to the start location after jumping over the answer tile.

3.4.4 Level 4: Digit Symbol Substitution Task

The Digit Symbol Substitution Task [19, 26] measures general cognitive performance. On the paper test, participants must match the symbol to a number quickly and correctly by pressing the number. This continues for a set duration.

Similar to level 3, after instruction dialog Fig.3.20 and Fig.3.21 closed, players will start at this position shown in Fig.3.22. Players have to search the number tile which corresponds to the symbol on the top of the board Fig.3.23, after jumping over the correct tile, the players' screen will then move to the start location Fig.3.24. Each player will have 30 seconds to answer as much as possible.

Algorithm 7 Algorithm of Level 4 Random Icon

```
if Start level or player answered then
    Random one answer symbol from the symbol list
    Random one tile for the answer location
    Get answer spawn location from the tile
    Set target position for AI to move to answer location
    Set the question board with the randomize symbol
end if
```

Figure 3.20: First instruction dialog from level 4.

Figure 3.21: Second instruction dialog from level 4.

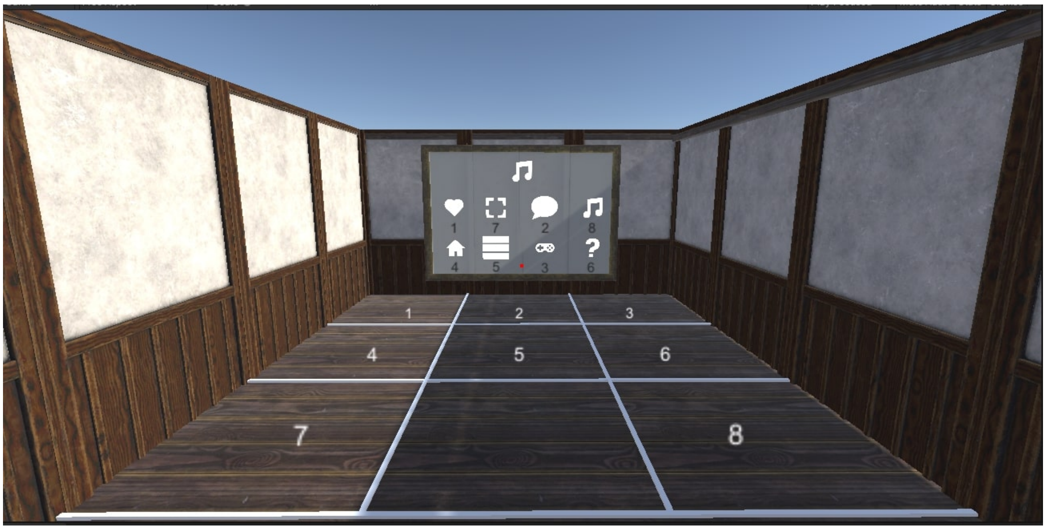

Figure 3.22: The game after the player finished the instruction dialog. The start location is where now the player is standing.

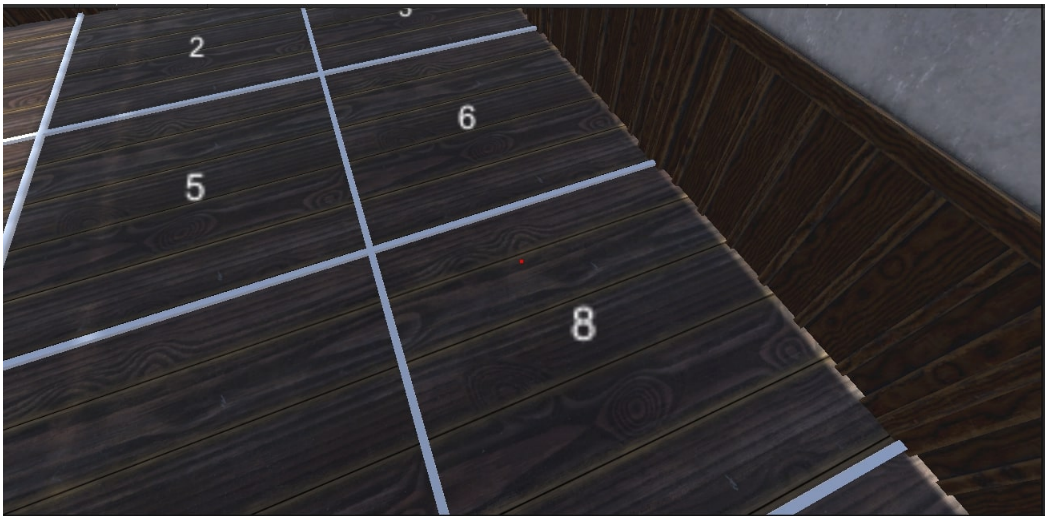

Figure 3.23: Answer tile with the corresponding number to the symbol on the board.

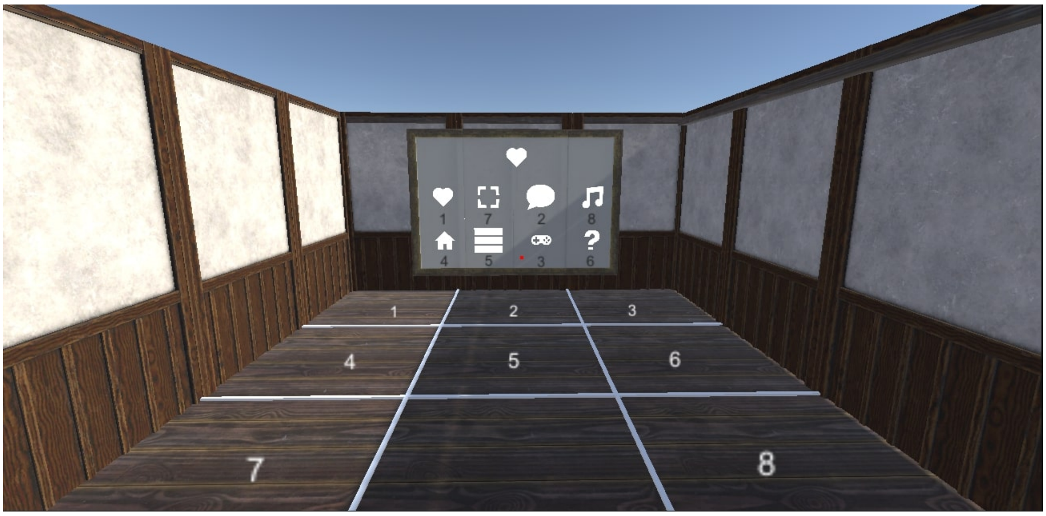

Figure 3.24: Player was moved to start location after jumping over the answer tile.

3.4.5 Level 5: Delayed Recall

Delayed Recall task measures player long-term memory [19, 21]. At the end of the battery test, without seeing the original list again, participants are presented with the 20 previous words before and asked if they have seen them or not.

Similar to level 1, after finishing the instruction dialog Fig.3.25 players have to shoot the word lantern with the same word shown previously on the level 1 board but this time players were unable to look at the word board.

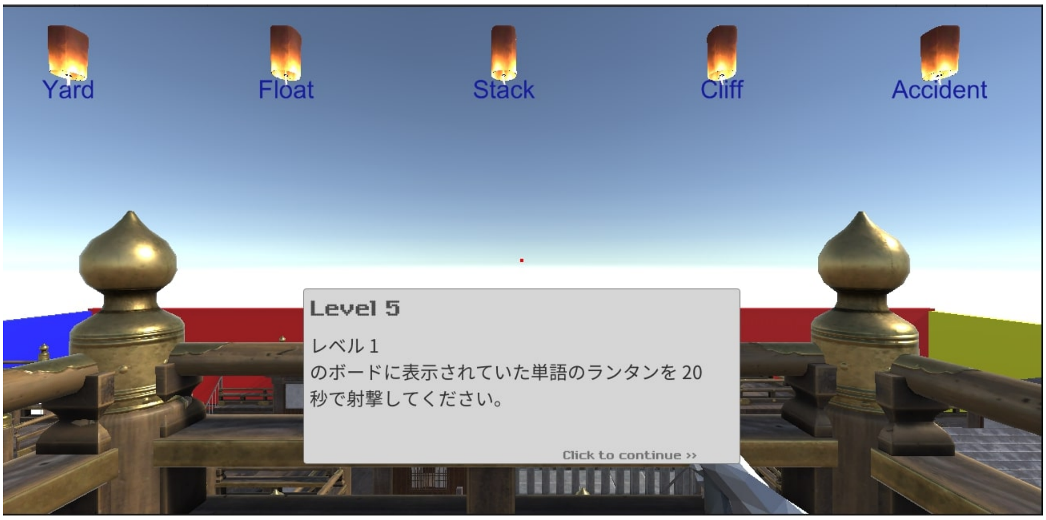

Figure 3.25: First instruction dialog from level 5.

3.5 Classification and Clustering Algorithm

Three different algorithms were introduced to the WEKA's classification of the data BayesNet, J48, and Lazy IBk.

3.5.1 BayesNet

BayesNet [27–29] is the Base class for a Bayes Network classifier which provides data structures and facilities common to Bayes Network learning algorithms. It uses Bayes's Theorem, a formula that calculates a probability by counting the frequency of values and combinations of values in the historical data.

3.5.2 J48

J48 [28–30] is an algorithm to generate a decision tree that is generated by C4.5. J48 calculates the result value of a new sample based on various attribute values of the new sample. Each node of a decision tree denotes the different attributes, the branches between the nodes show the possible values that the attributes can have in the observed samples, and the last nodes tell the classification of the new sample.

3.5.3 Lazy IBk

The Lazy Ibk [29, 31] classification algorithm is a k-nearest-neighbor classifier that uses the closest Euclidean distance metric. Predictions from more than one neighbor can be weighted according to their distance from the test instance and two different formulas are implemented for converting the distance into a weight.

3.5.4 SimpleKMeans

SimpleKMeans [32] is a clustering algorithm which uses k means algorithm. It is a simple unsupervised learning algorithm. Data objects ('n') are grouped into k clusters, with each observation belonging to the cluster with the closest mean. It can use either the Euclidean distance (default) or the Manhattan distance.

Chapter 4

Experiments and Results

The goal of this experiment is to verify if we could group players from their play data. We collected the play data from participants in an experiment. The data is then normalized and then fed into classification algorithms to group players from the most similar playing behaviors.

4.1 Participants and Procedure

3 healthy representatives of the laboratories are chosen from every 8 different laboratories at Ritsumeikan University, there are 24 participants in total. All 24 participants were asked to play the game once to collect the necessary data. Players must complete all 5 different levels from 1 to 5 in order. Each level was designed to capture different scores and features of players.

The flow of the game is shown in this figure Fig.4.1

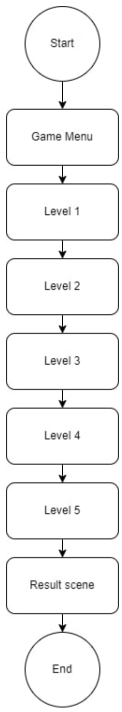

Figure 4.1: The game flow chart.

4.2 Result

All participants' data were included in the final data set with 10 different key features used in classification algorithms. The 10 key features are shown in Table 4.1.

Table 4.1: 10 Key features

Level	Data collected
Level 1	Shoot direction
Level 1	Normalized score
Level 2	Idle time
Level 2	Normalized time
Level 3	Level 3 score
Level 3	Normalized time
Level 4	Level 4 score
Level 4	Normalized time
Level 5	Shoot direction
Level 5	Normalized score

From the result shown in Fig. 4.2, Lazy IBk has the best accuracy to classify the data. Lazy IBk is a KNN algorithm, in the experiment we set the algorithm set to K=1 with 3-fold cross-validation. The result came out that we could classify players with their play data with an accuracy of 29.2% which is significantly better than chance level (12.5%).

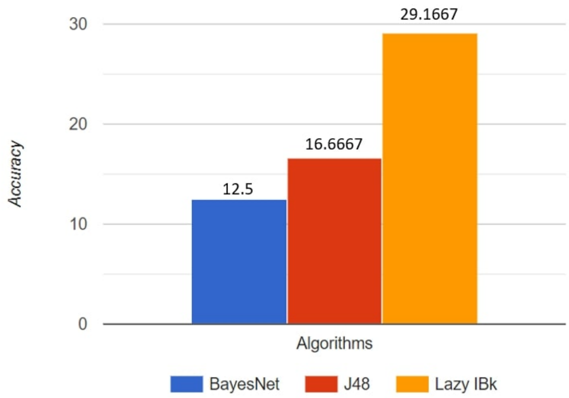

Figure 4.2: Results of 3 WEKA's algorithms on players' data.

Lazy IBk's confusion matrix Fig. 4.3 show that only 2 out of 3 participants from 3 labs were correctly classify (be, hvcs, and nips), each of these lab have 2 student with the same playing behavior, but in other lab other lab. We could not find any similar behavior because players from the other 5 lab are all unique or are similar to other lab which cause the algorithm failed to correctly classified the player.

```
=== Confusion Matrix ===

 a b c d e f g h   <-- classified as
 2 0 1 0 0 0 0 0 | a = be
 0 0 2 0 0 0 0 1 | b = cb
 0 1 0 1 0 0 0 1 | c = cns
 0 0 1 1 0 0 1 0 | d = em
 0 0 0 0 2 0 0 1 | e = hvcs
 1 1 0 0 0 0 0 1 | f = ice
 0 0 0 0 1 0 2 0 | g = nips
 0 1 0 0 2 0 0 0 | h = sys
```

Correctly classified

Figure 4.3: Lazy IBk's confusion matrix.

Result from clustering made with SimpleKMeans with k=8 are shown in Fig.4.4. From the results, 2 of the laboratories, all 3 participants were grouped in to their own group, cluster 6 for be and cluster 3 for hvcs, and 2 participants in cluster 7 are from nips. This shows that 3 participants from laboratory be and hvcs and 2 participants from nips have the same playing behaviour.

```
Class attribute: Name
Classes to Clusters:

 0 1 2 3 4 5 6 7  <-- assigned to cluster
 0 0 0 0 0 0 3 0 | be
 0 0 0 0 1 1 1 0 | cb
 0 1 0 1 1 0 0 0 | cns
 1 1 0 1 0 0 0 0 | em
 0 0 0 3 0 0 0 0 | hvcs
 0 0 0 1 0 1 1 0 | ice
 1 0 0 0 0 0 0 2 | nips
 0 0 1 1 0 0 1 0 | sys
```

Figure 4.4: Lazy IBk's confusion matrix.

Chapter 5

Conclusion and Future work

The research purpose of this thesis was to show that we could separate players into groups of players' play data. We gamified five different cognitive paper assessments into five different game levels. To collect the necessary data, players have to play the game once. We ran into some problems while implementing WEKA's .dll file into the game, which we could resolve by exporting data collected in the game into a Google form so that we could then manually feed the data into WEKA's application.

The result from WEKA's Lazy IBk classifier algorithm has an accuracy of 29.2% and is better than chance level (12.5%). These results suggest that players can be grouped based on their play data.

It might be possible to add more game levels to the current version of the game, not just from gamification of cognitive assessments, but also from normal games, to collect more diverse data in the future. We could improve our current results by collecting more data on key features. Furthermore, other algorithms for classifying game data may also improve the results.

References

[1] Ludwig Küster, Carola Trahms, and Jan-Niklas Voigt-Antons. Predicting personality traits from touch-screen based interactions. In *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6, 2018. doi: 10.1109/QoMEX.2018.8463375.

[2] Natalie Hayes and Stephen Joseph. Big 5 correlates of three measures of subjective well-being. *Personality and Individual Differences*, 34(4):723–727, 2003. ISSN 0191-8869. doi: https://doi.org/10.1016/S0191-8869(02)00057-0. URL `https://www.sciencedirect.com/science/article/pii/S0191886902000570`.

[3] Gerald Matthews, Ian J Deary, and Martha C Whiteman. *Personality traits*. Cambridge University Press, 2003.

[4] Peter A Hausdorf Manon Mireille LeBlanc and Anuradha Chawla. Cognitive ability testing and employment selection: Does test content relate to adverse impact? *Applied HRM Research*, 7(2):41–48, 2003.

[5] Paul R Sackett and Steffanie L Wilk. Within-group norming and other forms of score adjustment in preemployment testing. *American Psychologist*, 49(11):929, 1994.

[6] Cognitive ability tests: The ultimate guide for recruiters, 2022. URL `https://www.testgorilla.com/blog/cognitive-ability-tests-the-ultimate-guide-for-recruiters/`.

[7] Samantha Groppell, Karina M Soto-Ruiz, Benjamin Flores, William Dawkins, Isabella Smith, David M Eagleman, Yael Katz, et al. A rapid, mobile neurocognitive screening test to aid in identifying cognitive impairment and dementia (braincheck): Cohort study. *JMIR aging*, 2(1):e12615, 2019.

[8] N.J. Mount, C. Chambers, D. Weaver, and G. Priestnall. Learner immersion engagement in the 3d virtual world: principles emerging from the delve project. *Innovation in Teaching and Learning in Information and Computer Sciences*, 8(3):40–55, 2009. doi: 10.11120/ital.2009.08030040.

[9] JN Antons, J O'Sullivan, S Arndt, P Gellert, J Nordheim, S Mller, and A Kuhlmey. Pflegetab: Enhancing quality of life using a psychosocial internet-based intervention for residential dementia care. In *Proceedings of International Society for Research on Internet Interventions*, pages 1–1. International Society for Research on Internet Interventions, 2016.

[10] Yves-Alexandre de Montjoye, Jordi Quoidbach, Florent Robic, and Alex Sandy Pentland. Predicting personality using novel mobile phone-based metrics. In *International conference on social computing, behavioral-cultural modeling, and prediction*, pages 48–55. Springer, 2013.

[11] John K Haas. A history of the unity game engine. *Diss. WORCESTER POLYTECHNIC INSTITUTE*, 483:484, 2014.

[12] Tony Parisi. *WebGL: up and running.* " O'Reilly Media, Inc.", 2012.

[13] Webgl: 2d and 3d graphics for the web, 2022. URL `https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API#Reference`.

[14] Unity user manual 2021.3 (lts)platform developmentwebgldeveloping for webglinteraction with browser scripting, 2022. URL https://docs.unity3d.com/Manual/webgl-interactingwithbrowserscripting.html.

[15] Include c++ dll in webgl build, 2014. URL https://forum.unity.com/threads/include-c-dll-in-webgl-build.371130/.

[16] Ikvm.net in details, 2020. URL https://www.c-sharpcorner.com/UploadFile/abhijmk/ikvm-net-in-details/.

[17] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[18] Ernest Adams. *Fundamentals of game design*. Pearson Education, 2014.

[19] Suosuo Yang, Benjamin Flores, Rotem Magal, Kyrsti Harris, Jonathan Gross, Amy Ewbank, Sasha Davenport, Pablo Ormachea, Waleed Nasser, Weidong Le, W. Frank Peacock, Yael Katz, and David M. Eagleman. Diagnostic accuracy of tablet-based software for the detection of concussion. *PLOS ONE*, 12(7):1–14, 07 2017. doi: 10.1371/journal.pone.0179352. URL https://doi.org/10.1371/journal.pone.0179352.

[20] Elizabeth B Bernhardt. Testing foreign language reading comprehension: The immediate recall protocol. *Die Unterrichtspraxis/Teaching German*, 16(1):27–33, 1983.

[21] Veronica Eileen Comerford, Gina Malke Geffen, Chris May, Sarah Elizabeth Medland, and Laurence Basil Geffen. A rapid screen of the severity of mild traumatic brain injury. *Journal of Clinical and Experimental Neuropsychology*, 24(4):409–419, 2002. doi: 10.1076/jcen.24.4.409.1044. URL https://doi.org/10.1076/jcen.24.4.409.1044. PMID: 12187455.

[22] John D Corrigan and Nancy S Hinkeldey. Relationships between parts a and b of the trail making test. *Journal of clinical psychology*, 43(4):402–409, 1987.

[23] Infinite procedurally generated city with the wave function collapse algorithm, 2019. URL https://marian42.de/article/wfc/.

[24] Procedural generation with wave function collapse, 2019. URL https://www.gridbugs.org/wave-function-collapse/.

[25] Robert D Melara and Daniel Algom. Driven by information: a tectonic theory of stroop effects. *Psychological review*, 110(3):422, 2003.

[26] Timothy A. Salthouse. The Role of Memory in the Age Decline in Digit-Symbol Substitution Performance1. *Journal of Gerontology*, 33(2):232–238, 03 1978. ISSN 0022-1422. doi: 10.1093/geronj/33.2.232. URL https://doi.org/10.1093/geronj/33.2.232.

[27] Remco R Bouckaert. Bayesian network classifiers in weka, 2004.

[28] Nor Rahayu Ngatirin, Zurinahni Zainol, and Tan Lee Chee Yoong. A comparative study of different classifiers for automatic personality prediction. In *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pages 435–440, 2016. doi: 10.1109/ICCSCE.2016.7893613.

[29] Ghada Hammad AL-Rawashdeh and Rabiei Bin Mamat. Comparison of four email classification algorithms using weka. *International Journal of Computer Science and Information Security (IJCSIS)*, 17(2):42–54, 2019.

[30] Weka dataset, classifier and j48 algorithm for decision tree, 2022. URL https://www.softwaretestinghelp.com/weka-datasets/#J48_Classifier.

[31] S Vijayarani and M Muthulakshmi. Comparative analysis of bayes and lazy classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(8):3118–3124, 2013.

[32] K means clustering using weka, 2021. URL https://www.geeksforgeeks.org/k-means-clustering-using-weka/.