University of Plymouth

School of Engineering,

Computing, and Mathematics

COMP3000

Computing Project

2022/2023

Helping   
A cosy serious game that teaches healthy wellbeing habits

Josie Wood

10509521

BSc (Hons) Computing & Game Development

## 

## 

## 

## Acknowledgements

## Abstract

## Table of Contents

## 

[Acknowledgements 1](#_Toc125362752)

[Abstract 2](#_Toc125362753)

[Table of Contents 3](#_Toc125362754)

[Word Count 4](#_Toc125362755)

[Code link 4](#_Toc125362756)

[1 - Introduction 5](#_Toc125362757)

[2 - Background, objectives & deliverables 5](#_Toc125362758)

[2.1 Project Background 5](#_Toc125362759)

[2.2 Objectives 5](#_Toc125362760)

[2.3 Deliverables 6](#_Toc125362761)

[3 - Literature review (if applicable. Most usually found in a research project) 7](#_Toc125362762)

[4 - Method of approach (or Methodology if a research project) 7](#_Toc125362763)

[5 - Legal, social, ethical and professional issues 7](#_Toc125362764)

[6 - Project management 7](#_Toc125362765)

[7 - End-project report 7](#_Toc125362766)

[8 - Project reflections 7](#_Toc125362767)

[9 - Conclusions 7](#_Toc125362768)

[Reference list and bibliography 7](#_Toc125362769)

## 

## Word Count

## 

## 

## Code link

## 1 - Introduction

Managing and looking after mental health is as important as maintaining good physical health, but people aren’t always taught the best practices and techniques to do so. In some cases, this can lead to people only receiving support once they reach a ‘breaking point’, rather than learning preventative and healthy habits early on.

In recent years, more people than ever are playing video games and feeling the mental health benefits - community, relaxation, and rest time. As the industry has grown, so has the number of games centred around mental health, and the wholesome games movement.

This project aims to combine these two elements. Research has been conducted into how media, particularly games, affect their audiences, and how the fields of medicine and psychology approach mental wellbeing and improving mental health. This informed the creation of a game which aims to support the mental health of players, offering both short term benefits of relaxation during play and long-term benefits of learning and adopting healthy habits outside of play.

## 2 - Background, objectives & deliverables

### 2.1 Project Background

Games for wellbeing and mental health fall into the subcategory of serious games (or applied games). There’s no single definition of a serious game, but the most used is ‘Games which do not have enjoyment, entertainment, or fun as their primary purpose’ (Chen & Michael, 2005).

Serious games have existed since the early days of video games, and multiple titles have reached high levels of popularity and financial success. One of the earliest examples is the Microsoft Flight Simulator franchise, the first edition of which was released in 1982. It is one of the few flight games which focuses on civil aircraft and doesn’t feature combat. The 40th anniversary addition released in 2022 reported over 10 million players and is still receiving updates (Neumann, 2022). In addition to commercial success, studies have suggested that the game has ‘capability to improve novice student performance in an aircraft’ (Callender, et al., 2009), supporting the educational benefits of serious games.

The design of serious games specifically to improve mental health has also proven successful. A meta-analysis of ten studies into serious games for mental health suggested that they can be effective for reducing disorder-related symptoms in patients (Lau, et al., 2017).

### 2.2 Objectives

The primary objective of this project is to create a serious game that provides entertainment and escapism whilst simultaneously teaching the players good mental wellbeing practices.

To better understand this objective and stay on task throughout development, it has been broken down into smaller, measurable objectives as follows.

* Create a research document on mental health practises and therapeutic techniques informed by psychology. Write up at least 3 techniques that could be applied to the project by February so they can be referenced in the writing portion of development.
* Write at least one character dialogue with a full storyline that they player can interact with. Complete this by 13th February so it can be sent out as a standalone text game to at least 5 testers for early feedback.
* Write at least 5 minutes’ worth of dialogue for the player to read by the completion of the project. Measure this by entering the wordcount into wordstotime.com.
* Make a slice of gameplay that can run from start to end without any major errors or bugs (eg, software doesn’t crash, players can move around and access the storyline as expected).

### 2.3 Deliverables

The deliverables for this project include both the working build of the game and the research that informs it.

The working game will be delivered as an executable which can be played on a PC with mouse and keyboard/keyboard only. The game will be a vertical slice of the game, with limited characters and storyline to interact with, with the intention that it can be built upon and expanded at a later point. All art assets featured will be original, and copies of the original PSD files and PNG files will be saved in a folder in the submission folder.

Alongside the game itself, two small scale research documents will be delivered. These will inform the game, and act as an educational resource available for others. One will detail the steps and methods employed in the development of a serious game, whilst the other will provide a quick reference guide to different mental practises and habits encouraged in therapy.

## 3 - Literature review

### 3.1 Psychological approaches to managing mental illness.

### 3.2 Serious games

Serious Games Conceptual Framework

## 4 - Method of approach

### 4.1 Tools

#### Asset Creation

All the visual elements of this project have been custom made using the Game Design Document as reference.

The 2D sprites, including characters, environments, and user interface, are made in Adobe Photoshop. They can all be found as both PNGs and original PSDs in the assets folder.

Calligraphr is used in conjunction with Adobe Photoshop to create a custom font for the game.

Audio elements featured in the project are a mixture of original and third-party assets. Original audio assets are recorded and edited using Audacity and can be found in the assets folder. Third-party assets are referenced in the credits and the Third-party-assets.txt file in the assets folder.

#### Development

The game is developed in the Unity engine. The development environment is Visual Studio for Unity is used, and all executable code is written in C#.

The branching narrative is written using YarnSpinner, an opensource dialogue tool that has Unity integration.

#### Project Management

Google sheets is used for planning and tracking development, for example, the creation of the Gantt chart.

Trello is the primary sprint planning tool. A Kanban board is regularly updated to keep track of tasks.

Git and the GitHub Desktop application are used for version control.

### 4.2 Technical implementation

* Dialogue system
* New input system
* Algorithm to use and write about

#### Dialogue and branching narrative

The branching narrative system for the project is built around the Yarn Spinner writing tool and the accompanying plugin for Unity. All dialogue is written in .yarn files, a plain text format that utilizes special syntax to create files which can be read by the Yarn Spinner framework.

The Yarn Spinner Visual Studio Code extension makes it easier to write in the correct syntax, and provides other helpful features, such as the ability to preview dialogue within VSCode. This was used in the development process to check flow and rhythm of dialogue quickly throughout writing. The extension also makes it easy to export dialogue in a runnable html format, which was used in the dialogue user testing (see section blah).

A key advantage of using Yarn with Unity is the capability to create C# and yarn scripts which can communicate with one another and trigger events.

An example of this is the functionality have future dialogue informed by previous conversations with characters, which is controlled by the Interactable.cs and the YarnManager.cs scripts.

Text

Description automatically generated

A new command handler is added to the dialogueRunner by the YarnManager with the yarn command advanceEntryNode, which references the C# function AdvanceEntryNode(). The AdvanceEntryNode() function takes in the name of the new node as a string, and passes it through to the current interactable so it knows what node should be opened when a player starts dialogue with them.

With the new yarn command advanceEntryNode, the node can be advanced from a specific line in the yarn script.

Text

Description automatically generated

This means that as players interact with characters in the world and return to them, the characters dialogue and writing can change to progress the story forward rather than starting from the beginning every time.

#### Fog of war using Marching Squares

The Marching Squares algorithm is applied to create a system whereby the player explores an area to reveal corresponding areas of a mini map. Based on which vertices on a grid are on or off, a case is identified, and the corresponding square image is displayed in that tile. Using Marching Squares to achieve this fog of war effect on the map is an efficient way of checking the players world location and updating the map visuals accordingly without manually setting each individual pixel.

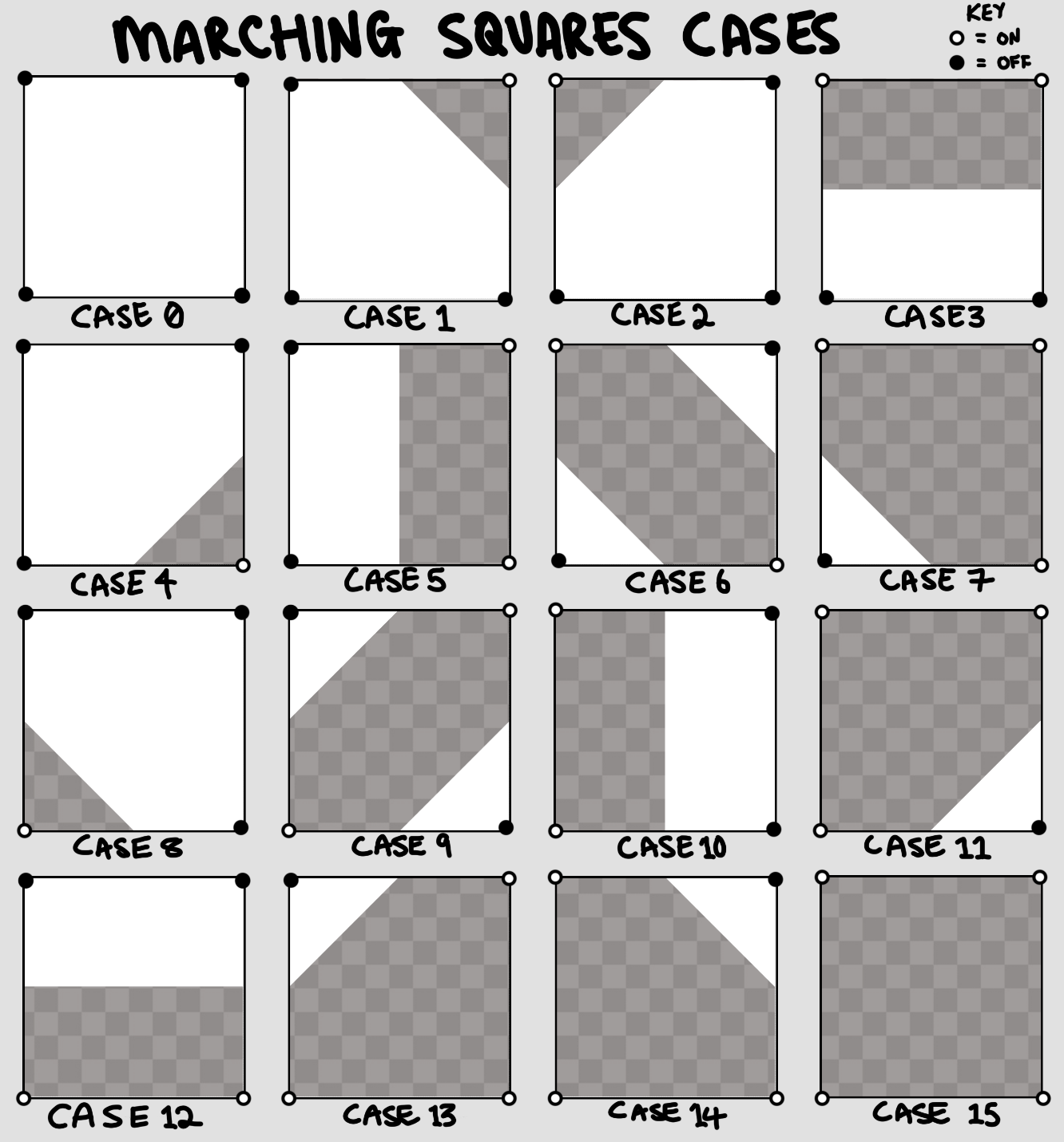
s

Figure 1 - The Marching Squares cases

All of the operations involved in the implementation of Marching Squares is controlled from the MarchingManager.cs class, which is attached to a GameObject in the relevant scene. The custom classes Square.cs and Vertex.cs represent the vertices and squares that make up the grid used for Marching Square calculations.

The Square.cs class contains a list of the four vertices associated with the Square, a conversion list used to convert vertices info into Marching Squares references, a reference to the Marching Manager in the scene, and a Vector3Int containing the Squares position.

It has two functions. OnVertexUpdated() is called by the vertex script and checks the state of each vertex attached to the square, converts the values into a reference number between 0 and 15 which refers to a Marching Squares image.

The updateTileSprite() function takes in the new Marching Squares reference number and updates the current tile accordingly.

The Vertex.cs class contains a list of the squares it’s associated with, a bool for current state named isOn, and a Vector3Int position.

It’s only function is IsOn() getter/setter. The get function returns the value of isOn. The set function is called by the MarchingManager.cs script when player movement triggers a new vertex to be turned on, and calls OnVertexUpdated on each of it’s associated Squares.

On Start, MarchingManager.cs uses a provided tilemap to create a 2d array of squares based on the bounds of the drawn map. This allows the developer to draw any tilemap before runtime and have it automatically acknowledged and converted to the array which can be accessed by other scripts.

Then the script creates corresponding 2d array of the vertices from the squares array, applying offsets to each of the square positions to find the four corners of each square.

On Update, MarchingManager.cs gets the player position in world space, uses the helper function translatePlayerPosToTilemap() to translate it to a Vector3Int position on the tilemap, and calls turnVertexOn() on the vertex at that position on the tilemap.

In the product, this is used to create a fog of war effect over a mini map in the corner of the screen when exploring the forest. The Marching Squares sprites are varying proportions of opaque white and transparency. As the player explores, more of the tilemap becomes transparent, revealing the hand drawn map asset on the layer behind.

### 4.3 Project Management

#### Start of the project

At the start of the project, the Gantt chart was created in Google Sheets, informed by the project vision and risk analysis outlined in the Project Initiation Document.

To account for the biggest risk to the project, delays due to chronic illness, three buffer periods were planned into the Gantt chart to make up for any missed time. In winter of 2022 a prolonged period of illness halted development. Because of the planned buffer periods, a few months of illness only put the project 3 weeks behind schedule. A revised Gantt chart was created in January to account for this delay and add more buffer periods in the following term.

The Trello board was also started at the start of the project. It is a Kanban board with columns for *Backlog*, *To-do this sprint*, *Doing*, *In Review*, *Completed this sprint*, and *Finished* tasks. Cards were created and placed in the Backlog column using the Gantt chart tasks, and were colour coded and labelled according to sprint.

An empty Sprint Record document was started with sections to document the sprints as the project progressed.

#### Sprint Process throughout the project

**Sprint planning** – In Trello, move items from *Backlog* to *To-do this sprint*. Write up the plan in the Sprint Record.

**During sprint** - Work on current items and move them in Trello to *Review* and then *Completed this sprint*.

**Sprint Review** – Take a step back to review work completed. Move Trello cards from *Completed this sprint* to *Finished.* Write up completed work in the Sprint Record, answering the questions:

* What did I do this sprint?
* What went well?
* What didn’t go well?
* Were there any blockers?
* What actions can I take to avoid/work past these blockers in future?

**Backlog refinement** - Look at the state of the project, remaining time, and project goals and use this to add, remove, or edit items in the Trello backlog ready for the next sprint. Ensure all branches of version control are up to date with one another before starting a new sprint.

Every two weeks, a meeting is held with the supervisor to discuss progress and get advice on the project.

### 4.4 Data collection procedures

* Identify how many data collection sessions, how they run, how they are used.

## 5 - Legal, social, ethical and professional issues

## 7 - End-project report

## 8 - Project reflections

## 9 - Conclusions

## Reference list and bibliography