Contents

[Analysis 2](#_Toc1202905957)

[Problem identification 2](#_Toc489855512)

[Stakeholders 3](#_Toc1439028085)

[Why it is suited for a computational solution 3](#_Toc1540261053)

[Research 3](#_Toc634962781)

[Existing solutions 3](#_Toc562315962)

[Survey 6](#_Toc34212995)

[Key features 8](#_Toc621878721)

[Success criteria 10](#_Toc746687901)

[Limitations 11](#_Toc1447279773)

[Hardware requirements 11](#_Toc1885306524)

[Software requirements 11](#_Toc414305454)

[Design 12](#_Toc1398777832)

[Hierarchy diagrams 12](#_Toc1760733417)

[Variables 13](#_Toc1964573202)

[Class diagrams 13](#_Toc540439601)

[Algorithms 14](#_Toc1915328307)

[Data structures 15](#_Toc493495301)

[GUIs 16](#_Toc656288257)

[Usability 18](#_Toc1466915649)

[Testing 18](#_Toc740199328)

[Bibliography 18](#_Toc312436156)

[Notes (temp) 19](#_Toc1853615947)

# **Analysis**

# Problem identification

Many students use videogames as a way to relax and take their mind off school, but many find that the games that have come out in recent years are not very challenging in order to allow for wider audiences to play them. Dungeon crawlers are a popular choice for many gamers as there are many to choose from, however, many lack a level of strategy as the combat systems make the game quite repetitive causing them to be quite easy. Also, many dungeon crawlers rely heavily on story driven gameplay meaning that some parts later in the game become quite complex as they require knowledge from earlier in the story and this also causes them to have little replayability.

The features that often cause dungeon crawler games to be limited seem to be combat systems and complex storylines so therefore this problem may require no or an altered combat system, little to no reliance on a story and more strategy focussed gameplay. Also, currently available dungeon crawlers very rarely make use of deckbuilding gameplay features which could allow for greater replayability as it would affect aspects of gameplay through the use of random card draws that provide power-ups and it can add more strategy as the player would have to build up their deck over time and consider the cards they use to reflect their own play style.

# Stakeholders

The demographic for this product would be teenagers aged 13-18 who have an interest in videogames as they are often looking for games that will not be too distracting or time consuming but are also challenging and require strategy. This game will be able to be played on the students’ PCs and would have gameplay features that encourage short playtimes so that they do not have to worry about not having enough time to finish rounds or dedicate to a storyline which allows them to just open and play the game whenever.

# Why it is suited for a computational solution

The problem works well with a computational solution as it allows for the interactivity needed to make the game enjoyable as many features of online games are difficult or impossible to replicate outside of a computer such as monster AI as it requires the ability to move around and track the player. It is also suited for a computational solution as it requires large amounts of data to be processed at once, such as player location and statistics relating to all features, which would not be possible to track without computational assistance.

# Research

# Existing solutions

**Decked out 2**

**Overview**

Decked out 2 is a deck-building dungeon crawler game built within Minecraft where players must enter the dungeon to search for an artifact which they must bring back at the end of their run, in return for frost embers to be able to upgrade their card deck, there are also 5 levels of difficulty which affect how far the player must travel through the dungeon to retrieve their artifact. Decked out has many systems to increase the difficulty of the run the longer the player stays in the dungeon including hazard which shuts off pathways through the dungeon to make it more difficult for the player to exit, but not impossible, clank which builds up slowly throughout the game depending on the player’s actions and once it reaches maximum, many new monsters and obstacles spawn to make leaving the dungeon as difficult as possible and finally, a GUI map which is used to allow players to keep track of their statistics which update throughout the game as cards are drawn from their deck, however, one system the game lacks is a combat system.

**Parts I could apply**

The lack of a combat system in this game seems to work well as it increases the strategy players need to apply throughout the game as monsters need to be avoided and maneuvered to progress through the dungeon instead of just killed like in most dungeon crawlers. This strategy focussed gameplay is something I want to incorporate into my solution so this could be a good way to include it. Also, the hazard system shutting off paths as more time is spent in the dungeon is useful in discouraging players from staying in the dungeon for too long and something I want to think about in my solution. Finally, the clank system adds another level of strategy as players need to be careful when moving through the dungeon which prevents them from running through it fast which is something I want to think about in my solution, but I may not have enough time to add anything similar.

**The Binding of Isaac**



**Overview**

A top-down dungeon crawler in which you can use unlockable characters with different attributes to run through dungeons that are random, procedurally generated levels. Players must fight all the monsters in a room before they can proceed to the next which is done through range combat. The player can also get powerups that are either items or effects which can stack. Once a room has been cleared, it remains cleared and can be returned to throughout the level and along the way players can pick up coins which are used to buy more powerups, keys to special rooms and weapons. Once the player loses all their hearts, it is a permadeath and the level must be started again.

**Parts I could apply**

The powerups which can give effects or items could work well in my game as it allows for more variety in gameplay than if the powerups only gave effects and the ability for effects to stack would also work well. I could also use the ability to gather coins throughout the dungeon which can be exchanged later for more powerups or items as it allows for more strategy, and it adds more of an incentive to explore the levels more to gain more money. The permadeath system is also quite useful for my game as it makes it more difficult and stops players from being able to force their way through levels by relying on respawning as it resets their progress on that run. I could also make use of the different enemies as it provides variation throughout the game and allows for easy difficulty increases as enemies could change rather than have their stats upgrade and this also allows for the player to know how difficult each enemy they come across will be.

**Dicey Dungeons**



**Overview**

Dicey Dungeons combines both roguelike and deck-building gameplay in a gameshow-like game where adventurers have been turned into dice and players choose between one of six characters which defines their starting equipment. The player moves their character across a dungeon map and encounter monsters, treasure, shops, upgrade stations and exits. The combat system is turn based and players can gain buffs throughout the combat which continues until either the enemy’s or character’s health reaches 0. Each battle gives in-game monetary rewards and character experience and as the character levels up, their health and amount of dice increase.

**Parts I could apply**

The different things that the player can find across the map such as shops and treasure I could include in my game as the treasure can be found throughout the dungeon and instead of using the shops in the dungeon, they could be outside, and the treasure collected would be used to get new items so that the player could progress further. I could also use a levelling system like in Dicey Dungeons but instead of increasing health of the player, number of cards increase as this would allow for players to progress further.

# Survey

1. Have you played any dungeon crawler games before?

2. How many different ones?

3. What features did you like?

4. What features did you dislike?

5. What features do you want to see added to/ more of in this genre?

6. Have you played any deck-building games before?

7. How many different ones?

8. What features did you like?

9. What features did you dislike?

10. What features do you want to see added to/ more of in this genre?

11. Put these features in order of how important they are to you in any game

Results:

**Have you played any dungeon crawler games before?**

* Yes [4]
* No [1]

**How many different ones?**

* 1-3 [2]
* 4-6 [1]
* 7-9 [1]

**What features did you like?**

* Multiplayer with my friends
* Combat and the different enemy sprites
* Changing and flickering light aesthetic
* Cool enemies
* Fun interactive battle system

**What features did you dislike?**

* Repetitiveness
* Straight box like shape- Add depth to the system
* Repetitiveness in style for different levels
* Enemy ai being too simplistic
* Sometimes feels like a clicking simulator
* Repetitive puzzles

**What features do you want to see added to/ more of in this genre?**

* Interactive items
* More unexpectedness
* More advanced enemies with their own special skills rather than straight line walking and body contact dmg
* Traps and other non-enemy threats
* Time limits/timers
* Dungeon ecologies (e.g. spider infestation), and certain actions affect the area around you (you kill the spiders, then later it will be infested by an even bigger threat)

**Have you played any deck-building games before?**

* Yes [3]
* No [2]

**How many different ones?**

* 1-3 [2]
* 4-6 [1]

**What features did you like?**

* Variety of decks possible
* The freedom to create my own deck
* Cool card designs
* Intricate abilities that made it so that everything was balanced

**What features did you dislike?**

* Non-balanced cards
* Sometimes fights can be unfair
* Overly complicated rules/too many rules
* Too many extra mechanics that get confusing

**What features do you want to see added to/ more of in this genre?**

* Handicaps for overly powerful decks
* Mix of strategy and RNG
* Less ability to control the draw of cards to make it more challenging

**Put these features in order of how important they are to you in any game**

1. Art
2. Strategy
3. Mix of enemies
4. Music
5. Easy to read interface
6. Challenging gameplay
7. Difficulty
8. Mix of mechanics
9. Fast paced
10. Storyline

Analysis of survey

Liked: differing enemies, deck varieties, balance, art, strategy, interface, difficulty

Disliked: repetitiveness, enemies being too simplistic, lack of interactivity, lack of balance, overly complicated

Add: Interactivity, RNG, more advanced enemies, more unique enemies, non-enemy threats, time limits, balance, more strategy

# Key features

|  |  |  |  |
| --- | --- | --- | --- |
|  | Requirement | Solves | Justification |
| 1 | Menu screen | Users having one screen to navigate to the main game, options and to quit | Provides an easy to use and understand way of allowing the user to access different parts of the game |
| 2 | Play button | Users being able to load the main game | Allows for the main game to be quickly and easily loaded |
| 3 | Options button | Users being able to access and change settings | Users need to be able to change settings to adjust accessibility features and general settings to improve their experience of the game |
| 4 | Quit button | Users being able to close the game | Allows the user to easily close the program once opened as the window will be full screen which makes it harder to close |
| 5 | Main camp | An immersive menu that lets the player change around items, access the item shop and load the main level of the game | This part being an interactable area allows the user to get used to the controls when they load the game for the first time without having the pressure of trying to survive against monsters. It also helps to improve the immersion in the game as it reduces the amount of plain clickthrough menus |
| 6 | Playable character | The user being able to move around and interact with the game | Allows the user to interact with the game with a visual guide as to where they are and what they are doing |
| 7 | Map | Having an area for the main game to take place | The game needs a map to allow for the main game to happen within as it provides an enclosed area for the monsters and treasure to be placed and for the player to navigate around |
| 8 | Monters with pathing AI | Increases the difficulty of the game  Making the game more difficult for the player | The monsters change the game from just the player navigating around a map to find treasure by adding a threat to the player as the player can then be killed |
| 9 | Power-ups through cards | Increases strategy and variation in the game and adds more for the player to work towards | Power-ups allow for the game to have more variation as many aspects can be constantly changed rather than the game being highly repetitive and adds an aspect of random luck into the game as the draws are unpredictable |
| 10 | Visible player stats | Allows player to keep track of their health | Users need to be able to always see the character’s health so that they can adjust how they continue playing accordingly as the character can be killed |
| 11 | Visual tracker of active power-ups | Allows user to track what power-ups are currently active and their durations | Users need to track the power-ups to know how their gameplay will be affected and to plan how to use each one to their advantage which increases strategy |
| 12 | Treasure system | Adds a main goal for the player to work towards and a way to progress through the game | The main goal is to get the assigned treasure out of the dungeon which can be exchanged for cards. The player also has the option to leave without collecting the treasure so allowing them to collect coins throughout which can be exchanged for items adds a reason to leave without risking collecting the main treasure |
| 13 | Item shop | Allows for more variation between each game | Provides a reason for players to explore to collect coins as they can exchange them for extra items to help them through the game |

# Success criteria

|  |  |  |
| --- | --- | --- |
|  | **Success criteria** | **Justification** |
| **1** | - Windowed  - Game title  - 3 equally sized buttons |  |
| **2** | - Can be clicked to load the main game  - Game loads within 10 seconds |  |
| **3** | - Can be clicked to load the options screen  - Options screen should load within 5 seconds |  |
| **4** | - Can be clicked to exit the application  - Closes application within 3 seconds |  |
| **5** | - Map  - Area to access items  - Area to load game map |  |
| **6** | - Visible sprite  - Moves in different directions with WASD  - Can interact  - Can attack | The user needs to be able to track where they are in the game to be able to navigate and engage in combat. They also need to be able to move around the map and interact with any items and areas they need to. They also need to be able to attack enemies |
| **7** | - An area with set boundaries including walls  - All areas are accessible to the player | Provides an enclosed playing space for the user which can add difficulty to the game as they are not free to roam everywhere and all created areas should be accessible to the player as otherwise it may make certain aspects of the game impossible |
| **8** | - Can navigate around the map  - Can path find to the player within a limited range  - Can attack the player |  |
| **9** | - Cards with different abilities  - Each card in the deck has an equal chance to be activated every 30 seconds  - When activated, each card provides different power-ups |  |
| **10** | - Updating player health bar | The user will need to track their health throughout the game as it impacts the decisions they will make while playing |
| **11** | - Updating tracker of cards in deck  - Tracker for each type of power-up  - Tracker for the duration of each power-up |  |
| **12** | - Coins scattered randomly across the map  - Coins continue generating throughout the duration of the game  - Main treasure locations added throughout the map |  |
| **13** | - Interactable menu screen  - Allows player to exchange coins for cards and other items |  |

# Limitations

A limitation of my solution is that I will not have enough time to be able to create the full game due to deadlines and limitations in my ability to code games meaning that I will have to spend time learning how to implement certain features into the solution. Another limitation will be the features available to me through pygame as I can only use add what is compatible with the programming language I use. Another limitation is that I will need to take time to ensure that the solution can be made as accessible as possible through settings available to the user, however some may not be compatible with my solution.

# Hardware requirements

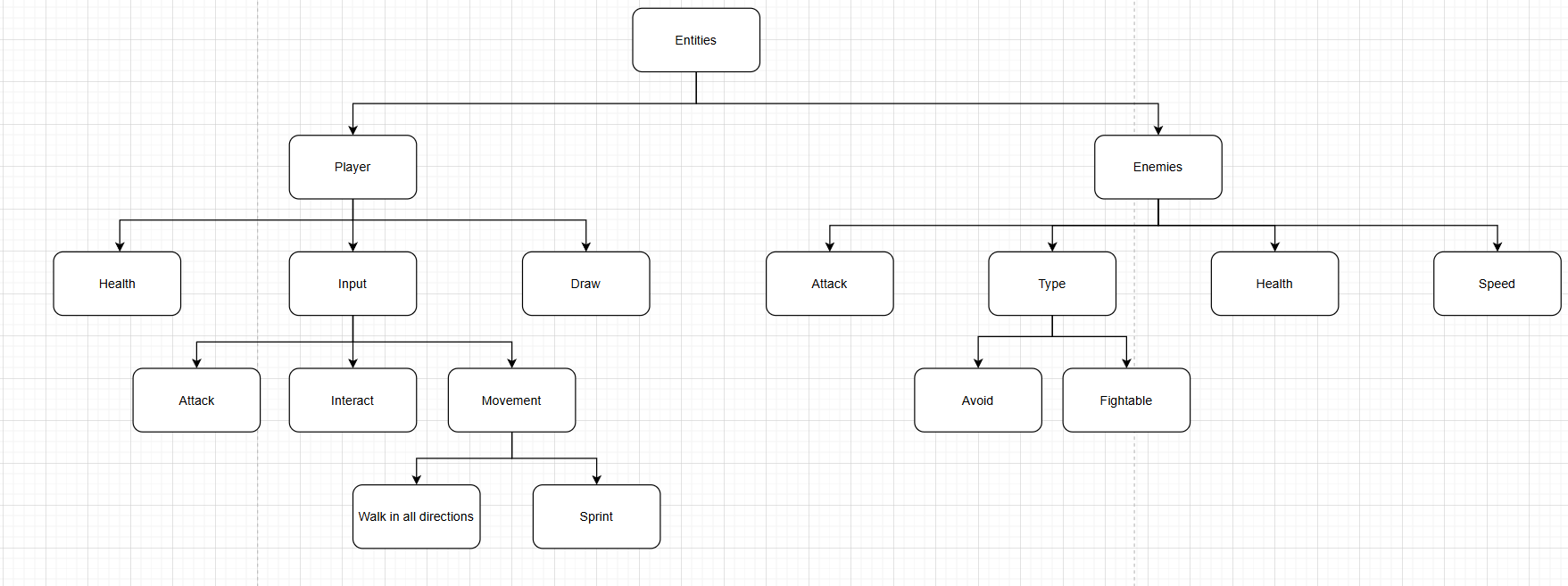
To run the game, users will need a computer or laptop that can run the game which uses standard IO peripherals as the game requires constant inputs from the keyboard and mouse, output on the monitor and the computer needs to have a fast enough processor to manage the constant updates within the game while it is running.

# Software requirements

Users will need a Windows, Mac or Linux operating system to run the game as these are all supported by python. They will also need a python interpreter as the code will all be written in python so it cannot be run without it. They will also need pygame for python as the code will use the pygame library for most of the game.

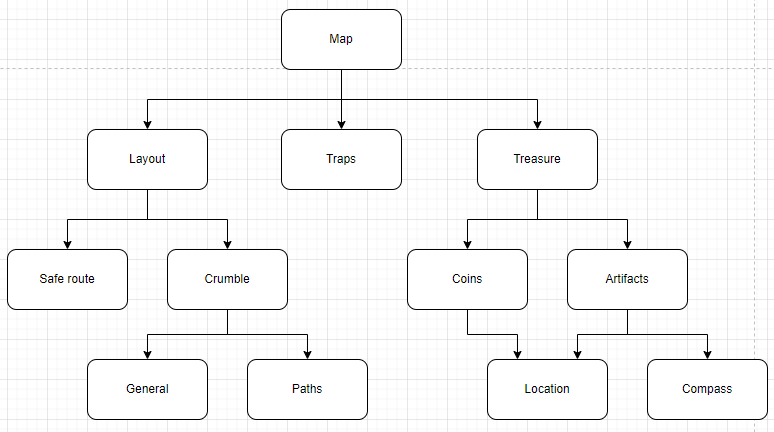
# **Design**

# Hierarchy diagrams



I have broken down the problem starting with entities as they will have similar characteristics so they will be grouped together. Entities have been further broken down into player and enemies as although they have some shared characteristics, they also have a lot of differences as the player relies on input for any movement or actions whereas the enemies will rely on algorithms for those things so they cannot be made together. For the player I have broken down the characteristics into health, input and draw as the player will have an updating health value as they take damage and heal throughout the game, the player will also need to be drawn to the screen each frame otherwise the sprite will not be displayed and finally the player also needs to take input for the user to be able to interact with the game. Input has been broken down further into attack, interact and movement as each require different inputs and require very different processes. For movement, it has also been broken down into speed and directions as the player’s speed will be updated during the game dependant on the cards drawn

Enemies- attack constant per different enemy, type whether player is encouraged to fight or avoid as some will be invincible, health constant depending on enemy type and each different enemy, speed constant dependant on different enemy

Layout- safe route considered first to make sure the game does not become impossible, crumble general for chance of crumble being activated at regular intervals, paths for chance for each individual path to close when crumble is activated

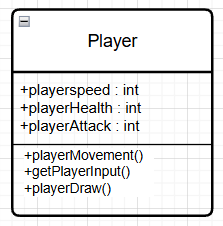
Traps- may not add

Treasure- coins spawn locations and amount dropped, artifact set locations but one picked randomly each run and compass needs to point to the one that can be retrieved

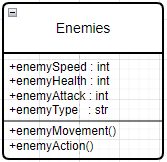
# Variables

|  |  |  |
| --- | --- | --- |
| Name | Data type | Justification |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Class diagrams

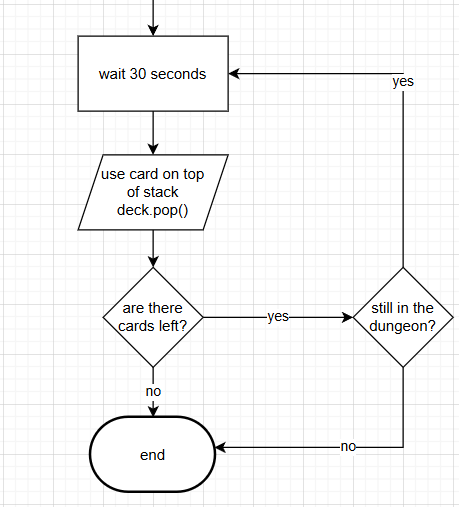
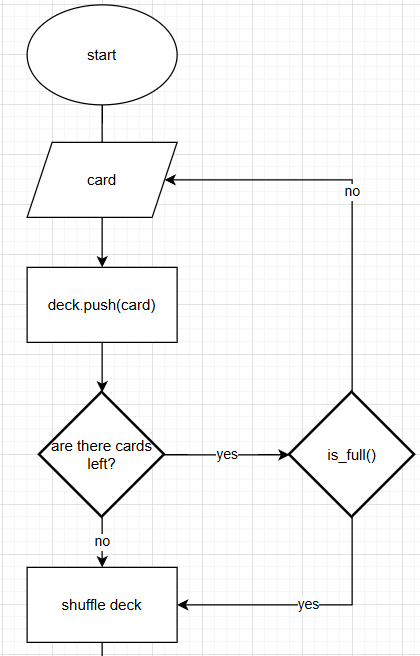


The player class will have the attributes for speed, health and attack, all of which will take integers and will have methods for movement, input and draw (cards changing attributes).



The enemies class will have attributes for speed, health and attack which will be integers and type which will be a string. It will also have methods for movement and action which will control how they move about and when they attack or get hit.

# Algorithms

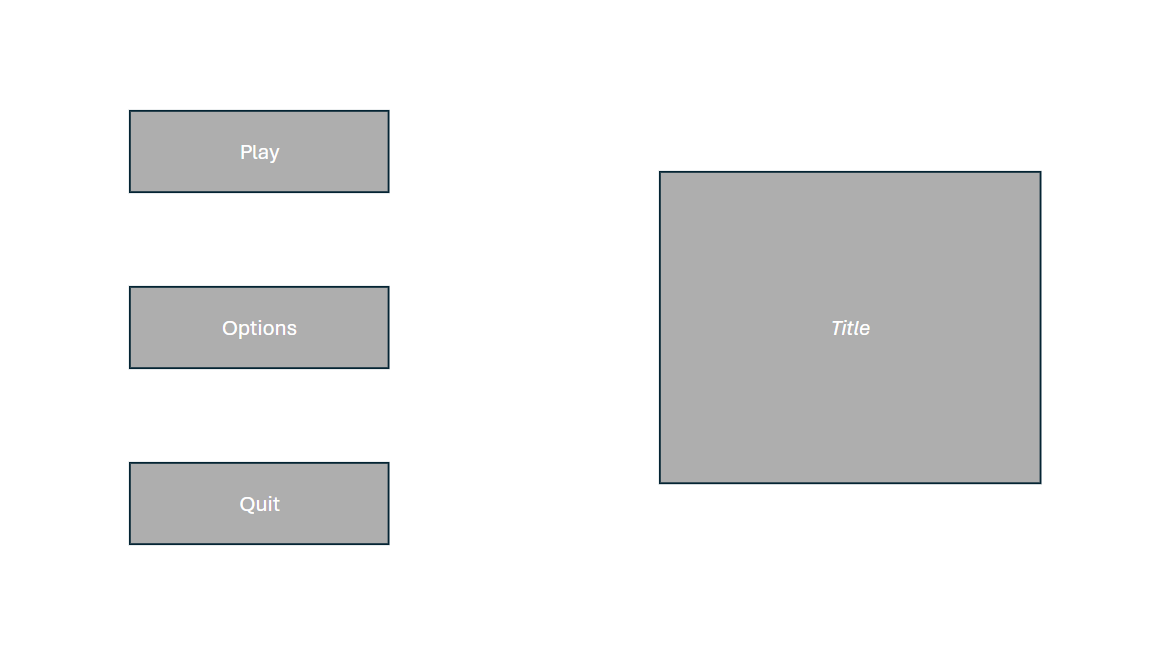


# Data structures

For the deck, I will use a stack as it replicates how cards are used in real life and it allows for me to shuffle the order of the cards that are put in as will be needed to prevent users from being able to determine the order the cards are drawn in the dungeon which will lead to the game requiring more strategy from the user while playing as they will need to the cards drawn to make the best use of them. The stack will also allow me to use pop() and push() which allows for the deck to be easily updated throughout the game when users initially decide the cards to use and while the cards are being used in the dungeon

# GUIs

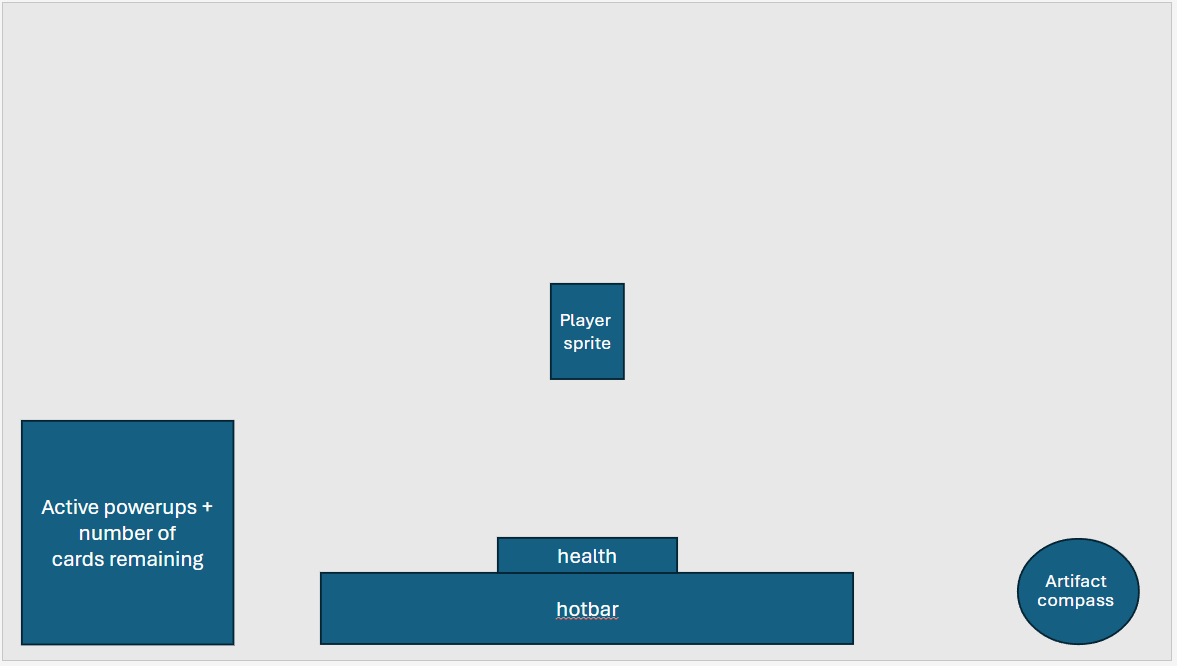
Title screen



As the title screen is the first thing users will see as they load the program, the game logo and title will be placed on the right of the screen, so the user knows what program they have launched. This screen will also have 3 buttons that allow the player to go into the game, enter the options menu to adjust any settings needed before they start the game and quit the program entirely. These buttons will be large and use a contrasting colour to the rest of the background to make them stand out and they will use large text to allow for players to read them easily.

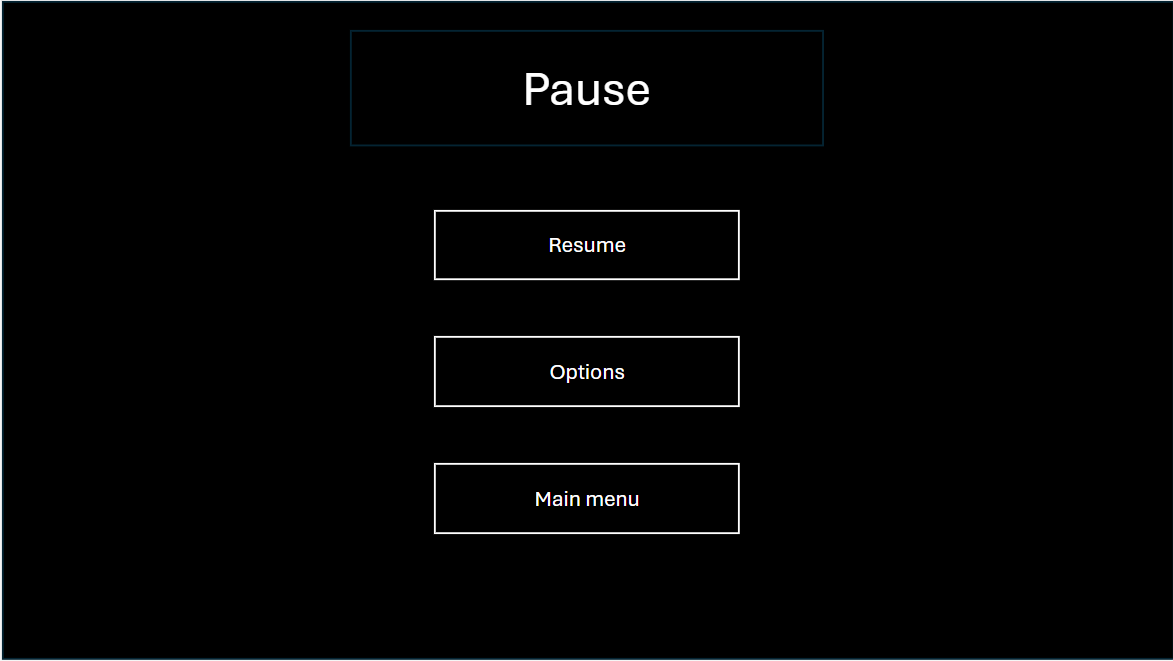
Game screen

Temporary design while rough proportions get worked out due to not having parts drawn out yet



Throughout the game the player sprite will be locked to the centre of the screen which allows the player to easily keep track of the character and makes sure their view in any direction is not blocked or limited due to being too close to the edge. All extra information has been placed at the bottom of the screen so that it is easy for the user to find. The compass has been placed to the bottom right of the screen as it is unobtrusive as users will not need to be focussed on it at all times. The powerups and card tracker has been placed to the bottom left of the screen to keep it out of the way as most players will not check this information too regularly, but it will still be useful. The hotbar and health bar have been placed at the bottom middle of the screen as users will be checking this information the most often as they need to switch between items often and will have to keep an eye on their remaining health throughout due to it being very limited so this being placed in the centre helps to keep players aware of these as they are easy to check.

Pause



For the pause menu, the background of the screen will be blacked out, this is to prevent users from using the pause feature to get an advantage in the game as they will be unable to see the map from this screen. The 3 buttons will be a light gray to contrast with the black background and the text will be black so that the buttons can be easily read. The buttons will allow the user to resume the game, go to the options menu to adjust any settings and return to the title screen to be able to quickly close the program when needed.

# Usability

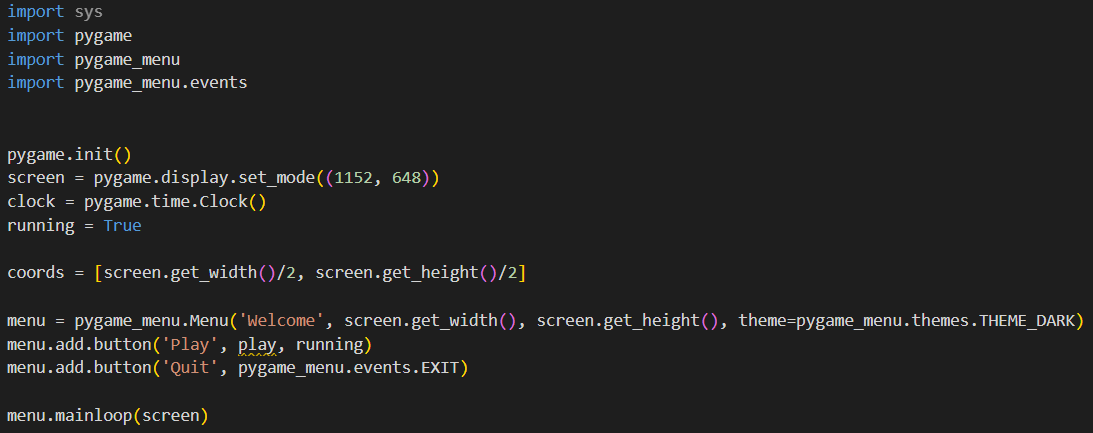
I have made all the buttons and text quite large and given them contrasting colours to allow for people to be able to easily read them and interact with the menus. I will also make sure there are explanations about gameplay features which are easily available so that people will understand how to play and use features correctly. To make the gameplay accessible, I will allow for users to change their keybinds which allows for them to make the inputs more accessible to them as if I left them unable to be changed the user may find it difficult to use the preset controls due to the distance between each keybind, the keybind changes will also allow for users to bind each input to the mouse as this may be preferred, finally the user will be prevented from binding inputs to the same key to prevent them from making the game unplayable for themself. The game will also be able to be paused at any time which allows for users to step away from the program whenever and for however long they need without being punished.

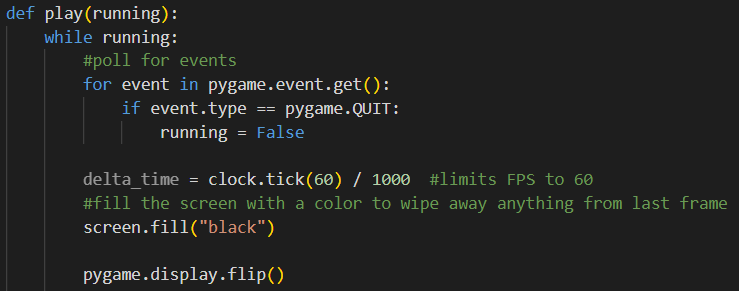
# Testing

# **Development and testing**

# Stage 1: main menu

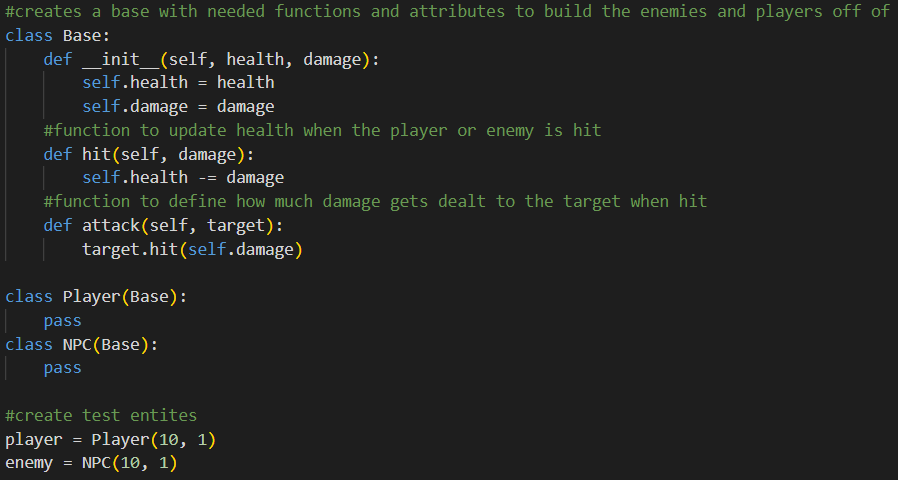
Figure out what actually needs to go in documentation

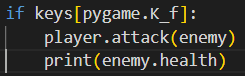
First, I imported the relevant libraries for my code, pygame as it is what the majority of my code will be using and pygame\_menu and pygame\_menu.events so that I am able to create functional menu screens for my game. Then I had to start by creating the main menu and declaring some of the variables I will need for the main game loop and menu screen.

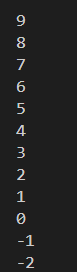
After I created the main game loop in a function so that it could be called by the menu, which currently creates a blank screen. This allowed for me to finish testing the functionality of the main menu

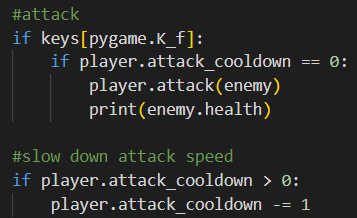
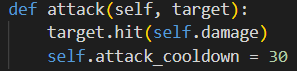
Review stage 1: what was in the stage, testing, what next

When testing the menu screen, I had to remove the play button as I did not yet have the function defined causing my program to not run.

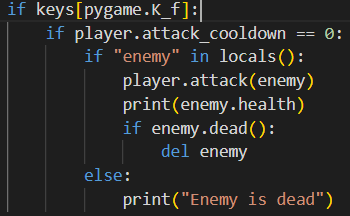
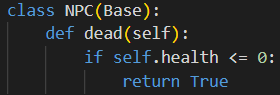
Made test things they work



 failed attack, multiple hits when tapped



Limit attacks to one every 0.5 seconds



Enemy can die when health reaches 0

# Bibliography

|  |  |  |
| --- | --- | --- |
| <https://hermitcraft.fandom.com/wiki/Decked_Out_2> | Research (p4) | 16.4.24 |
| <https://en.wikipedia.org/wiki/The_Binding_of_Isaac_(video_game>) | Research (p5) | 23.4.24 |
| <https://en.wikipedia.org/wiki/Dicey_Dungeons> | Research (p6) | 23.4.24 |

# Notes (temp)

**This section and all purple text will be removed**

**Analysis**

***Research***- *more games*

***Success criteria-*** *justify*

***Problem identification-*** *explain deck-building*

***Computational solution-*** *explain why and justify*

***Stakeholders-*** *describe how they will use the product, why it is appropriate to their needs, justify*

***Hardware requirements-*** *specific ones*

|  |  |
| --- | --- |
| ***RAG*** | ***Analysis*** |
|  | What are you going to make? |
|  | Why is the problem suited for a computational solution rather than a physical one? |
|  | Who is the solution aimed at? |
|  | How will they use it? |
|  | Why do they need it? |
|  | What are existing solutions to similar problems doing? |
|  | What could you use from this in your solution? |
|  | What do the stakeholders want to see in your solution? |
|  | What are the issues you might face when making it? |
|  | What are the essential features for the solution? |
|  | Why are they essential? (surveys, client feedback) |
|  | What are the hardware and software requirements? |

r- not started a-in progress g-complete to the best of my abilities so far

**Design**

Initial- flowcharts, pseudocode

Database- entity relationship model

First GUI design- paper/powerpoint/canva

Computational thinking- decomposition, abstraction, pattern recognition, backtracking

Testing plan

Program flowchart

Usability- justify (add onto GUIs)

Class diagrams

|  |  |
| --- | --- |
| ***RAG*** | ***Design*** |
|  | What smaller problems can it be broken down into? |
|  | Why are these smaller problems suitable for computational solutions? |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**GUIs**

Menus- Main menu, camp pause, dungeon pause

Maps- Camp, dungeon

Item menus- Tent (cards, deck, storage, inventory), camp shop (consumables, one use only cards), dungeon shop (cards)

Dungeon- player health, power-ups tracker, deck tracker (amount of cards left only), hotbar, artifact compass (stored in hotbar but always visible until artifact is retrieved)

**Decomposition**

Entities- Player, enemies

Player- WASD, sprint, attack, interact, *limited forwards only vision*

Enemies- fightable, avoid*, all limited areas*

Items- consumables, cards, artifacts, *keys*

Consumables-

Cards- some insta draw

Maps- camp, dungeon (camp move to menu, time dependant)

Camp- tent (inventory), item shop, dungeon entrance, interactive campfire (why not)

Dungeon- entrance, exit, traps, artifact locations, layout, treasure spots, crumble (hazard), *enemy zones*

Menus- Main menu, camp pause, dungeon pause

Main menu- title, buttons (play, options, quit)

Camp pause- paused, buttons (resume, options, help, main menu)

Dungeon pause- blackout screen/covering background, paused, buttons (resume, options, *false main menu*)

Crumble (hazard) (1/3 activation chance at consistent intervals, random door selected to close/each door has 1/6 chance to close) (must leave one hazard free path so exit is always possible)

**Development and testing**

**Stage 1**

What is being made?

Main menu with functional quit button and play

What tests?

All buttons are tried and corrected if not working

Prototype functionality? (function of the stuff made in this stage)

Completed the main menu which is the first thing to load upon opening the program to provide a basis for everything else to build off

**Stage 2**

What is being made?

Cards database and card selection

What tests?

All cards can be selected and will only allow legal deck builds (max amount of cards)

Prototype functionality? (function of the stuff made in this stage)

A base to build the main game off as this section loads first and allows for card functionality to be added

**Stage 3**

What is being made?

Main game with player controls, enemy ai and collision

What tests?

Player can move in all directions, not go out of bounds, can attack, enemies are able to move around without going out of bounds, enemies can track the player when close by, enemies can attack the player who will take damage

Prototype functionality? (function of the stuff made in this stage)

Basic gameplay features will be completed which allows for additional features to be added

**Stage 4**

What is being made?

Cards powerups and tracking

What tests?

Cards are selected at consistent intervals and each powerup gives stated buff

Prototype functionality? (function of the stuff made in this stage)

Players will be able to have their gameplay affected by these cards and powerups

**Stage 5**

What is being made?

Artifacts, tracking and scoring

What tests?

Artifacts spawn randomly in one of the set locations each time, are accurately tracked by the compass and give the correct amount of points upon successful collection

Prototype functionality? (function of the stuff made in this stage)

Adds the main objective of the game and creates the scoring system

**Stage 6**

What is being made?

Camp and shops

What tests?

Players can move freely around the camp without going out of bounds or getting stuck, shops are fully interactive and purchases can only be made if sufficient currency is available, purchases go into the players storage and can be equipped and used within the dungeon

Prototype functionality? (function of the stuff made in this stage)

Players have an interactive area to be able to buy and equip items before the game starts to provide a more immersive experience