|  |
| --- |
|  |
| Amaz’d  Michael Tan-Sikorski |
| |  |  |  | | --- | --- | --- | | Michael Tan-Sikorski (13FG) | Amaz’d | Computer Science Coursework | |

Centre number: 12610

Candidate Number: 4141

Project Log:

|  |  |
| --- | --- |
| Date | Task |
| 15/03/2023 | Initial project ideas: I came up with many different ideas I could chose for my project, that I had to make sure were complex enough. The ideas were: -**Poker game against ai**, **quiz app based on user interests**, **solar system simulation**, **battleships against ai**, **3D-board chess against opponent**, **restaurant finder in a certain radius of a user**, **backtracking algorithm maze puzzle game**. |
| 19/03/2023 | Project idea of backtracking algorithm maze puzzle game chosen to be researched in more depth as a strong contender to be my project. |
| 19/03/2023 – 02/04/2023 | Research into ways to code a maze puzzle game. -Looking at potential maze algorithms, coding language and method to make it (decided on pygame), looking at tutorials to potentially use to learn pygame, high- and low-level objectives for my game |
| 03/04/2023 – 18/04/2023 | Learning pygame by following clear-code and tech with tim tutorials |
| 01/05/2023-27/06/2023 | Much deeper analysis on project, creating a PowerPoint of all my information for: Stakeholders, objectives, identified third parties to give me feedback throughout the project, research into depth-first maze algorithm, creating a prototype UI for what it may look like in the game, identifying similar games in style to my game in terms of visuals, hardware + software requirements |
| 27/06/2023-  07/07/2023 | Doing some final pygame tutorials from clear code, to learn how to animate a moving character, using sprite classes in pygame in order to use object-oriented programming, and create a camera to move around a character |
| Summer Break  07/07/2023  -04/09/2023 | Starting to code the project, first started by coding a procedurally generated maze, using the “depth-first algorithm” for maze generation. |
| 04/09/2023-14/09/2023 | Added music and started to add the “powerup” system to the maze, which randomly spawn around the maze. Haven’t coded the collision detection yet, or the buff it will give to the player |
| 14/09/2023-25/09/2023 | Powerups now successfully collide with the player and makes a “ping” collect sound when the player collides. After collision, the powerup gets destroyed, and the user gets a speed boost. Powerups now have a “lightning in a bottle” icon, that should suggest to the player that it is a speed powerup. |
| 25/09/2023-01/10/2023 | Powerups icon is now animated and flips between 2 images. Updated powerup collect sound to be more indicative of a speed boost, due to stakeholder feedback saying the “coin collection sound” was misleading. |
| 01/10/2023-12/10/2023 | Stamina is in the game now as there was no reason not to sprint forever, now makes the faster sprinting a trade-off for stamina. Stamina is used up when the user is sprinting, and regenerates over time when the user isn’t sprinting. Once stamina hits zero, the user will no longer be able to sprint. Currently is on the screen as a 2 significant figure number on the screen that goes down from 1500-0 when the user sprints |
| 12/10/2023-07/11/2023 | Stamina bar graphics are now animated when the user uses stamina, and an indicator for how long a powerup will last for. Enemy code is started but unfinished, currently the enemies spawn in the maze but have no functionality. The user can also press “space” to attack, but the graphics are rough, and the animations haven’t been coded in yet. |
| 07/11/2023-09/11/2023 | Point system has been added. Will start at 500 points and decay every second. The points the user has at the end is their score. Points just displayed at the top of the screen. |
| 09/11/2023-16/11/2023 | Coins added to the maze. They randomly generate on empty tiles around the maze. When picked up, they make a “ding” pickup sound, and grant the player +20 points.  -Also tweaked speed powerup, so that the players running animation speeds up when they pick up a speed powerup |
| 16/11/2023  21/11/2023 | “Game Over” text message when the user runs out of points, and closes the game |
| 21/11/2023-25/11/2023 | Exit portal now generates on the final row of the maze and closes the game on collision (temporary).  -Also tweaked maze.py so that it is easier to change the size of the maze I want to generate (only have to change values in main.py now) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Project background and problem investigated:

When it came to creating a project, I **knew** I wanted to create something not only fun, but also testing problem solving skills.

-Due to the ever-growing number of mindless games, riddled with advertisements, made for the sake of a profit only. I wanted to make sure my game would have puzzles, and test the users, so my game wouldn’t be mindless, and would engage the user for their entire playtime.

-I have always loved puzzle games that require you to come up with clever and fulfilling solutions, games such as portal 2 and breath of the wild come to mind, which shaped my childhood.

-This gave me a spark, in which I decided to create a game. I knew I wanted it to have infinite replay ability, so something that could be procedurally generated, to have infinite levels was desirable for me.

-I figured making a game, centered around solving a maze would be perfect, as it fit my criteria of being a puzzle, involving player problem solving, and being able to be procedurally generated, so that each time playing would be different.

High and low-level Objectives

|  |  |  |  |
| --- | --- | --- | --- |
| Spec Point | High-level objectives | Objectives in detail | Achieved? |
| 1.1 | **Have a moving, animated 2D player** | A 2D player sprite that can move, up, down, left and right. The player must also have a sprint option in order to speed up and have animations for when they are moving and sprinting. |  |
| 1.2 | **Player has a stamina bar that stops the player from sprinting when they run out of stamina** | A stamina bar on the top of the screen, that will decrease when the player sprints, and will regenerate when the player is not sprinting. When the stamina bar is empty/ reaches 0, then the user will not be able to sprint, as they will have run out of stamina |  |
| 1.3 | **Procedurally generated maze that the character can move through** | A maze, created from a procedural maze algorithm (using the depth first algorithm for this project), that must have graphics for the walls and floor. The player should be able to move through the maze. |  |
| 1.4 | **Maze walls that the character and enemies can’t move through** | The maze walls must collide with any entities such as players or enemies that try to move through it, so that they cannot move through the walls. |  |
| 1.5 | **Music and sounds for the game** | There must be music for the menu, and the game. There should also be sound effects for when the player is moving, attacking and entering the maze. |  |
| 1.6 | **Main Menu screen with UI for user to interact with** | There should be a main menu that the user is greeted by when they first enter the game. It should show the player, the game title, text saying “main menu”, and options to “quit”, “play” and “controls”. The user should be able to click on these options. |  |
| 1.7 | **Multiple levels for user to play** | There should be multiple levels/mazes that the user can go through and play. |  |
| 1.8 | **Weapons for characters and attacks** | The player should have weapons, and a small inventory so they can switch between items. They should be able to attack with these weapons through inputs such as mouse clicks. Weapons should include a sword. |  |
| 1.9 | **Computer enemies with logic to target the player** | There should be enemies, that can move, and are fully animated. They should try to attack the player but can be killed by the player. They will spawn randomly in the maze, but they can’t spawn too close to the player, and there should be a limit of how many can spawn in one level. |  |
| 1.10 | **A point system for the player** | The player should be able to get points, based on how quickly they finish the maze, and how many enemies they defeat. |  |
| 1.11 | **Coins for the player to pick up** | Coins that randomly spawn in the maze, that should be able to be picked up by the player and give them points. |  |
| 1.12 | **Exit in the maze so the player can escape** | A maze that generates far away from the player, that they must reach in order to get to the next level. |  |
| Spec Point | Low-level objectives | Objectives in detail | Achieved? |
| 2.1 | **Have collectable powerups for the user randomly spawning in the mazes** | Powerups that give the user bonuses such as increased speed, increased weapon damage against enemies. These powerups should spawn randomly in the maze but be relatively rare. |  |
| 2.2 | **Minigames within the maze** | There could be small minigames that the player can play in the maze, which could give them a random powerup as a reward. |  |
| 2.3 | **Add easter eggs to the game** | Add little details and secrets in the game, that sharp-eyed players can discover while they play the game. |  |

Target stakeholders and identified third parties to obtain feedback from for this project.

-Players who love computer games

-Those who love problem solving and challenging their mind

-casual gamers who want to kill some time while also solving problems

**Third parties**:

-Theo Bahns (3D animation professional and computer game enthusiast)

-Kyrian Salas (Top student studying computer science)

-Daniel Tunyan (cyber centurion competition top 20 nationally)

Hardware and software requirements for the project

**Hardware**

-Computer and monitor (Computer needs a GPU in order to have graphical processing)

-Input/output devices (Keyboard, mouse, controller, speakers/headphones)

-Memory (at least 512 MB RAM but preferably at least 2GB of RAM)

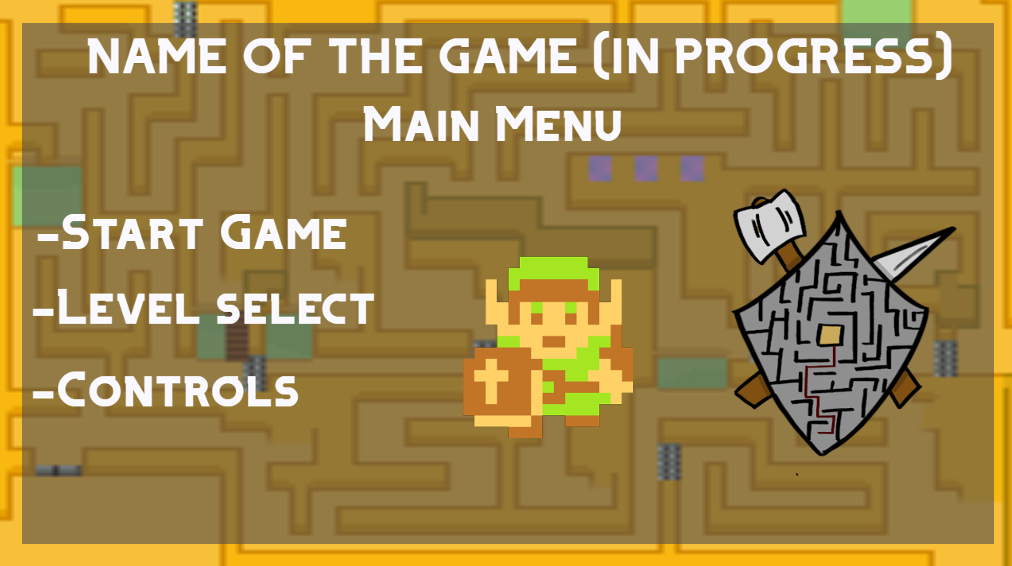
-Processor: Intel® Pentium® 4 2.0 GHz equivalent or faster processor

**Software**

-Python 3.0.0 or higher installed on the computer

-pygame library installed on python

Prototype UI for the main menu concept



**Title:**

-There will be the name of the game at the top, with the icon of the character, and the logo of the game.

**-There will be 3 options to choose from in the main menu:**

-Start Game

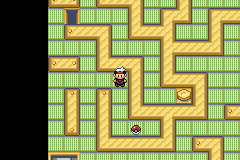
-Level Select

-Controls

**Stakeholder feedback:**

From stakeholder feedback, I could also add a “Quit Game” and “key binds/ settings” to the menu or rename “controls” to “key binds/ settings” and add extra functionality if the user selects that option.

Inspiration from other games (what my game will look like, how gameplay may function)

(screenshots taken from Pokemon Ruby and undertale)

<https://docs.google.com/file/d/0B58fq8yIsMahNEtPVkNXVUdHd00/edit?resourcekey=0-CLMprHJsUDqcqVtnfeAkig>

