OCR GCE A

COMPUTER SCIENCE PROJECT

H446-03

Name: Osas Osaghae

Candidate Number: <INSERT CANDIDATE NUMBER>

<Institution Name>: Salford City College Pendleton Center

Title of Project: Platformer Game

H446-03 – Project CONTENTS

Table of Contents

[A. Analysis 4](#_Toc114610638)

[Problem Description 4](#_Toc114610639)

[Stakeholders 4](#_Toc114610640)

[Justification of Computational Features 5](#_Toc114610641)

[Why this game is going to be developed with a computer. 6](#_Toc114610642)

[Abstraction 6](#_Toc114610643)

[Decomposition 6](#_Toc114610644)

[Thinking Ahead 7](#_Toc114610645)

[Thinking logically 7](#_Toc114610646)

[Research 7](#_Toc114610647)

[Interview 8](#_Toc114610648)

[Super Mario bros 10](#_Toc114610649)

[Fancy Pants Adventure 11](#_Toc114610650)

[Features 14](#_Toc114610651)

[Limitation 23](#_Toc114610652)

[Hardware and software requirements 23](#_Toc114610653)

[Success criteria 24](#_Toc114610654)

[B. Design 26](#_Toc114610655)

[Systems diagram 26](#_Toc114610656)

[Summary OF PROCESS 27](#_Toc114610657)

[Key variables and data structures 29](#_Toc114610658)

[development test data 29](#_Toc114610659)

[algorithms 30](#_Toc114610660)

[post development test data 30](#_Toc114610661)

[SIGN OFF PROPOSAL 31](#_Toc114610662)

[Developing the coded solution (“The development story”) 31](#_Toc114610663)

[Level generation 07/09/2022 35](#_Toc114610664)

[Prototype 1 36](#_Toc114610665)

[Prototype 2 37](#_Toc114610666)

[Prototype 3 38](#_Toc114610667)

[Prototype 4 40](#_Toc114610668)

[Player movement 14/09/2022 44](#_Toc114610669)

[Prototype 1 44](#_Toc114610670)

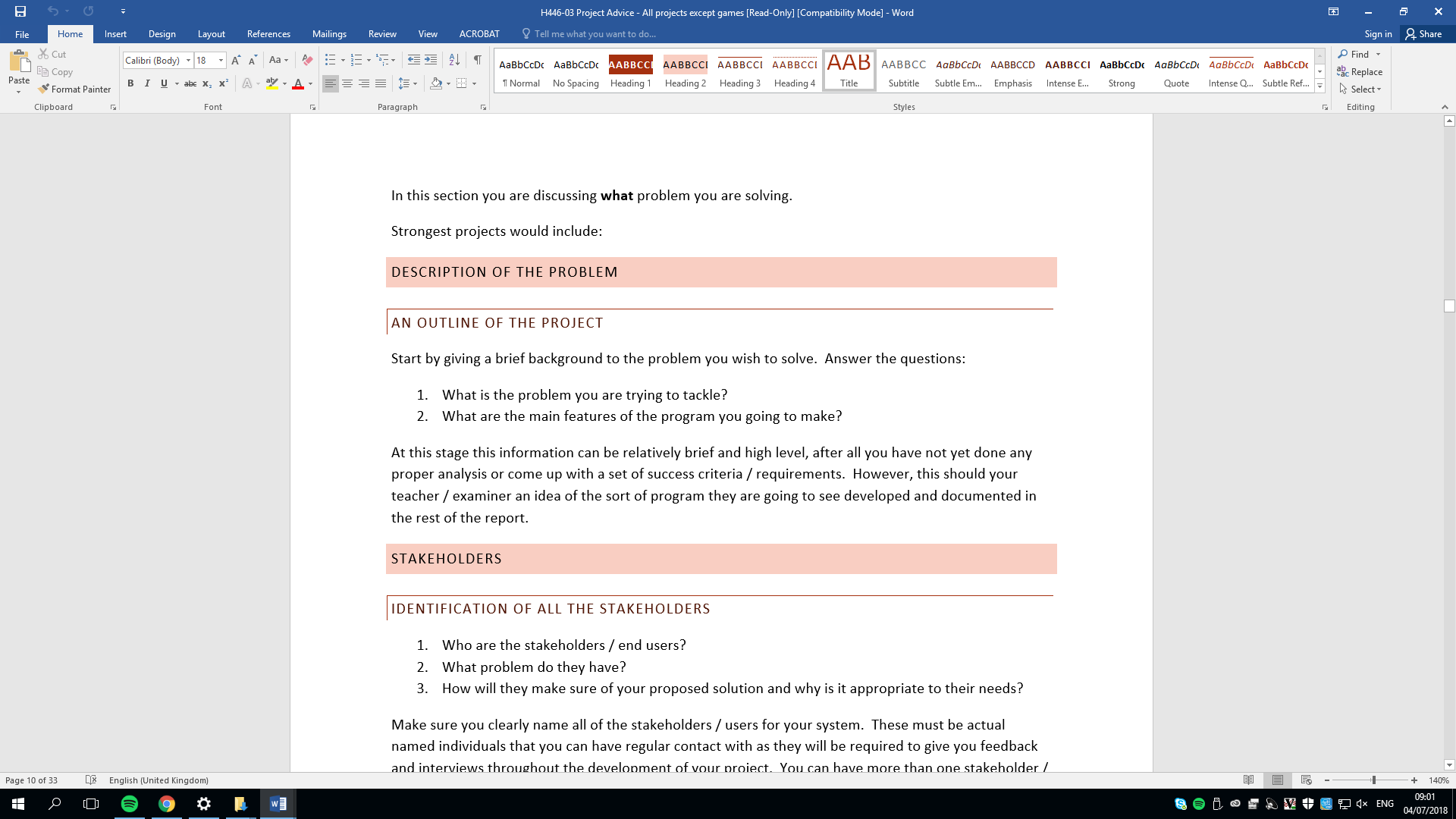
[B. Evaluation 45](#_Toc114610671)

[Evaluation-final solution 46](#_Toc114610672)

[Project Appendixes 47](#_Toc114610673)

# Analysis

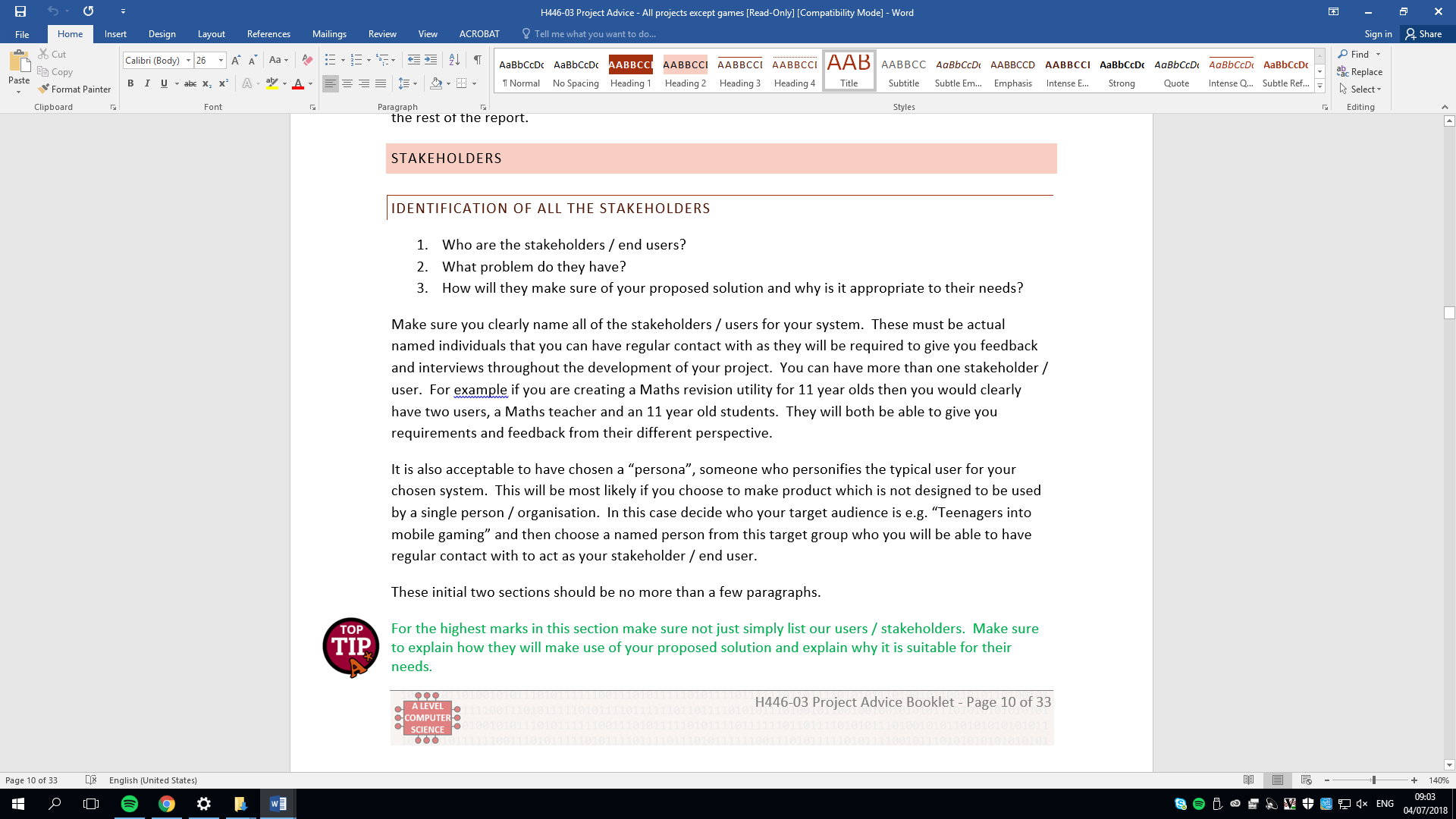
## Problem Description



In this project I am going to create a 2D single-player platformer game like Super Mario Bros using python. In this game the user is required to reach the goal at the end of the level to do this the user needs to control a character in the screen using certain keys to run and jump across platforms and avoid danger like pits spike-traps enemies. If the character collides with any of these dangerous objects it dies, loses a life and then it will have to restart the level from the beginning. Once this happens three times in one level the user will have to restart the whole game from the first level. The game has multiple level and in each level is designed differently so the course and platforms towards the goal the user need travers will be different.

This game is tailored for people between the age of 12 – 20 because of this this game will have a kid-friendly design however have a difficulty that would interest young adults. Also, this game will be able to be played by any computer that can run python code.

## Stakeholders



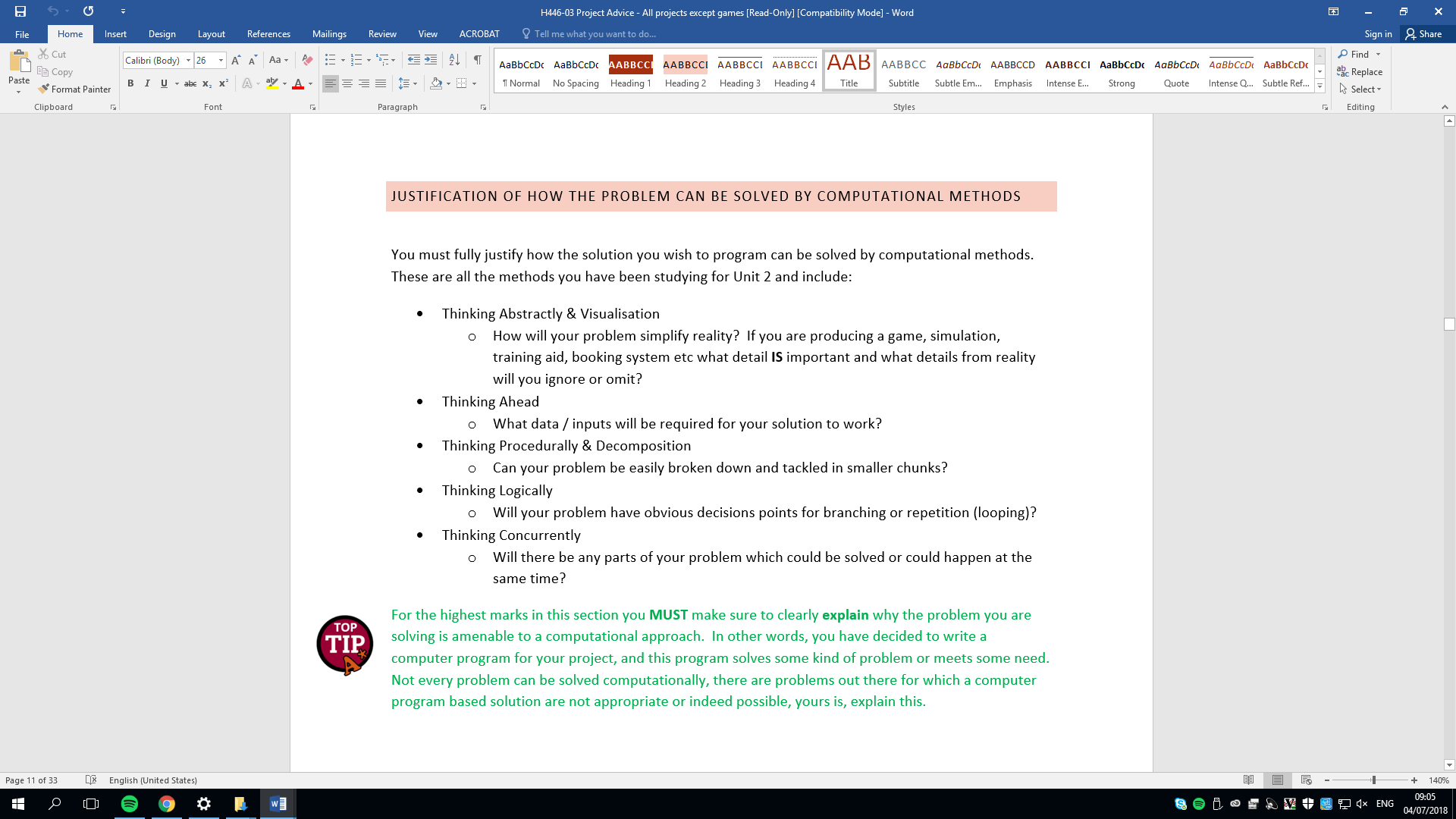
My stakeholders will be Abdullah and Jacy.

Abdullah is a suitable stakeholder because he is a gamer who is into games made for people aged 13-16 years which fits a side of the target audience for this game. So, he would be able to give me feedback from a perspective of a younger individual. He also has experience on developing 2D platformer games as he previously developed one. I could benefit from his experience because he can help give me some ideas. To give an example he could help give ideas on each levels design which will help my levels look better and also, he could give me quality feedback on the mechanics that I have implemented like: is the gravity too high? Are the platforms too small? Should the player move more faster? etc

Jacy is another stakeholder. He is suitable for this position because he has experience playing games for young adults aged 17 – 20. This is good as he would be able to give me feedback of my game from a perspective of an older individual. He also studies photography. I could benefit from his experience of it because he would be able to give high-grade suggestions on the graphics of the game. For instance, he can give me good suggestions on how the background of the game should look like from his experience of taking pictures of many platforms and landscapes in his studies. He can also suggest good sprite images I could use for the player, enemy platforms and dangerous objects.

The stakeholders are also in my computer science in class in college, so I would be able to easily contact them for any feedback on updated versions of the game faster. They are also gamers so they can give me features I could add into my game for other games they have played in the past.

## Justification of Computational Features



### Why this game is going to be developed with a computer

To develop this project, I will be using a computer. This is because my project will be a game that will require many calculations to be performed at very fast frequencies which a computer is best suited for this purpose. Moreover, it is much easier to develop games using computers because there are many game development tools in computers that are already made which saves me development time making these myself. Lastly, potential distribution of this game will be much easier if I develop it in a computer because computers are commonly used these days.

### Abstraction

In this project abstraction can be particularly useful in order to remove any unnecessary information in the game this is because it saves me time developing features that do not directly fit the theme of the game. For example, to save time me creating a pixel perfect complex collision system around the player image that reacts to the environment I could simply put a rectangle around the player image and dictate how the player should react to other objects like planks. Another thing I could abstract out to save valuable time and reduce complexity is the unnecessary details in the background for example I could remove wind mechanics and remove fine texture in the blocks or I could take out realistic physics from the game like how the weather can impact friction which will make it harder for the player to traverse the levels so I can focus on developing the main game rather than features that are not needed for the finished game and save computer resources so the game can run faster and without stuttering.

### Decomposition

While developing this game I am going to use decomposition to structure my game so the development can be well structured into manageable smaller fewer complex parts. This would things easier as it allows me to work on smaller problems to solve one at a time that would build up to the full game rather than me trying to solve all the problems at once. It also allows me to see my progress and give a rough estimate on how much is needed to finish the game.

In this project I could split the game into these modules:

Player Sprite

* Would handle how and where the player is presented in the screen
* Health System
* Player Movement

Block Sprite

* Would handle what the surface the player is standing on should look like and where it is presented in the screen.

Game Class

* Handle setting up each level by creating all the necessary sprites and background design
* Background music
* Handle collision system and how each object should interact with each other
* How the HUD should look like
* User Input

Menu Class

* Handle process to initialise the game
* Handle Special user settings like how loud the background music should be, how loud the sound should be overall, Game Mode.
* User input

### Thinking Ahead

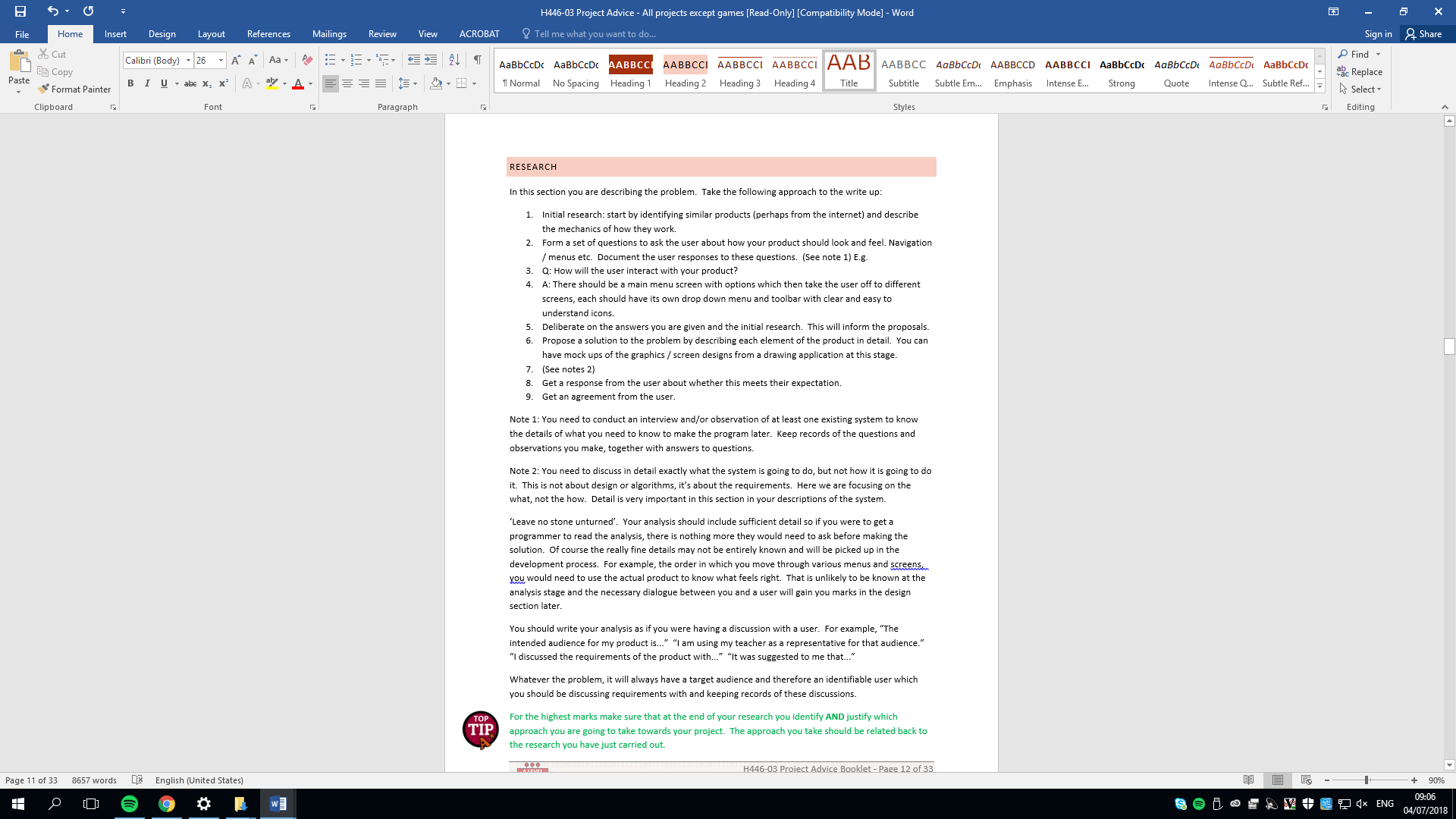
In this game the user will be required to input certain keys in the keyboard to be able to navigate the game and control the character. For example, for player movement the user will have to use keys on the keyboard to be able to jump, move left and right to be able to traverse the level to reach the goal. Also, the keyboard will be used by the user to for special game commands like quit, back to main menu, pause or restart current level. The mouse will also be used but it will be to mainly navigate the game like clicking buttons in the main menu to be able to start the game, select levels to play, change game settings or quit back to desktop.

Additionally, as this project will be a video game, there will be some output screens, so the end user is able to see the game and understand what is going on. An example that will be in the game is the HUD, this enables the user to see the players vital statistics like health, points, bonus attributes and level. Other output screens that will be used are main menu to be able to navigate the game and settings to see the current mode of the game.

### Thinking logically

To be able to play this game, the user will be required to make some decisions to progress the levels, change game modes and settings, pause the game if necessary or quit back to desktop. For example, once the game is booted by the user the game will display a menu of different options like start new game, select level, open controls, and quit to desktop.

## Research



### Interview

**Jacy**

|  |  |
| --- | --- |
| Name 2 2D platformer games you have enjoyed playing before? | Super Mario Bros, Fire boy and Water girl |
| Is difficulty something you would enjoy about a game?   * If yes, from 1-10 1 being very easy 10 being extremely hard, rate how hard should an enjoyable game be? | Yes.   * 8. I would like the game to be reallychallenging for me but however I still want to be able to complete the level. |
| Do you like customisation? | Yes   * Some sort of costume change for the player. It could be as simple as changing the colour of the player clothes or allowing the option to pick different players |
| How should the colour design be for the menu? | I think the menu should have a dark theme so maybe use dark colours like dark blue and black use a lighter tone that fits the dark theme for the buttons as well to make the buttons visible. |

**Abdullah**

|  |  |
| --- | --- |
| Name 2 2D platformer games you have enjoyed playing before? | Super Mario Bros and Fancy Pants Man Adventure |
| Is difficulty something you would enjoy about a game?   * If yes, from 1-10 1 being very easy 10 being extremely hard, rate how hard should an enjoyable game be? | Yes   * 5. I want the game to be a bit challenging so I can invest some time in the game to beat the level. However, I don’t want the level to be so hard it is frustrating to play. |
| Do you like customisation?   * If yes, what sort of customisation would you like to see | No. I mainly care, and I am more interested about how the game actually plays. Like Player movement and Enemy Behaviour. |
| Any specific mechanics you would recommend me to develop and why? | Probably a powerup system. I tend to like games that have a sort of powerup item because it allows you to complete levels in different creative ways. Also, its generally fun to play around with power ups. |
| If I was going to add an account system what would you recommend me to do. | You could store account details in a normal text file. Even though its not a secure way to store account details they are not very significant as this is not going to be a very competitive game. |

**EVALUATION OF INTERVIEW**

From Interviewing Jacy and Abdullah, I have gathered some information to plan out what I am going to implement into my game. Firstly, I will research the game Mr. Fancy Pants and Super Mario Bros since they are games my stakeholders are interested into magpie some ideas from them games and add to my game.

When designing and developing the game I will make it a bit challenging because my stakeholders suggested that difficulty increases user engagement and interest in the game

If I have time during development of the game, I will add a customization feature in the options menu maybe to customize player sprite clothes as Jacy says this is a good feature to implement. Also, Abdullah suggested I should add a powerup system similar to Mario to increase creative ways levels can be completed.

### Super Mario bros

Graphical user interface

Description automatically generated with low confidence

Super Mario Bros is a popular old 2D platformer game. In this game the user controls a character in the screen called Mario with specific keys on the keyboard the aim of the game is to reach the end goal flag.

**Movement**

In this game the user controls Mario with the keyboard. Up to jump down to duck, left and right to move left and right. I like this because it is straight forward and is commonly used within games which will avoid confusion for the user when learning the controls. Also, in the game when Mario stops moving or tries change direction there is a slide mechanic. I do not like this because it can be annoying and makes the game harder to play when moving or landing on small platforms.

**Dangerous Objects / Enemies**

While going to the end goal Mario will have to avoid dangerous objects like pit fall and enemies like (goombah, Kooper, piranha plants, etc). I may try to implement this system of killing enemies, so it gives the user an option to remove obstacles and I could add a reward system with this for example If the player kills an enemy, they get a point. If Mario contacts any of the dangerous objects he dies and will have to restart the level from the beginning and when he dies twice the user will have to restart the whole game from the beginning. I do not like this feature in the game because the game cannot be enjoyed by players who want to casually play the game. Instead, maybe I could make this an option in the settings where the user could select chose a difficulty and if the user selects hard mode this feature is activated

**Powerup**

Mario can also pick up powerups if he jumps and collides with the bottom of the question mark blocks. Some examples of power ups that Mario could pick up is the mushroom, fire flower and the super star. The mushroom is makes Mario double in size and gives Mario an extra life so if he comes into contact with a dangerous object, he returns into his original state without the user having to restart the current level. The fire flower transforms Mario into fire Mario state, this state allows Mario to fire fireball projectiles, with this projectile he is able to any enemy. The Super Star transforms Mario into Invincible Mario, this state Mario is invisible, and he can defeat all enemies by touching them. I like this feature of Mario because it adds variety to the game and allows levels to be complete in different ways

**Summary**

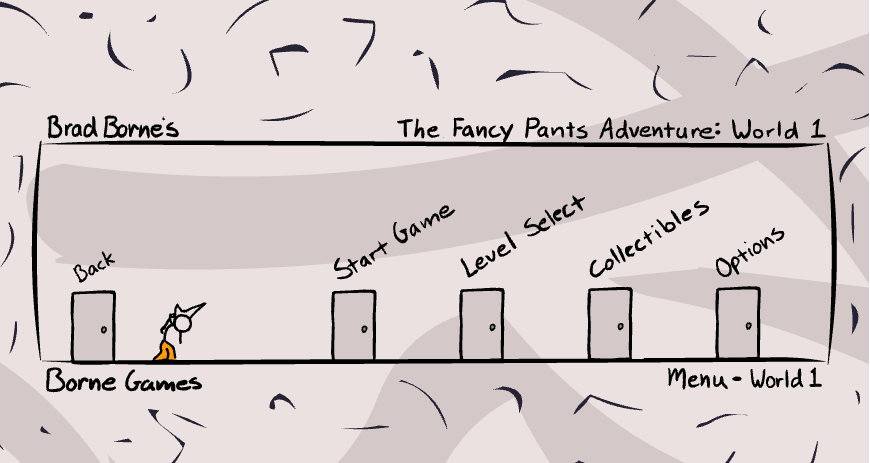
With the player movement, I like this game simplistic controls that most people will be able to understand. However, I don’t like the sliding mechanic it has because it can be annoying when moving around small blocks.

With the enemies, I like the way the player can kill them by jumping on top of them because it is a very kid friendly way to eliminate an enemy it suits my game audience.

In how the level runs, I don’t like the way if you die 3 times in the same level you have to restart the whole game. So instead of putting this in the normal game I could allow the user in the main menu to decide if they want to play the game in hard mode or easy mode and Hard mode will have that feature.

I also like the powerup system in this game that gives the player unique abilities

### Fancy Pants Adventure

Fancy Pants adventure is a side scrolling 2D platformer based on parkour. In this game the user controls a character called Fancy Pants Man. The aim of the game is to defeat the boss at the end level which completes the game.

**Level/Enemies**

In order to defeat the boss and complete the game, the user must explore and navigate the map to find doors that lead to the next level some doors lead to bonus level. While the user tries find doors into the next level, he must avoid enemies and dangerous objects. If the user comes into contact with these enemies the player loses 5% of its health and once the health bar is empty the player fully dies and respawns at the beginning of the current level. Examples of enemies in this game are Spiders, Snails and Mice that shoot guns. However, the user can defeat these enemies it jumps and lands on top of them.

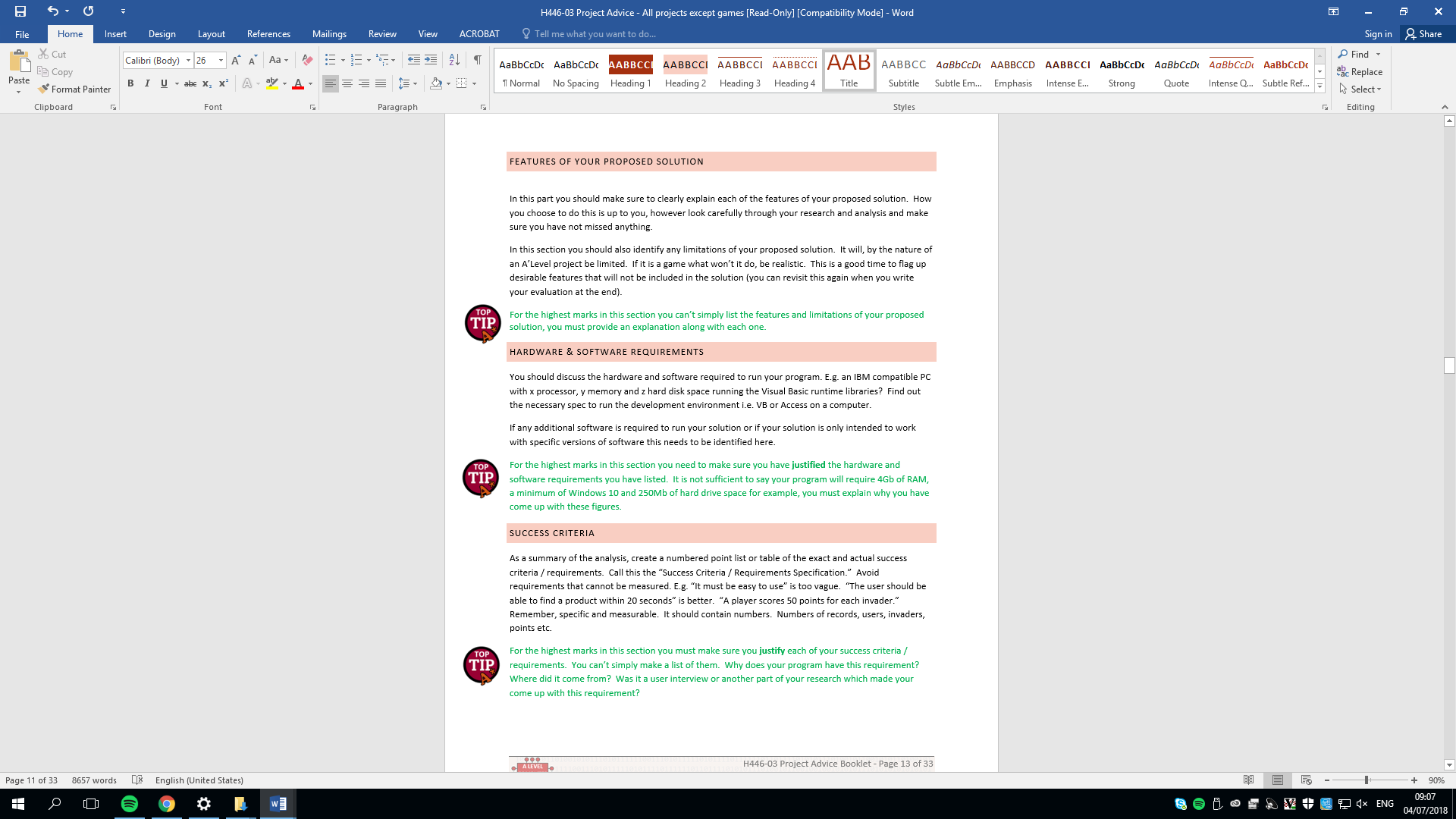
**Menu**

Fancy Pants Adventure features an interactive menu. When the game begins the game loads the user into a menu. The menu is navigated in the same way the user will navigate around the in-game levels, the menu has 5 doors (4 if it is the users first time), to enter the doors the user needs to stand on the door and press S. Each door is labelled to shows the user what opening the door does. The Start Game door starts a new game and put the user in the first level. The Level Select Door puts the user into another menu with a set of doors representing each level, Collectibles door puts the user in a room that displays collectibles that have been collected during gameplay. Options door allow puts the user into another room that gives the user an option to change Fancy Pants Man pants colour and reset game data.

**Summary**

A thing I like about this game is how the game ends once you fight the final boss. I also like the way you need to explore the game map to find the doors that will lead you to the next level. I also like the selection of the enemies like spiders and mice who shoot scribble guns because it suits my target audience. I also like the menu design because it is very interactive and unique I may implement a similar interactive menu if I have enough development time.

## Features



Essential Features will be written in black.

All essential features will be prioritised over the desirable feature as all essential features are needed for the finished product of the game

Desirable Features will be written in green.

Desirable Features will only be implemented when all essential features are completed as this project is time sensitive

**Menu**

|  |  |  |
| --- | --- | --- |
| Feature | Justification | Success Criteria |
| 1. Menu | When the game first loads the menu will be displayed. The menu will give the user some options that can be picked using the mouse. | When program starts it displays the menu screen if login system isn’t implemented |
| * 1. Start new game | The first option will be to start a new game button. This starts the game and puts the player in the first level. | * Program puts the user in the first level of the game. * Program also positions player at the start point of the first level |
| * 1. Continue game | Continue game button starts the game and puts the player into the level they are currently on. |  |
| * 1. Select level | The third option is select level button. This puts the user into another menu that makes the user pick what level they want to start on.  If a login system is implemented the user can only select the levels, they have already completed in that account. | * Puts user in another menu * All levels are displayed * When the level is pressed that the user is loaded into that specific level. * The player is positioned at the start point of the level |
| * 1. Options | The fourth option is the settings button. When this button is clicked the player gets put into another menu and options like volume and difficulty.  I could add an option that allows the user to change their key binds | * The user is put into another menu * Menu allows the volume of the music to be changed by the user |
| * 1. Quit to Desktop | Finally, the button closes the game application and puts the player back to the desktop. | * When the user clicks this button, it closes the pygame window and ends the program |
| 1. Sound | There will be background music while the user navigates the menu. | * When the menu starts, menu music should be played * When the music ends it should loop back to the beginning |
| 1. Menu design | A way to add interactivity is to the menu is to allow the player to navigate the menu using the player like in fancy pants adventure game. For instance, the menu will consist of doors each door will be labelled as an option the user can take. To enter the doors the user will have to move the player to the door and press the O key on the keyboard to open the door. |  |

**Login/Account System**

|  |  |
| --- | --- |
| Features | Justification |
| 1. Login Screen | This feature could be a good addition to the game because it allows different users, using the same computer, progress to be independent from each other. If the login system is implemented when the program first runs the first login screen would load. In the login screen the player can pick two options. |
| * 1. Login | If this option is picked the user will be asked to their username and password. Once they are typed into the 2 text boxes the user needs to press enter in their keyboard to submit their details so they can be verified with the account details text files. After this the menu is loaded. |
| * 1. Create Account | If this option is picked the user will be asked to create a username and password using the text boxes. The account details will then be saved into a text file that contains all account details. After this the menu is loaded. |

**Gameplay**

|  |  |  |
| --- | --- | --- |
| Features | Justification | Success Criteria |
| 1. HUD | HUD stands for the heads-up display. The HUD will only be displayed during the game play. The HUD will display the time, health of the player and the current level number. | * During gameplay these should be shown:   + Player Health   + Timer * Current level number |
| 1. Sound | There will be background music while the user is playing the game. | * When the menu starts, menu music should be played * When the music ends it should loop back to the beginning |
| 1. Player | Features that involved |  |
| * 1. Movement      1. Slide Mechanic or Roll | In the game I will use some elements of Super Mario Bros movement. In Super Mario, the user controls the main character with certain keys on the keyboard. A to move left D to move right S to duck and W or SPACEBAR to jump.  A slide mechanic or roll could be implemented into this game like in Super Mario Bros and Fancy Pants Adventure. So, a way I could do this is when the player runs if he ducks at the same time it is able to slide or roll to a stop. This could be used to make more obstacles in a level to add variety to the game. | * W should make the player jump * A should make the player move to the left * D should make the player move right * S should make the player duck |
| * 1. Animation | There will be specific animations for movement. There will be a walking animation jumping animation and a duking animation.  Instead of using basic animations I could use the acrobatic parkour animation or something similar to make the player movement look more appealing. For example, backflips when the player jumps and rolls when the player lands | * Walk animation toward the direction of travel should be initiated when the player walk * Duck animation should be initiated when the player crouches * Jump animation should be initiated when the player jumps * Fall animation should be initiated when the player moves downwards |
| * 1. Health | The player will have 3 lives this will be visualised as 3 hearts in the game. Every single time the players collides with a dangerous object one life will be lost, this will make one heart disappear for the HUD. | * Health should be deducted by 1 when player collides with a dangerous object |
| * 1. Sound | A specific sound will be played when the player jumps or walks. | * Footstep sound should be played when the player walks * Jumping sound should be played when the player jumps |
| 1. Dangerous Objects |  |  |
| * 1. Enemy |  |  |
| * + 1. Movement | The enemy will have a basic movement. When the enemy first spawns in the game it will be given a horizontal direction to move when the enemy hits the wall it moves to an opposite direction.  A way I could make the enemy have more complex and make the game harder is to make the enemy chase the player instead of moving in a predictable path. | * The enemy should move when the game starts * The enemy should move opposite direction once it hits the wall |
| * + 1. Collision | If the player collides with the enemy the player loses a life.  I could add a feature I found in Fancy Pants Adventure where if the player collides with the enemy sprite the player gets launched to the opposite direction it was originally traveling. | * Enemy should deduct life of the player when it collides with the player |
| * + 1. Animation | The enemy will have the same animation style as the main player so when the enemy walks a certain direction a walking animation towards that direction will be triggered. | * Walk animation toward the direction of travel should be initiated when the enemy walk * Fall animation should be initiated when the player moves downwards |
| * + 1. Sound | There will be a sound that will be played when the player collides with an enemy to alert the user. | * A sound should be played when the enemy collides with the player |
| * + 1. Projectile Enemy | Another Feature I found in Fancy Pants Adventure is projectile enemies. If I was to implement this enemy type into the game the enemy will fire projectiles towards the direction of the player at regular intervals. This could be a good feature to add if I have enough time to make the game harder |  |
| * 1. Other Dangerous Objects (Spikes/Pitfalls) |  |  |
| * + 1. Collision | With pitfalls when the player falls of the screen the game is automatically over and the player will have to restart the level. However, with objects like spikes the player just loses a life. | * Dangerous Object deducts a life from player if they collide |
| * + 1. Sound | There will be a sound that will be played when the player collides with a dangerous object to alert the user. | * A sound should be played when the Dangerous Object collides with the player |
| 1. End Goal | The end goal will be a door once the player is at the same position as the door the user will have to press O to open the door to proceed to the next level. | * If the player is at the same position as the door if the O key in the keyboard is pressed the player moves on to the next level |
| 1. Powerups | This is a feature I found in Super Mario Bros I liked. These powerups add variety to the game which will make the game more interesting especially for my target audience. Powerups will be found in chests and will only last 10 seconds |  |
| * 1. Wings | This power up allows the player to remain in the air for longer to float when jumping. |  |
| * 1. Blaster | Gives the player a projectile weapon that can be used to kill enemies. |  |
| * 1. Athletes shoe | Makes the player faster and jump higher. |  |
| * 1. Star | Makes the player invincible, faster, jump higher and kill enemies when they collide. |  |

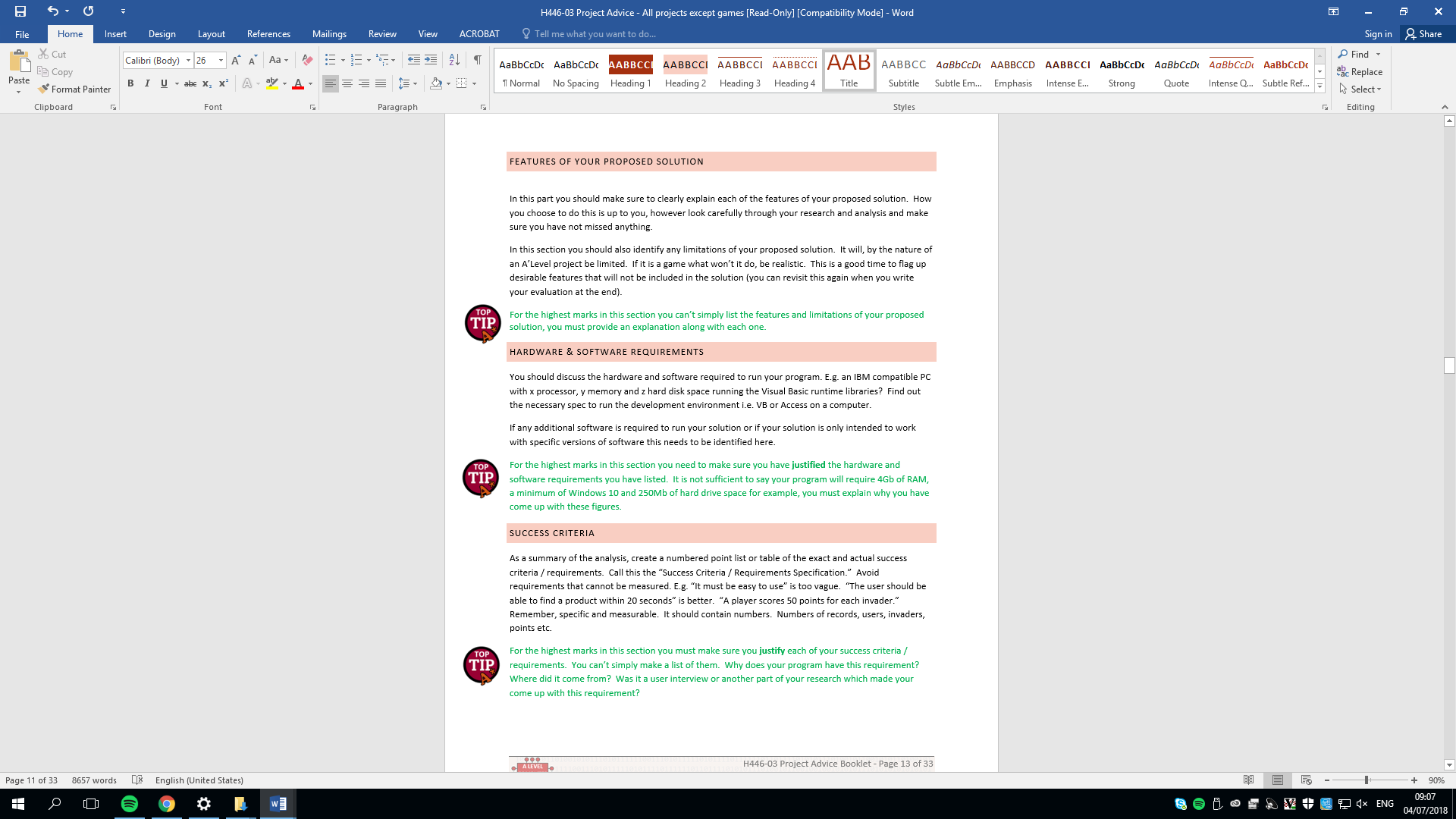
## Limitations

There are features in this game that may not be implemented. One of these features is online and multiplayer play. I am not going to implement this because the game is a single player game, all the levels are designed for 1 player and it will not need any online features to be able to connect to other Players or receive any information from the internet because all data will be stored in the machine the game is installed in. Also, as this project is time sensitive this removal will help me better focus on key features like menu and the actual gameplay to be able to meet the system requirement before the deadline of the project.

This game will also not feature any 3D graphics because this game is a 2D platformer. Also, the library I am going to be using isn’t designed for 3d games and doesn’t support 3D graphics. Moreover, I am familiar with using pygame so learning to using another library will be time consuming as I would need to find a suitable library that supports 3D graphics and learn how to code 3D graphics for the specific library which can get difficult to do.

Furthermore, this game will only be played on PC as the pygame library only supports PC operating system. If I was to try make this game be playable in multiple platforms like android devices, I would have to use something called pygame subset. Pygame subset allows for some pygame functionality on android devices. I will not be using these subsets because I do not know how to use them, and it will require me some time to learn and understand how to use them.

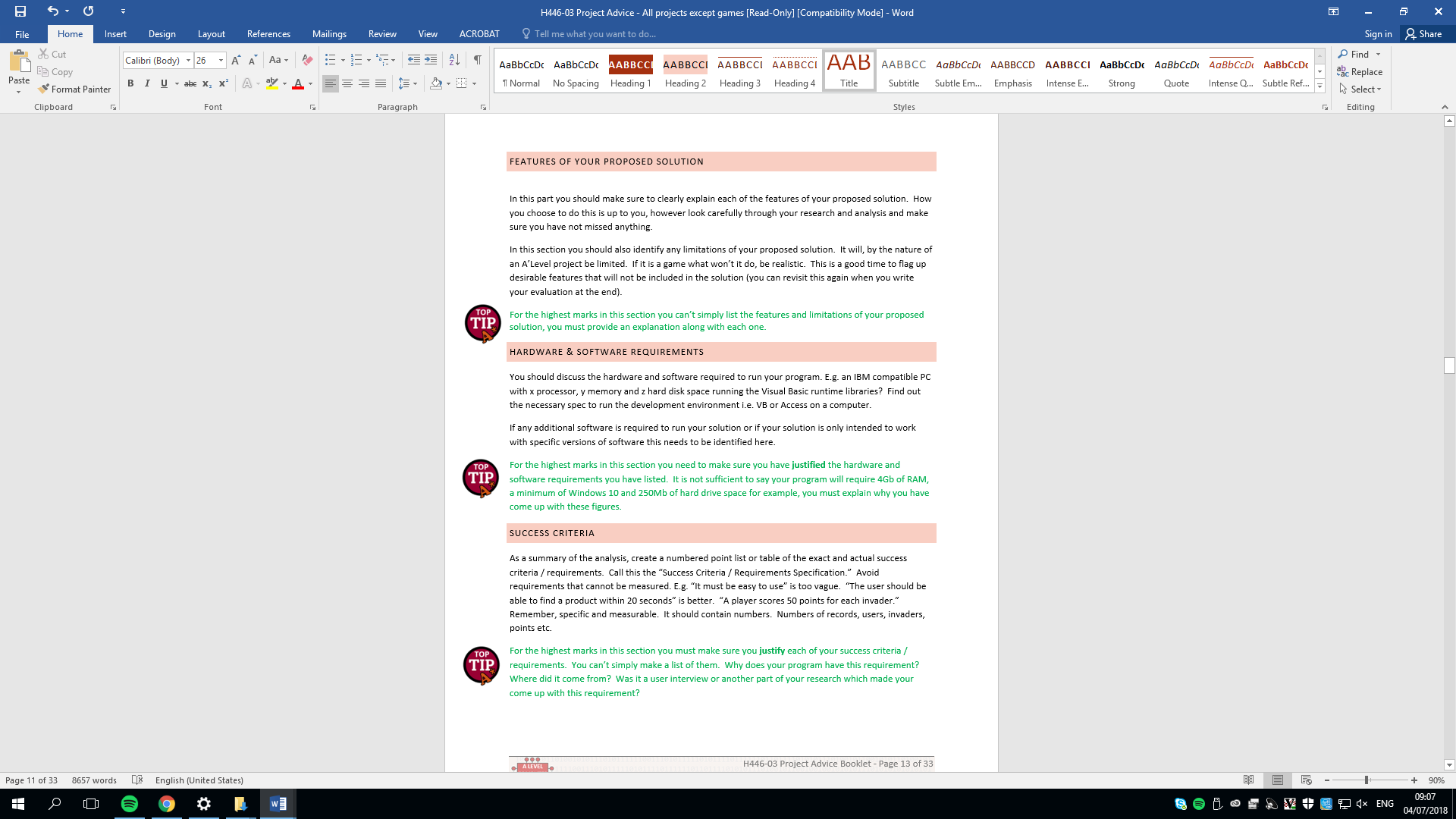
## Hardware and software requirements



* Hardware that your Trina use
* Y u going to use python
* Software library y u going to use me

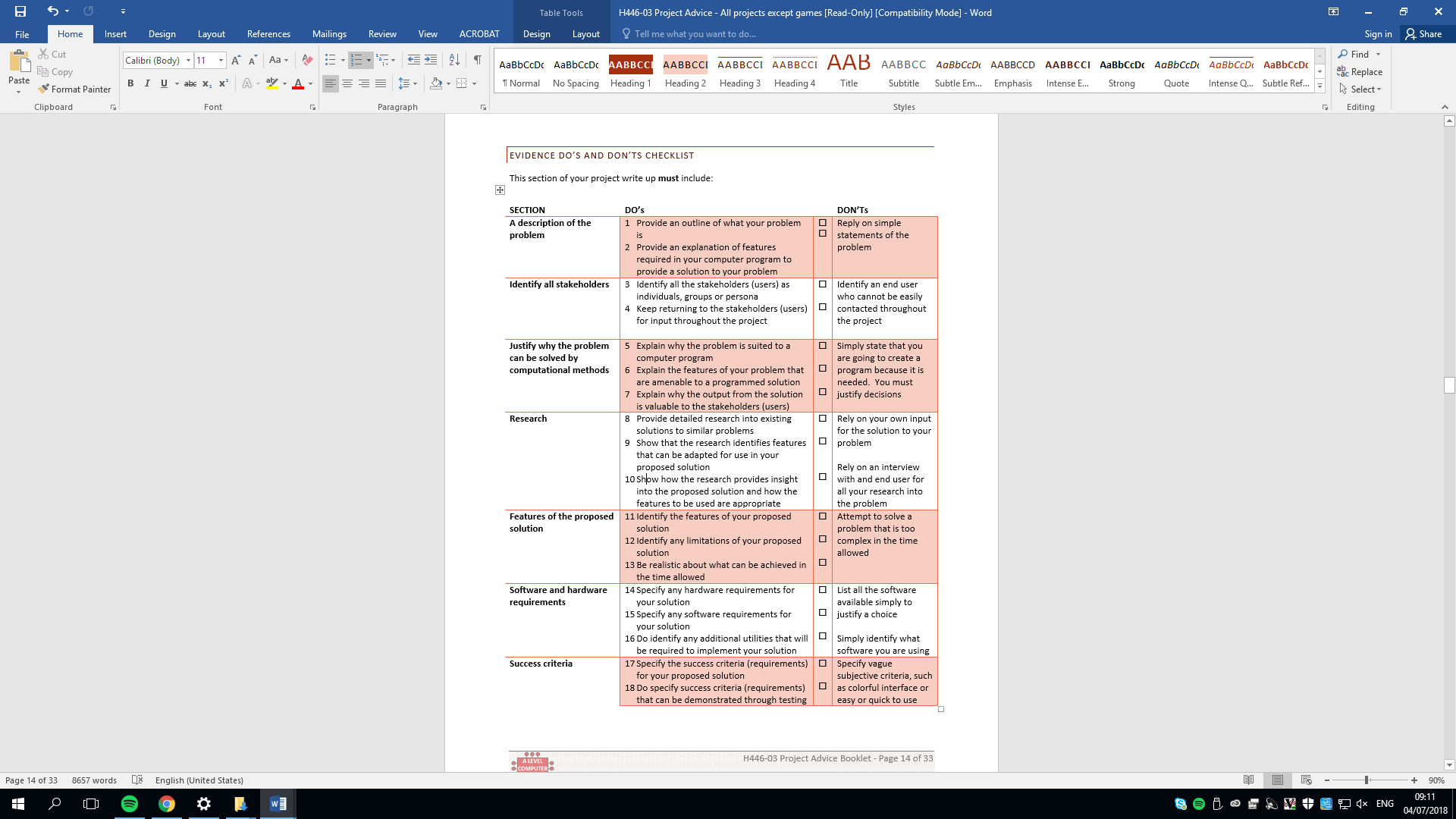
|  |  |
| --- | --- |
| Hardware | Justification |
| x86 64-bit CPU (Intel / AMD architecture) | This project is going to be developed using python and this is minimum requirement for running python code. |
| At Least 4 GB of RAM | This is required because to run low-spec games, like this one the user computer needs to have at least have 4GB of RAM to be able to run the code itself maybe run their own background tasks without decrease in performance |
| 5 GB free disk space. | This is required so the end user is able to download and install the game with its additional packages to translate the python code. |

|  |  |
| --- | --- |
| Software | Justification |
| Operating system like Windows 7 or 10, Mac OS (Operating System) X 10.11 or higher 64-bit, Linux: RHEL 6/7 64-bit. | These operating systems all support python which will be required by the end user to run the game |
| Pygame library | Pygame is a library that has python modules designed for writing games. I will be using this to develop my game because it saves me time making and debugging my own functions for this game as the library is well tested and is developed by experienced programmers. |



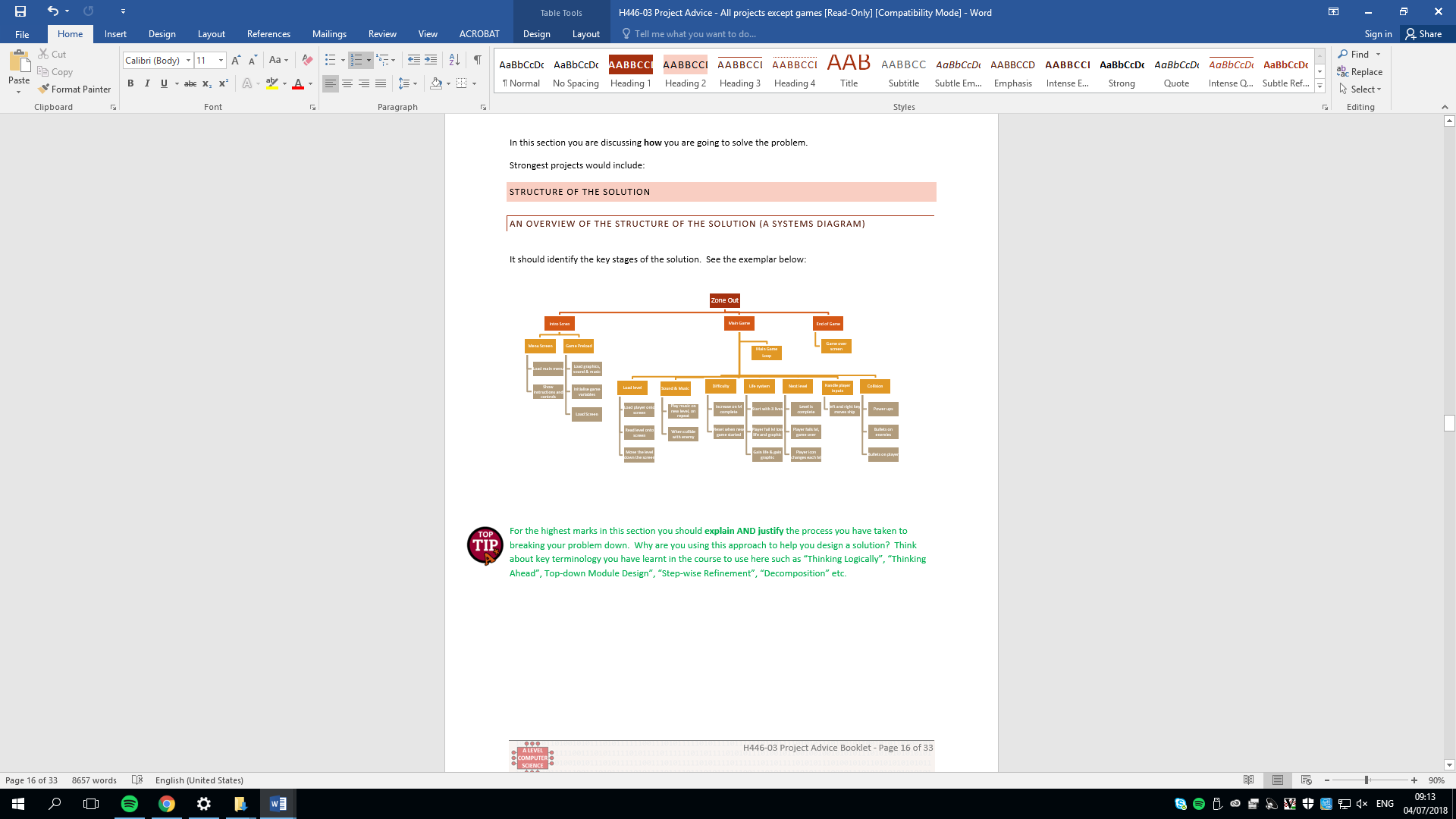
* List things that should be met that indicate success
* Measurable success criteria no wishy-washy success criteria like it opens quickly

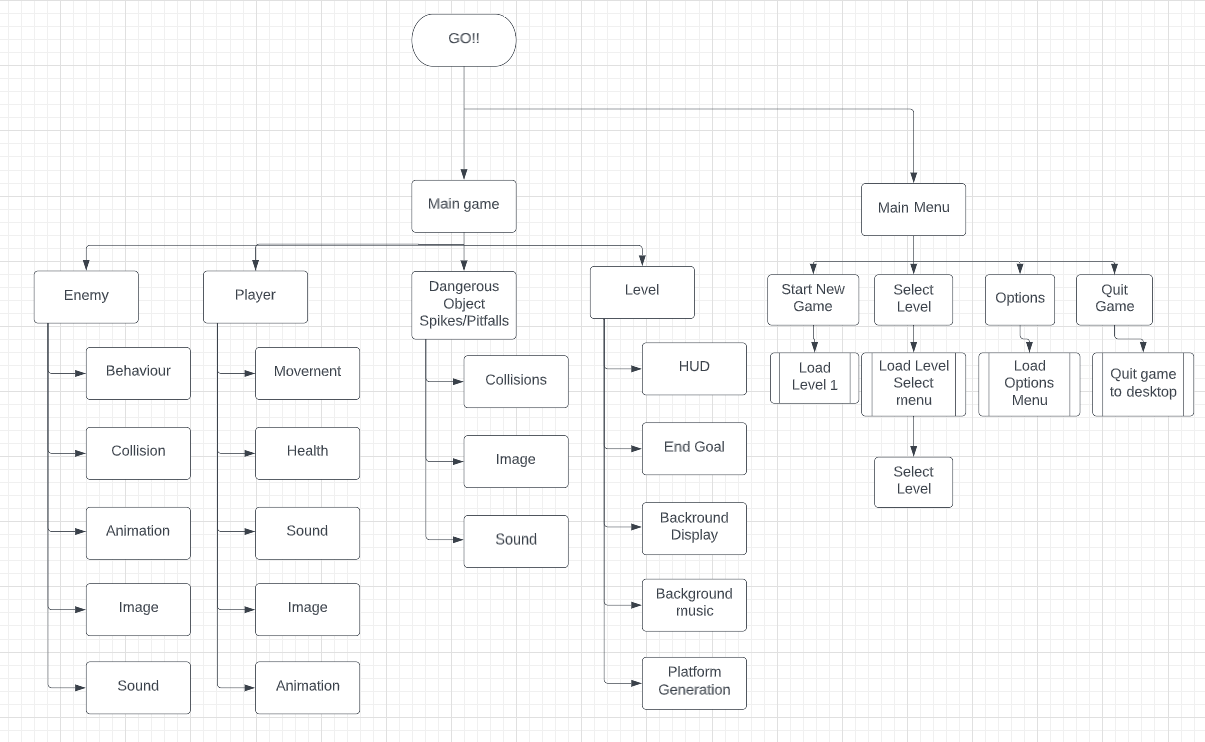
Checklist – remove at end



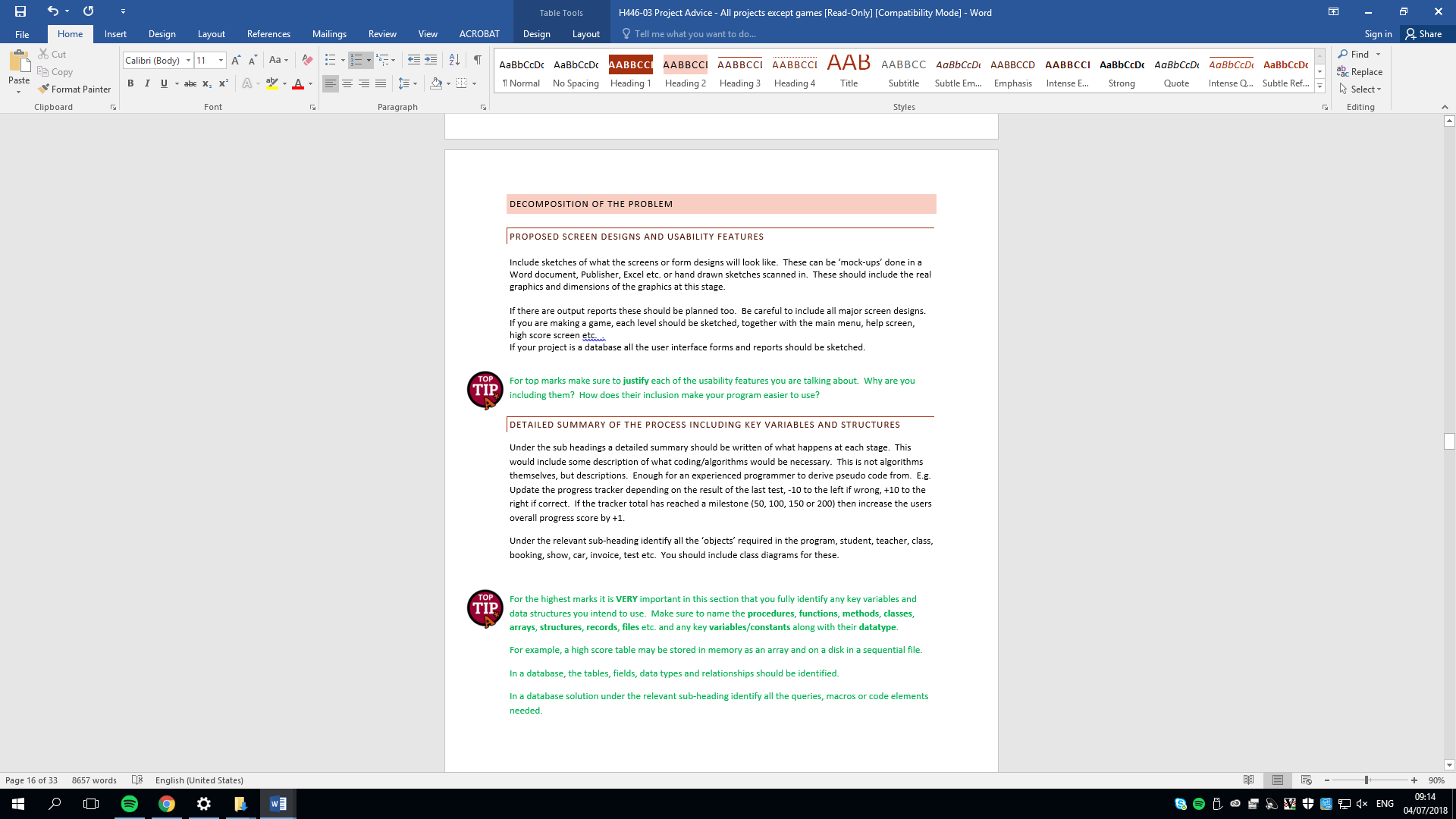
# B. Design

## Systems diagram





## Summary OF PROCESS



|  |  |  |
| --- | --- | --- |
|  | Process | Summary |
| Main | Start New Game | This option in the menu will start the first level |
| Menu | Select Level | This option in the menu will transport the player into another menu and allow the player to pick what level to play. |
|  | Quit Game | This option in the menu will quit the game application |
|  | Option | This option in the menu will take the player to another menu and allow the player to change volume of the background music. |

**Main Game**

|  |  |  |
| --- | --- | --- |
|  | Process | Summary |
| Player | Movement | The user controls the main character with certain keys on the keyboard. The A key will move the player to the left D will move the player to the right, S will make the player duck and the W key or SPACEBAR will make the player jump. |
|  | Health | The player has 3 lives. Every single time the players collides with a dangerous object one life will be lost, so in the player class there variable that holds the number of lives the player has left. |
|  | Sound | A footstep sound will be played when the player walks and a Jump sound will be played when the player jumps |
|  | Image | A player image will represent the rect of the player during the gameplay |
|  | Animation | The player will scroll through a series of images when moving to give a real-life movement illusion in the game meanwhile the only thing actually moving is the players rect. When the player moves right there will be a walking animation to the right and this will be flipped when the player moves to the left. There will also be a jumping animation and a fall animation as well. |

|  |  |  |
| --- | --- | --- |
|  | Process | Summary |
|  | Behavior | When the enemy first spawns in the game it will be given a horizontal direction to move when the enemy hits the wall it moves to an opposite direction. |
| Enemy | Collision | When the enemy collides with the player it deducts one health from the player class. |
|  | Sound | Sound will be played when the enemy collides with the player. |
|  | Image | An image will be used to represent the enemies rect. |
|  | Animation | Enemy will have the exact same animation as the player but with a different image. |

|  |  |  |
| --- | --- | --- |
|  | Process | Summary |
| Dangerous | Collisions | When the enemy collides with the player class it deduct one health from the player class |
| Objects  (Spikes\pitfalls) | Image | An image will be displayed where the dangerous object rect is displayed |

|  |  |  |
| --- | --- | --- |
|  | Process | Summary |
|  | HUD | During the game, information about the game will be displayed on the screen. For example, time, level number and player health. |
|  | End Goal | The end goal will be displayed as a door. Once the player is at the same position of the end Goal the player needs to press a key to proceed to the next level. |
| Level | Background Music | During the main game there will be music played. |
|  | Background Display |  |
|  | Platform Generation | In the beginning of the level there will be a program that reads a 2D array and depending of the content in each cell will determine where the platforms should be placed around the screen. |

## Key variables and data structures

|  |  |  |
| --- | --- | --- |
| **Key variables** | **Type** | **Meaning** |
| DISPLAYSURF | class | This variable will store the main game surface where all the sprites, background images and HUD will be drawn to. |
| Run | Boolean | Stores a Boolean value. If the value is true the game loop continues if its false the loop terminates |
| levelMap | list | The display is split up into a grid. The 2D list represents the whole screen. Each cell in the list will contain a number that shows what should be displayed in the screen. 0 means nothing, 1 is a player, 2 is a block, 3 will be an enemy |
| blocks | class | This variable is a class that will store a block sprite. |
| Players | class | This variable is a class that will store the player sprite. |
| game | class | This variable will store the game class. This class will store functions and variables necessary for the gameplay like levelMap, setUpLevel(), draw(). |
| BlockGroup | class | A class that acts as a container that stores all the block sprites |
| PlayerGroup | class | A class that acts as a container that stores the player sprite |

## development test data

**Level generation Test**

|  |  |  |
| --- | --- | --- |
| **Num** | **Description** | **Expectation** |
| 1 | This code should open a pygame window named “Test Window” and make it 640x704 pixels wide | * Open 640x704 pygame window * Name it “Test Window” |
| 2 | Code allows the user to close the pygame window | * Close pygame window if esc key is pressed or the top right X button |
| 3 | This test is to see if the code I have added will draw the platform into the pygame screen with the given size and position | * White block 100x100 pixels big should be displayed at the position y:50 x:50 |
| 4 | This test is to test if the setUpLevel() function gives the block sprites the correct position according to the level map. | * The screen should have a similar pattern to the self.levelMap where there is a 2 a white block sprite should be in that position |
| 4.2 | I am going to change one of the 0 into 7 and check if that cell creates any errors or puts a white block in that position | * No changes to the level design or errors |

**Player Movement**

|  |  |  |
| --- | --- | --- |
| **Num** | **Description** | **Expectation** |
| 1 | Test is to see if the player is displayed in the screen in the correct position in the screen | * Red rectangular player smaller than the platform blocks at the start position given by the map |
| 2 | This test will see if the player is able to move when the user uses the left and right arrow key | * If the left arrow key is pressed the player will move left * If the right arrow key is pressed the player will move right |
| 3 | This test will see If the player is prevented from moving off screen | * If the player tries moves past the right edge of the screen the player will appear to stop moving * If the player tries moves past the left screen of the screen the player will appear to stop moving |
| 4 | This test is to test if the setUpLevel() function gives the block sprites the correct position according to the level map. | * The screen should have a similar pattern to the self.levelMap where there is a 2 a white block sprite should be in that position |

**Player collision**

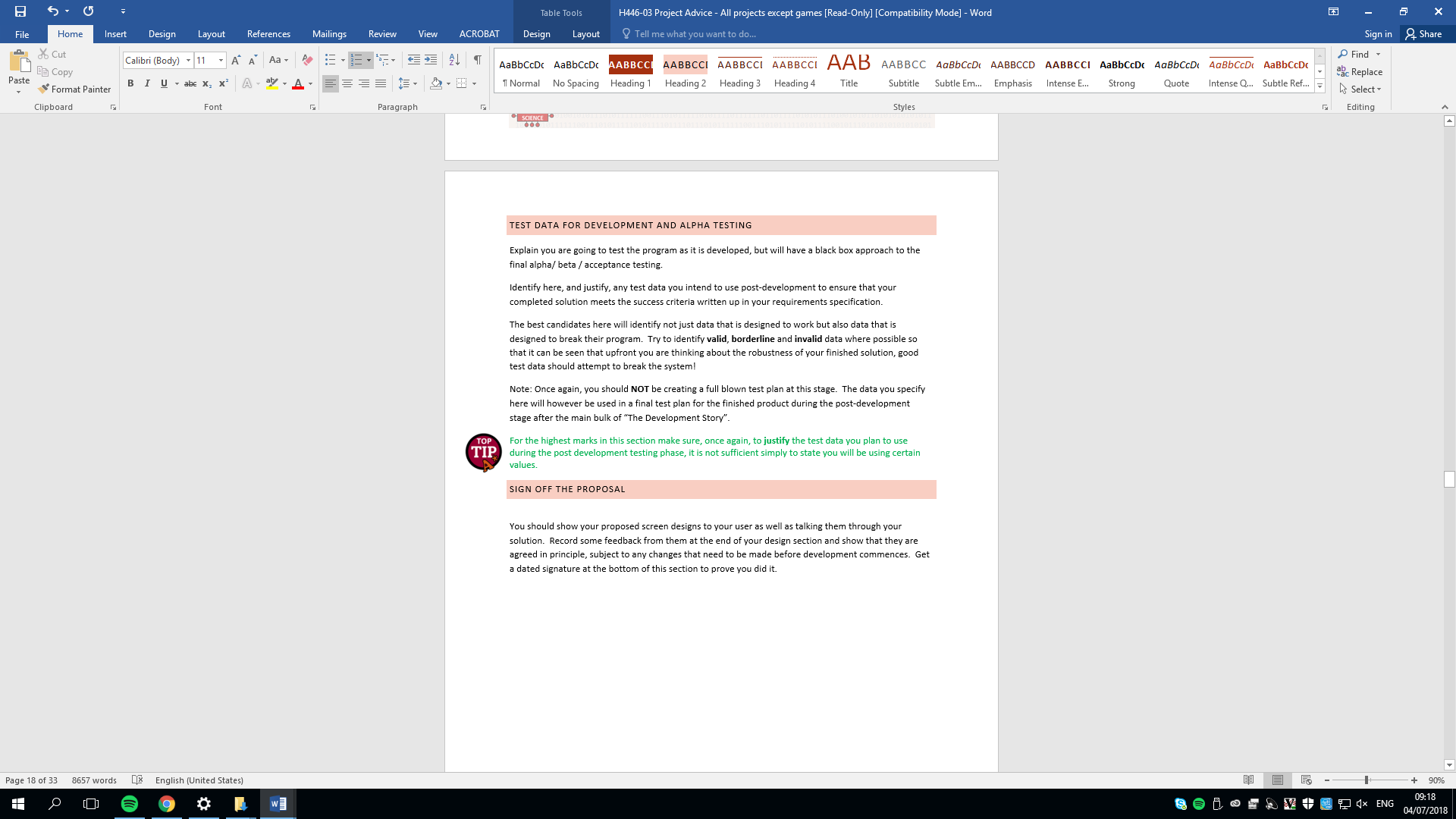
|  |  |  |
| --- | --- | --- |
| **Num** | **Description** | **Expectation** |
| 1 | This test will see If gravity is applied at all times unless the player jumps | * Once the game runs player should be falling down * While the player falls down if the spacebar is pressed the player jumps |
| 2 | This test will see if the player is able to stand on top of a block | * The player doesn’t fall through the block sprites * Player can move left and right on top of any block * Player can jump when on a block |
| 3 | This test will see if the player stops moving when the player reaches a side of a block | * The player stops moving when the user tries move through a block in the x axis |



## algorithms

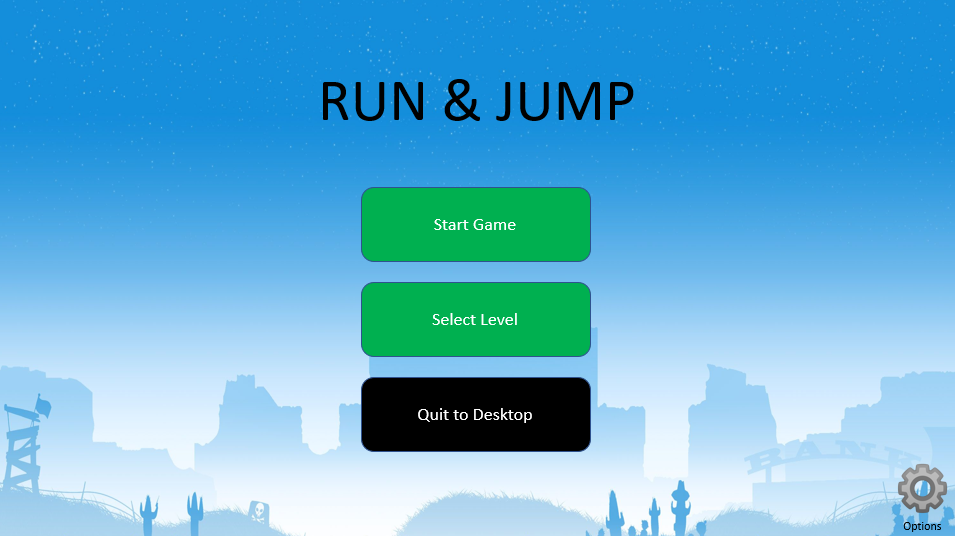


## post development test data



## Mock-Up

Main Menu Concept

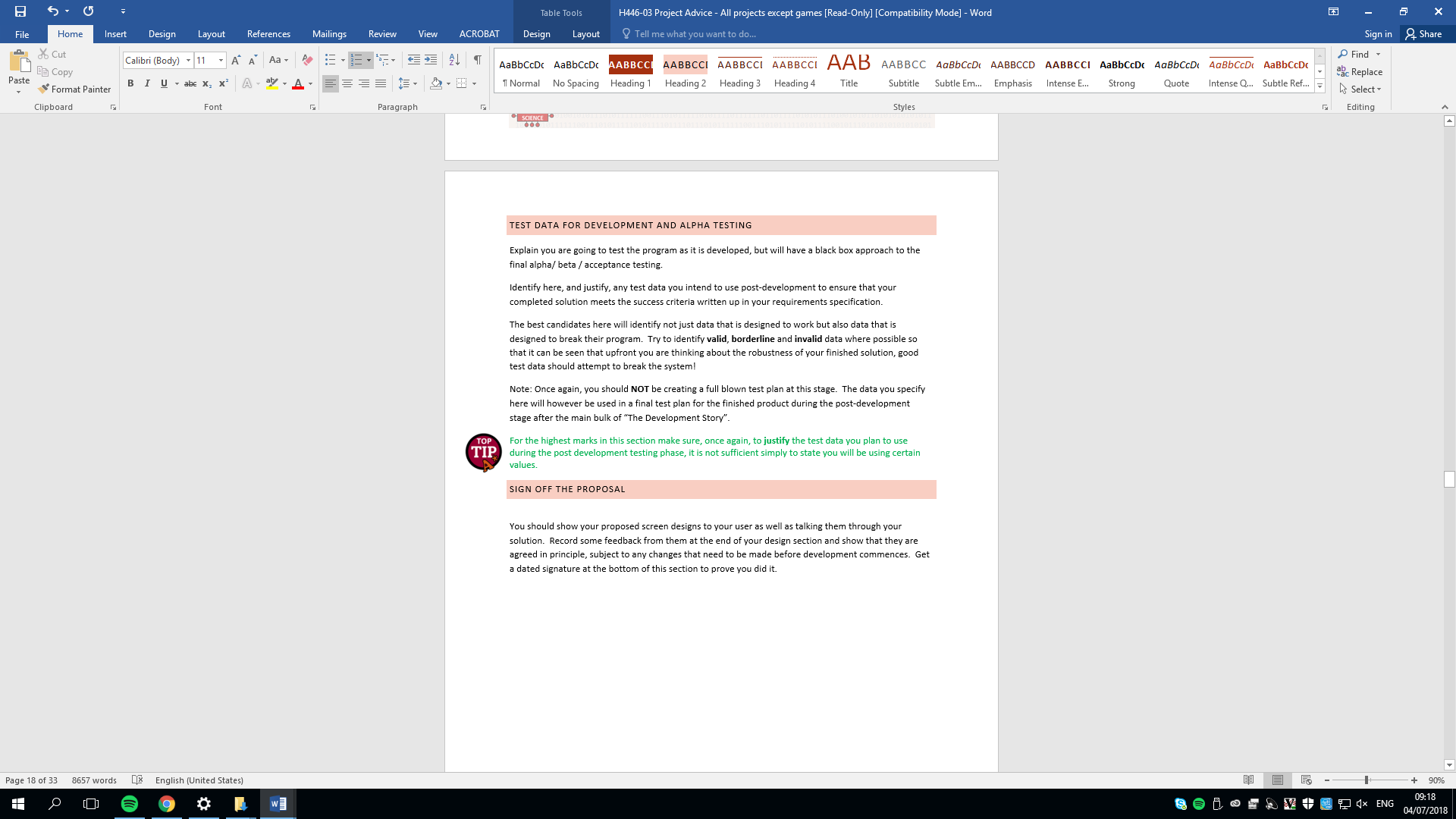


Main Game Concept

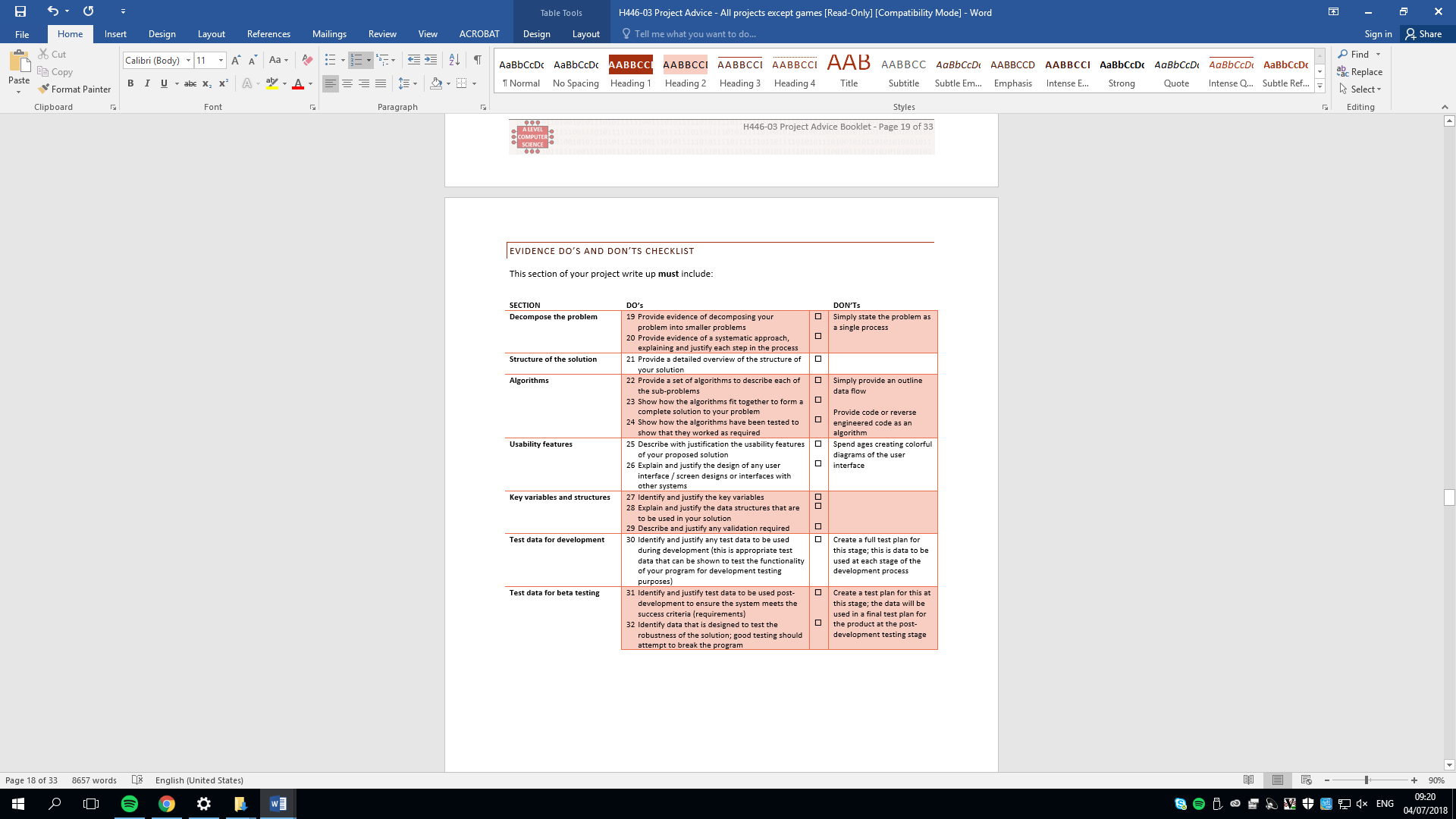
A screenshot of a computer

Description automatically generated with low confidence

## SIGN OFF PROPOSAL



checklist – DELETE WHEN CHECKED



# Developing the coded solution (“The development story”)

Prototyping

* The acc code

Code Explanation

* Explaining the code

Testing

* Screen shot or video of the outcome of the code
* Test if it works

Remedial action

* Fix code if there is problem
* Explain how you fixed the code

Feedback

* Ask stake holder how code could have been better

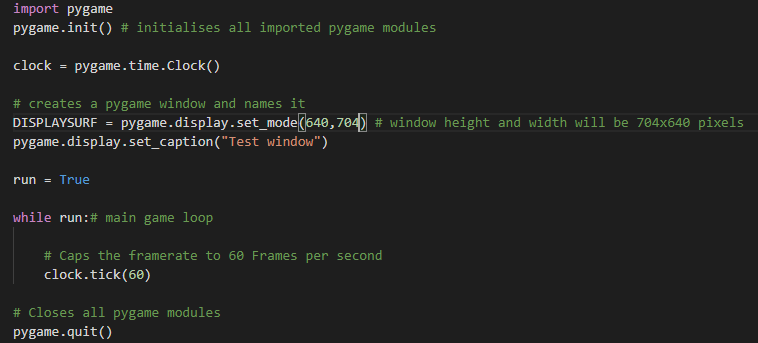
Final checklists delete

## Level generation 07/09/2022

### Prototype 1

#### Objective

My objective at the moment is to write a basic program that sets up an abstracted level that positions the platforms, player, and enemies on the screen. It will be used as a foundation so I can focus on the basic and essential features that make the game work. So in the future I can start adding more non-essential features like sound, HUD, pause menu etc.



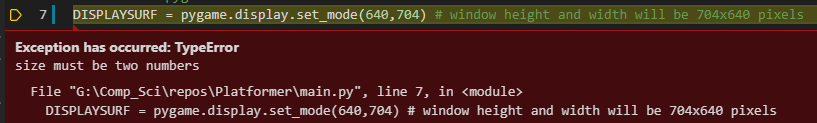
#### Explanation

This code should open a 640x704 pygame window named “Test window” and keeps it open using a while loop that loops forever. I specifically made the pygame window 640x704 big because I am planning on splitting up the whole window into a 20x22 grid.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 1 | This code should open a pygame window named “Test Window” and make it 640x704 pixels wide | * Open 640x704 pygame window * Name it “Test Window” | The code did not run and error was shown | Fail |

#### Error image



#### Remedial Action

Inside the pygame.display.set\_mode() function I was supposed to pass in a tuple containing 2 numbers not pass in 2 numbers on their own.

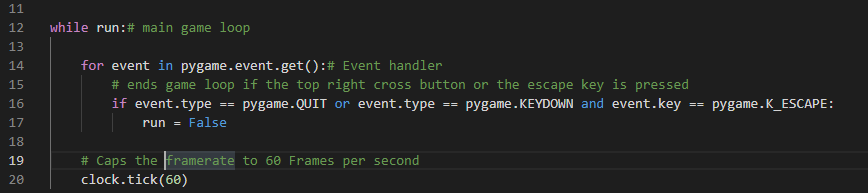
#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 1 | This code should open a pygame window named “Test Window” and make it 640x704 pixels wide | * Open 640x704 pygame window * Name it “Test Window” | The code ran and opened a pygame window named “Test Window” | Pass |

### Prototype 2

#### Objective

In this prototype I want to write something that allows the user to press the top right X button or the escape button to be able to close the pygame window.



#### Explanation

I added a function from the pygame library inside the main game loop called pygame.event.get() in a for loop that handles all events within the pygame window. Then inside the for loop I created an if condition that makes the run variable false if the user clicks the top right X button or the escape key is pressed. When the run variable is false the main game loop stops terminating all pygame modules which closes the pygame window.

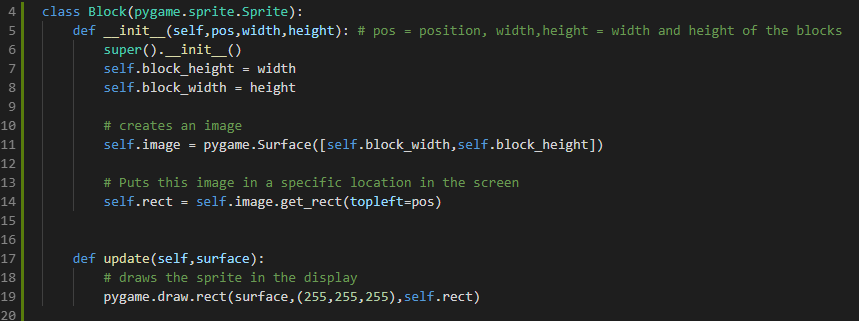
#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | Code allows the user to close the pygame window | * Close pygame window if esc key is pressed or the top right X button | Pygame window closed when escape key is pressed and when the top right button was clicked. | Pass |

### Prototype 3

#### Objective

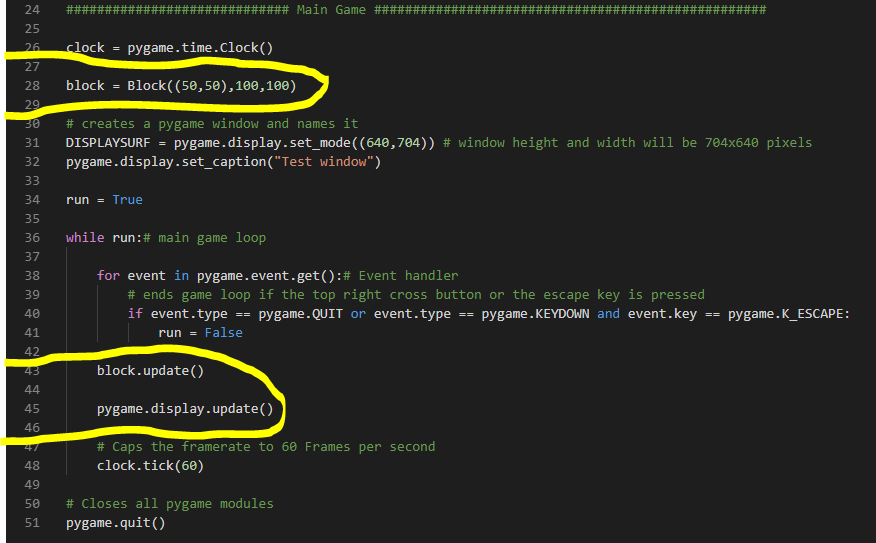
Now I have created a pygame window I am going to add some sprites like platforms players and enemies so I can set up and design the level.



#### Explanation

I created a block class within a class. I have done this because I want to keep separate sections of the code self-contained so in the future for instance if I want to add functionality or change a feature to the block all of that will be available to do inside the block class. Also, I have used the sprite module from the pygame library in my objects because it allows me to easily attach features like images, sound and rect (hitbox) to the object. At the moment, all the block class has is the most basic features like block height, width that will be filled with an image that drawn using the update() function.

\*\*New code circled in yellow\*\*



Within the main game loop, to use the block class I created an instance of a class and gave its height width and x/y position and saved it into a variable called block. Then inside the game loop I after every cycle I call the update function from the block and call a function that updates the pygame screen so we can see the things I have drawn.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 3 | This test is too see if the code I have added will draw the platform into the pygame screen with the given size and position | * White block 100x100 pixels big should be displayed at the position y:50 x:50 | The pygame window opened but the program terminated at line 43 and displayed an error | Fail |

Error Image

Text

Description automatically generated

#### Remedial Action

There was an error in the program because the function required a surface so pygame knows what surface to draw the sprite on. So, I will pass in the surface of the pygame window called DISPLAYSURF.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 3 | This test is to see if the code I have added will draw the platform into the pygame screen with the given size and position | * White block 100x100 pixels big should be displayed at the position y:50 x:50 | White block with the dimensions and position I gave was displayed on the pygame screen | Pass |

### Prototype 4

#### Objective

In this prototype I need to find a method to be able to draw multiple blocks preferably without giving each of them specific location manually as that will be very time consuming if a design multiple levels.

Text

Description automatically generated

Text

Description automatically generated

#### Explanation

In this code I created a game class that contains the level map and functions that sets up the level and draws all sprites to the pygame screen.

The map of the level is created using a 2D array. Each cell in the array represents a location in the screen and carries a number to indicate what object should be placed in that specific location.

To set up the level I created specific groups using the pygame.Sprite.Group() method for the sprites because it creates an instance of a class within a variable that allows me to store many sprites with in it and this class has a useful method called update() that calls the update function within all contained sprites which is useful since I am dealing with many block sprites. Also, pygame features a method that can detect when an object from a group collides with an object in another group which could be useful in the future to programme some physics. I then made a for loop that reads through each cell of the array, checks what number is in the cell if the number found is 2 it creates a block class and gives it a position, size and adds it to a block group.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 4 | This test is to test if the setUpLevel() function gives the block sprites the correct position according to the level map. | * The screen should have a similar pattern to the self.levelMap where there is a 2 a white block sprite should be in that position | * Where a 2 was displayed in the self.levelMap a white block was displayed | Pass |

A picture containing text

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 4.2 | I am going to change one of the 0 into 7 and check if that cell creates any errors or puts a white block in that position | * No changes to the level design or errors | * No changes to the level design or errors | Pass |

A picture containing text

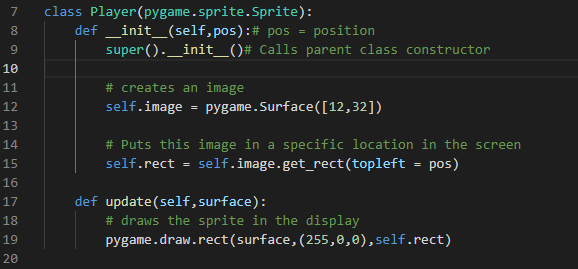
Description automatically generated

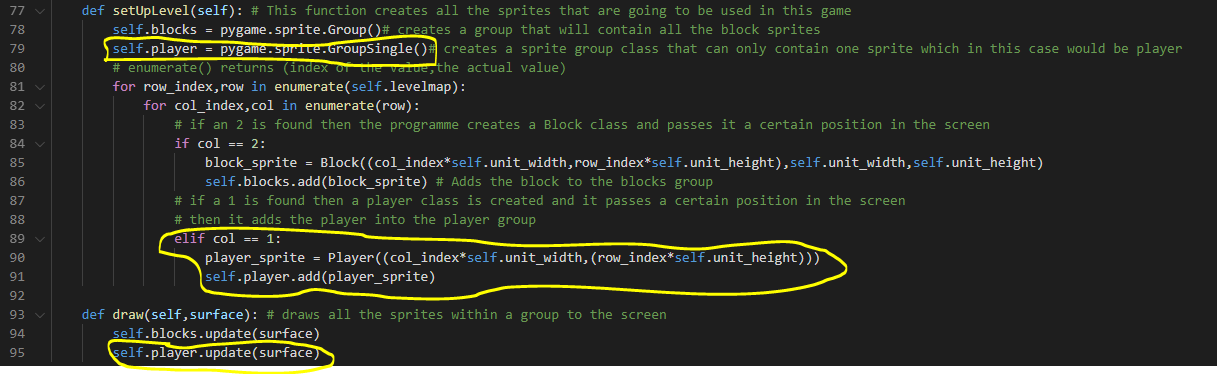
## Player movement 14/09/2022

### Prototype 1

#### Objective

Now I have got a simple level layout I will be adding an abstracted player



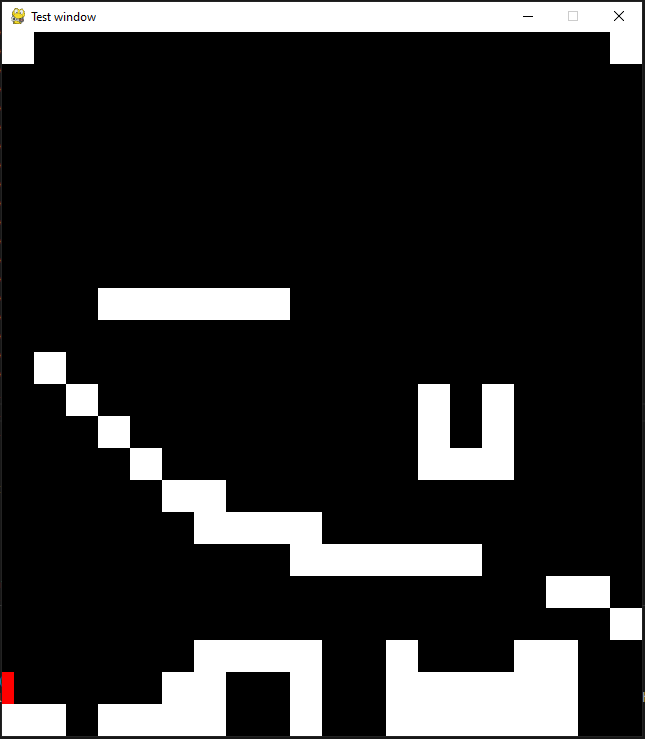


#### Description

I have created a player class that is RED and sized smaller than the blocks. Then within the setUpLLevel() method in the game class I create an instance of the class pass it a position according to the levelMap then add it to the player group. Then the player drawn using the player.update() method.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 1 | Test is to see if the player is displayed in the screen in the correct position in the screen | * Red rectangular player smaller than the platform blocks at the start position given by the map | * Red rectangular player smaller than the platform blocks at the start position given by the map | Pass |



### Prototype 2

#### Objective

Now I have a player in the screen I will add some code that allows the user to move the player using the left and right arrow keys.

Text

Description automatically generated

\*\* Inside game class \*\*

Text

Description automatically generated

Text

Description automatically generated

#### Description

In the player class I changed the code inside the update method because I found that there is a method inside the pygame sprite group class that draws all the sprites within the group. So removed the draw function inside the update method and replace it with code that allows the player sprite to move.

I also put all the current game play within a function called gameLoop() so in the future when I want to transition from the menu and the game loop all I need to do is call this function.

So in order to allow the user to move the player within the game loop function I added a pygame function called pygame.key.pressed() that gets all the Boolean values of all keys inside the keyboard. Then I passed this into a method within the game class called playerMove() that calls a method in the player class called update() that updates the position of the player inside the pygame window.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | This test will see if the player is able to move when the user uses the left and right arrow key | * If the left arrow key is pressed the player will move left * If the right arrow key is pressed the player will move right | * Pygame window opened but no sprites were displayed | Fail |

#### Remedial action

I reviewed my code and I think the issue is since I am using the draw function from the pygame group class for both the block and player instead of the pygame draw function I need to fill the image in when I create the block and player class. So I will use the self.image.fill(color) function to fill the image so the user can see the sprites.

This fills in the player image with the colour red.

Text

Description automatically generated

This fills in the block image with the colour white.

Text

Description automatically generated

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | This test will see if the player is able to move when the user uses the left and right arrow key | * If the left arrow key is pressed the player will move left * If the right arrow key is pressed the player will move right | * The sprites displayed * The player moved however when I held the arrow key the player only moved once   <https://youtu.be/VBgTTkX9UnE> | Fail |

#### Remedial action

I think the player only moved once because I only called the function to move the player inside the event handler for loop. So, the game did receive that I pressed the key down however after that the code didn’t register another event because I carried on holding the key down meaning the code didn’t enter the event handler for loop again resulting in the function to move the player not being triggered.

To resolve this problem, I will move the function to allow player movement out of the event handler for loop.

Text

Description automatically generated

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | This test will see if the player is able to move when the user uses the left and right arrow key | * If the left arrow key is pressed the player will move left * If the right arrow key is pressed the player will move right | * The player sprite moved constantly when I held the arrow keys * The last position of the player remains on the screen when the player moved   Video of this test:  <https://youtu.be/DOShBNXToJY> | Fail |

#### Remedial action

I believe the last position of the player remains on the screen when the player moved because the screen simply redraws the sprites on top of each other. So, I need to reset the screen and redraw the sprites. I will do this by using DISPLAYSURF.fill(color) function to draw all over the screen resetting the screen after every single loop in the game loop.

\*\*Inside game loop function\*\*

Text

Description automatically generated

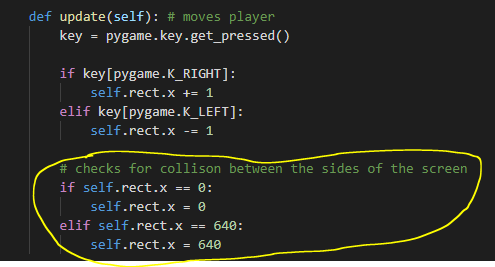
#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | This test will see if the player is able to move when the user uses the left and right arrow key | * If the left arrow key is pressed the player will move left * If the right arrow key is pressed the player will move right | * If the left arrow key is pressed the player will move left * If the right arrow key is pressed the player will move right   Video of Test:  <https://youtu.be/3PIj20AAffI> | Pass |

### Prototype 3

#### Objective

In this prototype I will be adding boundaries in the left and right sides of the screen so the player sprite can’t move off the sides of the screen.



Changes

* Set up level method in game class now returns sprites created in the function (move to block sprite)
* Deleted playerMove() method
* The update from the player class will be called in the main game loop to allow player movement now
* Update no longer needs a parameter to be passed in

#### Description

The code checks if the position of the player is bigger or smaller that the screen size. If the position is greater than the screen size the player position will be set to the max size of the screen. If the position is smaller than 0 the position of the player will be set to zero.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 3 | This test will see If the player is prevented from moving off screen | * If the player tries moves past the right edge of the screen the player will appear to stop moving * If the player tries moves past the left screen of the screen the player will appear to stop moving | * The player is stuck in the starting position and doesn’t move   Test Video:  <https://youtu.be/FlIAqpXlOfE> | Fail |

#### Remedial action

I reviewed the code and found that I forgot to call the update method for the player class in the main game loop to allow movement because I changed the way the player is moved in the code.

So to resolve this issue I will add the player.update method in the game loop to allow the player to move in the game loop function.

\*\* Inside game loop \*\*

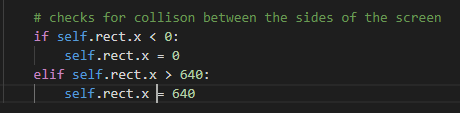


#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 3 | This test will see If the player is prevented from moving off screen | * If the player tries moves past the right edge of the screen the player will appear to stop moving * If the player tries moves past the left screen of the screen the player will appear to stop moving | * The player still moves off screen   Test Video:  <https://youtu.be/9uBDGO7f3T8> | Fail |

#### Remedial Action

I made it so if the position is smaller than 0 the player position will be also 0 so for instance if the player position is -3 for some reason the player position will still be set to 0.



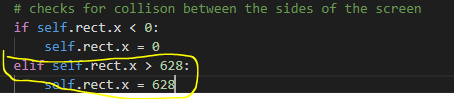
#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 3 | This test will see If the player is prevented from moving off screen | * If the player tries moves past the right edge of the screen the player will appear to stop moving * If the player tries moves past the left screen of the screen the player will appear to stop moving | * The player sprite didn’t move off the screen in the left-hand side * The player moves of the screen in the right-hand side   <https://youtu.be/eTXUftCCNE8> | Fail |

#### Remedial Action

I think the reason it moves off the screen in the right-hand side not the left hand-side because the position is determined by the top left edge of the player sprite so if the top left edge is the size of the full screen the right side of the sprite will still move off screen

So to resolve this problem I can take away the width of the sprite length of the max width the sprite can travel to.



#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 3 | This test will see If the player is prevented from moving off screen | * If the player tries moves past the right edge of the screen the player will appear to stop moving * If the player tries moves past the left screen of the screen the player will appear to stop moving | * If the player tries moves past the right edge of the screen the player will appear to stop moving * If the player tries moves past the left screen of the screen the player will appear to stop moving   <https://youtu.be/rt6ZmmzBASY> | Pass |

### Prototype 4

#### Objective

In this prototype I will add a jumping mechanic to the player sprite.

Text

Description automatically generated

Changes

I added 2 new attributes to the player class called speedX and speedY these 2 attributes will dictate how fast the player class can move in the X direction and the Y direction. I done this in case of the future if I need to change the player speed for some reason, I only have to change the values in the attributes. Also, it made programming this jump mechanic easier

#### Description

To make the jump mechanic work I created an attribute called jumping that is set to False. If the spacebar key is pressed and the jumping attribute is false it makes this jumping attribute True, this is done so the player can only jump when a full jump cycle is completed. Also speedY is set to 10, this indicates the initial jumping speed. After because the jump cycle is initiated, the speedY causes the player to move upwards. However after each cycle speedY decremented by an attribute I created called gravity (which holds the number 1 for now) causing the player to slow down while moving up, stop then start accelerating downwards. To allow the player to jump again I made an if statement that states if the speed is lower than -10 and jumping is True end the jump cycle.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 4 | This test will see If the player jump mechanic works | * If the user presses spacebar the player should jump up stop then return down * User should be able to jump again after the player moves down a certain speed | * When user presses the spacebar the player launches upwards and doesn’t come back down   Video of Test:  <https://youtu.be/tuQyS1oSULI> | Fail |

#### Remedial Action

The problem is the gravity shouldn’t be added to the x position of the player it should be added to the player Y speed.

Text

Description automatically generated

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 4 | This test will see If the player jump mechanic works | * If the user presses spacebar the player should jump up stop then return down * User should be able to jump again after the player moves down a certain speed | * When user presses the spacebar the player launches up but now even faster   Video of Test:  <https://youtu.be/VfKRf3mAPCc> | Fail |

#### Remedial Action

I reviewed the code and I think the problem is that I added gravity into speedY instead of subtracting gravity from speedY. So this change to the code should solve the problem.

Graphical user interface

Description automatically generated

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 4 | This test will see If the player jump mechanic works | * If the user presses spacebar the player should jump up stop then return down * User should be able to jump again after the player moves down a certain speed | * When user presses the spacebar, the player jumps correctly * The player is not able to jump again   Video of Test:  <https://youtu.be/9zFGXA7EBMc> | Fail |

#### Remedial Action

I reviewed the code again and found the issue. The if statement, where the program checks if the jump cycle should be ended, speedY should have been checked not speedX.

This change to the code should resolved the issue.

Graphical user interface, text, application

Description automatically generated

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 4 | This test will see If the player jump mechanic works | * If the user presses spacebar the player should jump up stop then return down * User should be able to jump again after the player moves down a certain speed | * When the user presses spacebar the player jumps up stop then return down * Able to jump again after the player moves down at a certain speed   Video of Test:  <https://youtu.be/ILbJ2rUQduk> | Pass |

## Player collision 12/10/2022

### Prototype 1

#### Objective

In this prototype I will add gravity to the player so it constantly falls when the game runs unless the player jumps



#### Description

Instead of only adding gravity once the player jumps I apply gravity everytime I call the update method so it always applies gravity.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 1 | This test will see If gravity is applied at all times unless the player jumps | * Once the game runs player should be falling down * While the player falls down if the spacebar is pressed the player jumps | * When the game starts the player falls * When the spacebar is pressed the player jumps   Video of Test:  <https://youtu.be/gYUn8FEqtyU> | Pass |

### Prototype 2

#### Objective

I will be adding y-axis collision system for the player.

\*\* inside gameloop function\*\*

Text

Description automatically generated

#### Description

The code loops through every block with in the blockgroup group. Then for each block the colliderect() function checks if the block has collided with the player and returns true if so. If the y axis speed of the player is positive that means the player is travelling downward so I set the position of the players bottom as the position of the block, the player collided with, bottom. Vice versa for if y axis speed of player is negative.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | This test will see if the player is able to stand on top of a block | * The player doesn’t fall through the block sprites * Player can move left and right on top of any block * Player can jump when on a block | * The player stops on top of the block but after a while it just falls through it   Video of Test:  <https://youtu.be/4mYW_NSJ3LU> | Fail |

#### Remedial Action

The problem is that the y axis speed still increases even when the player collides so eventually the player position updates to a position that is lower than the block. So to resolve this I need to reset the players y speed when he collides with the block.

Text

Description automatically generated

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | This test will see if the player is able to stand on top of a block | * The player doesn’t fall through the block sprites * Player can move left and right on top of any block * Player can jump when on a block | * The player doesn’t fall through the block * Player can move left and right on top of any block * Player can only jump once | Fail |

#### Remedial Action

I need to add in a jump reset function so when the player collides with a block it can jump again

Text

Description automatically generated

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | This test will see if the player is able to stand on top of a block | * The player doesn’t fall through the block sprites * Player can move left and right on top of any block * Player can jump when on a block | * The player doesn’t fall through the block * Player can move left and right on top of any block * The player can stay on the bottom of a block if the spacebar is held   Video of Test:  <https://youtu.be/-lC82rQW7Y4> | Fail |

#### Remedial Action

Only call the jumpReset() function when the player collides with the top of the blocks

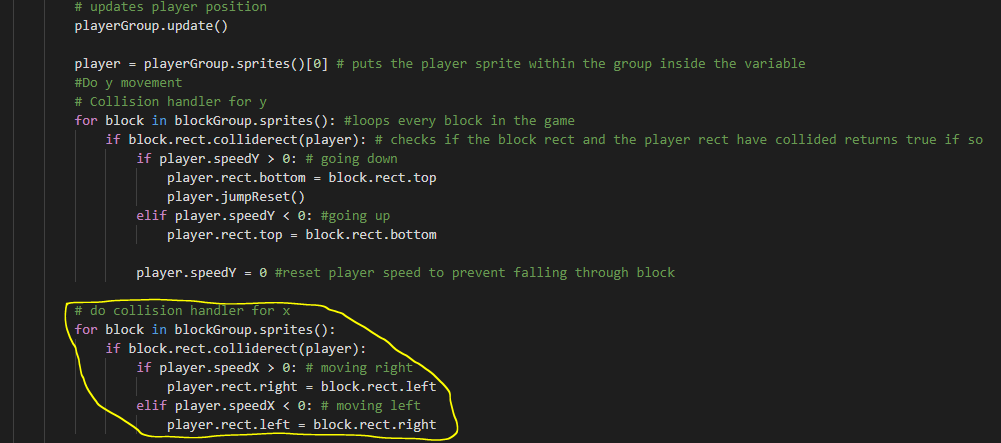
#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | This test will see if the player is able to stand on top of a block | * The player doesn’t fall through the block sprites * Player can move left and right on top of any block * Player can jump when on a block | * The player doesn’t fall through the block sprites * Player can move left and right on top of any block * Player can jump when on a block   Video of Test:  <https://youtu.be/Qohstwkbphs> | Pass |

### Prototype 3

#### Objective

To develop the collision system of the player in the x axis



#### Description

The code now searches through every block now however when a collision is detected the code also checks if an x axis collision has been made by checking if the speedX was positive meaning its moving right. Also if the player is moving right and a collision is detected the right position will be set as the left position of the block the player collided with. Vice versa for the left

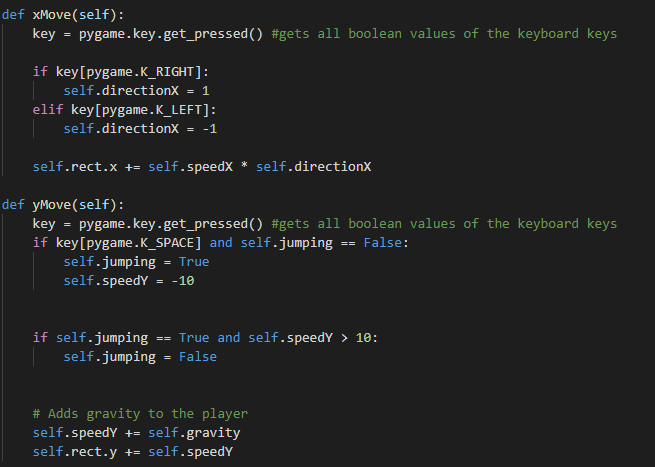
#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 3 | This test will see if the player stops moving when the player reaches a side of a block | * The player stops moving when the user tries move through a block in the x axis | * Nothing changed and the game still runs like prototype 3 | Fail |

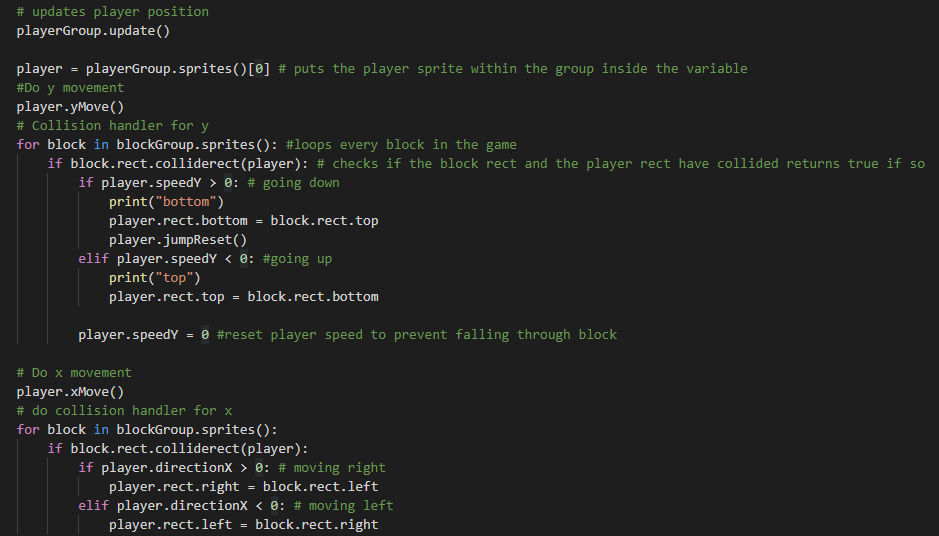
#### Remedial action

I reviewed the code and I found that the issue is I need to split the player movement y and x movement in different functions. This is because if a movement in the x axis caused a collision it will always treat it as a y collision because I check for y collision first before x axis collisions.

So in the player class I need to create 2 new functions that handle player x axis movement then player y axis movement.



Also I need to make the player move in the y axis check for a collision then move the player in the x axis then check for a collision.



#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 3 | This test will see if the player stops moving when the player reaches a side of a block | * The player stops moving when the user tries move through a block in the x axis | * Player isn’t able to move through any block   Video of Test:  <https://youtu.be/khUjqdMrcZk> | Pass |

### Stakeholder Feedback: Abdullah

Abdullah has played my early version of the game.

Things he likes:

* The blocks are structured well
* You can tell where the platform is
* The player jump mechaninc is good
* The player can also move while jumping
* The gravity works well

Things he thinks I should change:

* The player should not be able jump again until he touches the floor again
* The player should move faster so it is easier to jump across a gap

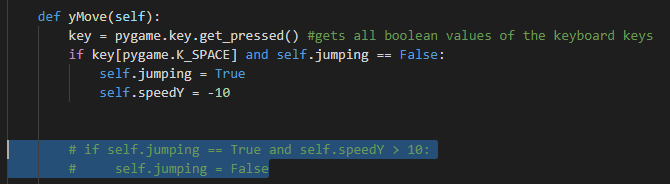
### Prototype 4

#### Objective

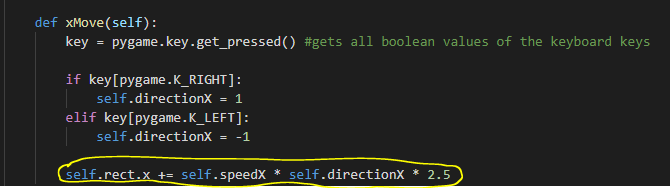
In this prototype I will be adding and amending features my stakeholder suggested which are

* The player should not be able jump again until he touches the floor again
* The player should move faster so it is easier to jump across a gap

#### Description



To make the player only be able to jump again after touching the floor I removed the code that allows the player to jump again after its going a certain speed downwards.





Also to allow the player to be able to jump across gaps easier I made the x direction speed 2.5 times faster than original and I made the initial jump speed of the player 2 values faster.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 4  Stakeholder suggestions | This test will see if the player is able to jump across gaps easier and the player can only jump after touching the floor. | * The player can jump across gaps easier * The player is only allowed to jump again after touching the platforms | * The player can jump across gaps easier * The player only jumps again after touching the platforms top | Pass |

## Player Animation 09/11/2022

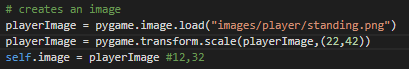
### Prototype 1

#### Objective

In this prototype I will be adding an image to represent the player.

#### Description

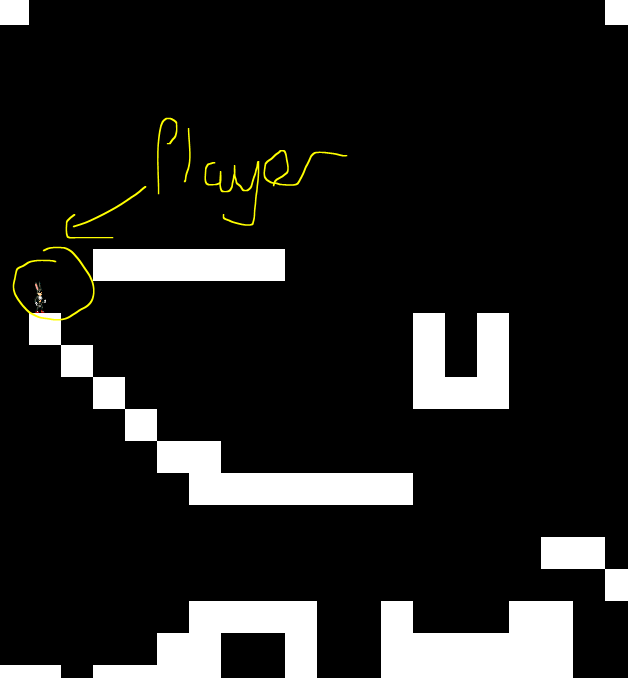
\*inside player class\*



The code loads a player image that I downloaded from the internet then changes it size so it 22 pixels tall and 32 pixels wide. Then it stores the image in self.image so when the game runs the code displays the image given rather that a rectangle filled with red.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 1 | This Test is to see if player is displayed as the image named standing.png from the player images | * The player is displayed the image standing.png * The player is 22 pixels tall and 32 pixels wide. | * The player is displayed the image standing.png * The player is 22 pixels tall and 32 pixels wide. | Pass |

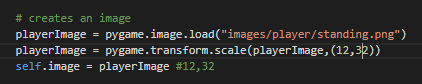


### StakeHolder Feedback

Stakeholder Jacy has tested the game again and has found that is hard to see the player sprite because it is small.

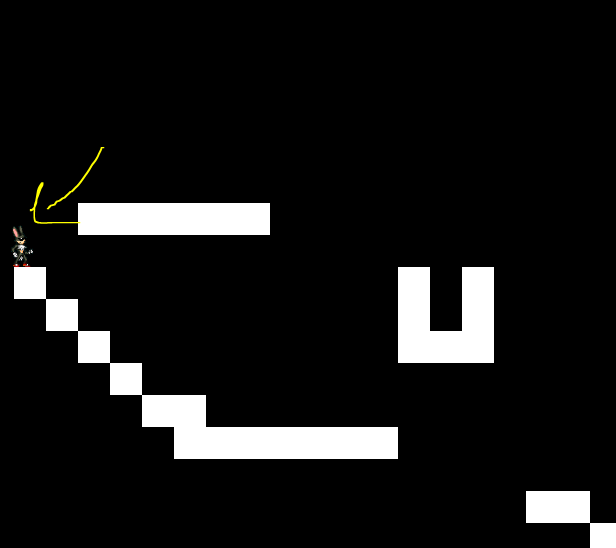
So because of this I will be increasing the size of the player

#### Remedial Action



In the code I have increased the player width and height by 10px

Now it looks like this



### Prototype 2

#### Objective

In this prototype I will add an idle animation to the player

#### Description

\*inside player class\*

Text

Description automatically generated

\*inside update method of the player class\*

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

In the loadPlayerImages method I put all the idle animation images within a list so they can be accessed sequentialy.

The animation function is called every time the update method of the player is called. In the animation function if the state of the player is idle it then increments animation counter by 1 which is an index to then set the image of the player as the corresponding value of that index in the idle image list.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | This test is to see If the player idle animation works | * The player should scroll through the idle animation list constantly | * The player moves through the blocks * The idle animation is too fast to see the animation happening   Video of Test: <https://youtu.be/UbNnOEF8jeI> | Fail |

#### Remedial Action

When looking for issue in the code I found that the rect was smaller than the actual image.

\*rect oulined in red\*

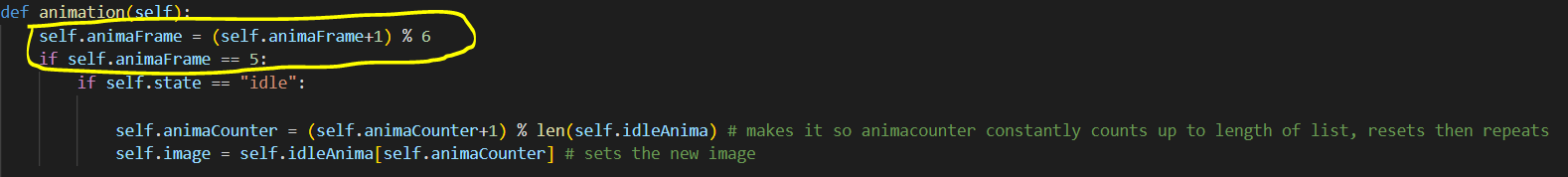
Icon

Description automatically generated

So when the rect stands ontop of the platform the player image would fall through the plaform as its taller that the rect.

To resolve this issue I will resize all the loaded images to the rect size in the method that loads all images.

Also to slow down the speed the code flicks through the idle images I added some sort of cool down so only after every 5 frames the code flicks to the next image.



#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | This test is to see If the player idle animation works | * The player should scroll through the idle animation list constantly | * The player scrolled through the idle animation list constantly   Video of Test: <https://youtu.be/0M1LHfo0nH4> | Pass |

### Prototype 3

#### Objective

In this prototype I will add a running animation to the player.

#### Description

A picture containing diagram

Description automatically generated

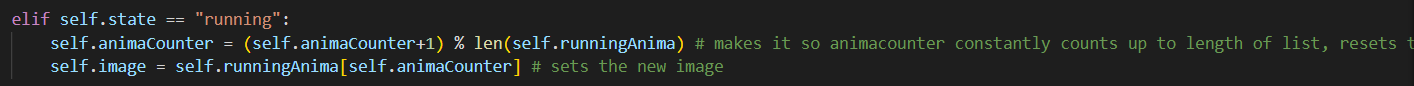
If the right arrow key or the left arrow key is pressed the state of the player is “running” else it is “idle”

\*inside the load images \*

Text

Description automatically generated

\*inside animation procedure\*



Both the same as prototype 2 explanation

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 3 | This test is to see If the player running animation works | * The player should scroll through the running animation list constantly when the player is running | * The player scrolls through the running images when the player is running | Pass |

### Prototype 4

#### Objective

I will be adding a jumping and falling animation to player

#### Description

\*Inside load images procedure\*

Graphical user interface, text

Description automatically generated

Explanation same as prototype 2

Text

Description automatically generated

First of all the code sets the state of the player to “jumping” if the player is currently jumping. Then the code loops through the first 6 images inside jumping Anima list and every time it does this it replaces the player image with the new image. When it reaches the sixth image and the player yspeed is bigger than one meaning its falling the player image is replaced with a falling image. When the player lands an idle animation is played or a running animation Is played depending on what key is currently being pressed

I put the jumping animation outside the (if self.animaFrame == 5:) if statement becausei want the jump animation to be faster than the other animations.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 4 | This test is to see If the player jumping and falling animation works | * The player should have a jump animation as its travelling upwards from a jump * The player should be going through a falling animation when it falls after a jump | * Player does a jump animation when going upwards in a jump * Player does a falling animation when falling during a jump sequence   Video of Test: <https://youtu.be/-oH7HUr-14g> | Pass |

### Prototype 5

#### Objective

In this prototype im going to make the player face the direction its moving towards.

#### Description

\*\* inside player animation method \*\*

Text

Description automatically generated

I created an attribute called self.facing in the player class and this code changes the value in it between left or right depending on If the player has change direction of travel.

Text

Description automatically generated

Then in the animation method I flip the image horizontally everytime a new image is set if the player is facing left since all the original images face towards the right.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 5 | This test checks if the image flips towards the direction of the player movement | * The player image flips towards the last direction of movement | * Player image flips towards direction of movement | Pass |

### StakeHolder Feedback

Stakeholder abdullah found a bug when testing the current game out. The bug is that when the jump button is held down the player image gets stuck in its falling state.

**Video of Bug**

<https://youtu.be/UacWN02vW8k>

#### Remedial Action

The bug was caused because animation counter wasn’t reset to zero when jump sequence animation was initiated.

This change in the code should fix the bug

Text

Description automatically generated

This resets the animation counter everytime the jump sequence is initiated.

## Platform Design

### Prototype 1

#### Objective

In this I will be changing the image of the platform.

#### Description

A screenshot of a computer

Description automatically generated with medium confidence

This code loads an image I downloaded from the internet and resizes it to the size the block is supposed to be then stores it in the self.image variable

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 1 | This test checks that the platforms images are replace to images for bricks. | * The platforms are now bricks | * Platform image is now a brick | Pass |

A picture containing timeline

Description automatically generated

## End Goal

### Prototype 1

#### Objective

In this I will create a sprite that will represent the end goal and which will also be positioned using the grid from the game map

#### Description

Text

Description automatically generated

This is just a simple boilerplate sprite. I set the color of it to blue and I put the size of it similar to the player size.

\*\* constructor method of game class \*\*

Graphical user interface, text

Description automatically generated

The 3 represents the endgoal. This will dictate where in the screen the end goal should be displayed

Text

Description automatically generated

In this I am creating the sprite and assigning it a positition depending on where in the level grid it is like I did to the other sprites. I add this sprite into a sprite group called goal.

Text

Description automatically generated with low confidence

To draw the sprite I place the draw method of the group into the draw method of the game class

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 1 | This test checks if the end goal is displayed in the right position on the level map | * The end goal should be displayed as a blue rectangle * The end goal should be displayed at the correct position in the level map | * The end goal is displayed as a blue rectangle * The end goal rectangle image enters the platform | Fail |

A picture containing timeline

Description automatically generated

#### Remedial Action

I failed to consider that the position of the end goal image will be decided from the top left point of the image and also the end goal is taller than the platforms. If I positioned the end goal the same way I did for the player part of the bottom of the end goal will overlap the platform.

To fix this I will minus some pixels from the y value I pass in to the end goal rect so the image is raised.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 1 | This test checks if the end goal is displayed in the right position on the level map | * The end goal should be displayed as a blue rectangle * The end goal should be displayed at the correct position in the level map | * The end goal is displayed as a blue rectangle * The end goal rectangle image doesn’t overlap the platforms anymore | Pass |

### Prototype 2

#### Objective

In this prototype I will use an image to represent the end goal and add some animation into it.

#### Description

Text

Description automatically generated

In this code I load the door images and I place them in a list similar to how I did in my player class. At first I set the image of the door as the first image in the list which is the closed door image.

Text

Description automatically generated

This is one of the methods I added into the Goal class. This method changes the image from its closed state to its opened state

\*\* inside section of gameloop that handles collisions \*\*

Text

Description automatically generated

This code triggers the switch method of the goal if the player collides with it

The purpose of this added code is too open the door when the player comes into contact with it.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | Test if the endGoal animation is working correctly | * The end Goal image should be a closed door when the player is not in contact with it * The end Goal image should be an opened door if the player has come in contact with it | * The door constantly opens and closes when the player comes in contact with it.   Video of Test: <https://youtu.be/b3Cqy6V4o-U> | Fail |

#### Remedial Action

To resolve this problem I added an attribute called self.state in my player class that will store the current state of the endGoal.

A screenshot of a computer

Description automatically generated with medium confidence

I also changed the way the switch method works in the player class.

A picture containing text, orange, dark

Description automatically generated

The switch method gets the triggered every time the player collides with the endGoal like before but this time it will not switch the image but it will only switch the state of the door to open

I then also created an animation method that deals with what image should be displayed depending on the state of the endGoal.

Text

Description automatically generated

This method is then called by the update method that will be called in the gameloop that triggers the animation method and resets the state to close incase the player isnt colliding with the endGoal anymore.

Text

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | Test if the endGoal animation is working correctly | * The end Goal image should be a closed door when the player is not in contact with it * The end Goal image should be an opened door if the player has come in contact with it | * The door opens when the player collides with it * The door closes when the player is not colliding with the endGoal   Video of Test:  <https://youtu.be/oynXATYB2rI> | Pass |

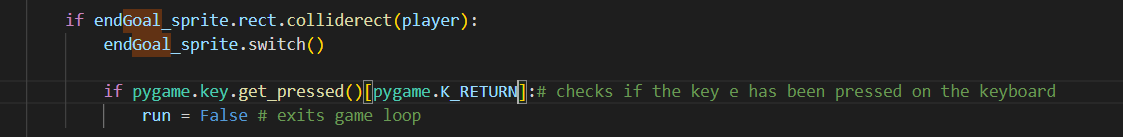
### Prototype 3

#### Objective

In this I will add functionality to the endGoal so when the player collides with it it ends the game.

#### Description

\*\* inside gameloop\*\*



The added code exits the game loop only when the player has collided with the endGoal sprite and the user presses the enter key.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | Test if the endGoal is able to end the game | * The end Goal terminates game when the player collides with the endGoal and presses the enter key | * The end Goal terminates the game when the player collides with the endGoal and presses the return key. | Pass |

## Dangerous Objects

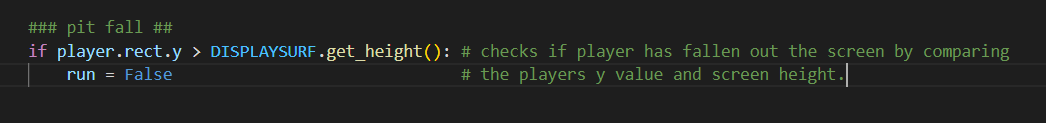
In this section I will be adding dangerous objects like enemies spikes and pitfalls that the player has to avoid.

### Prototype 1

#### objective

In this prototype I will be adding pitfalls that makes the player instantly die and end the game.

#### Description



This code is for the pitfall. It creates a pitfall that kills the player and ends the game. This is done by comparing the players y value and the screen height, if the y value is bigger than the screen height it ends the game loop.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 1 | The player dies when it has fallen of the screen | * The game ends if the player falls off the screen | * The game ends when the player falls of the screen | Pass |

### Prototype 2

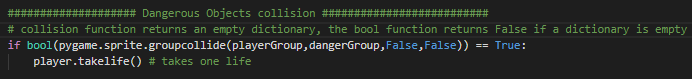
#### Objective

In this prototype I will be adding spikes to the game that tackes away 1 life if the player come into contact with it

#### Description









## Menu 09/11/2022

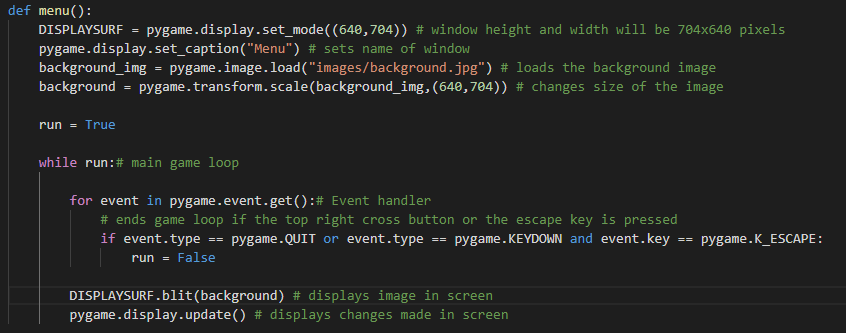
Since I have made most of the features that allow the game to run as it should at its most basic level I will be moving on to the menu which will take me less time to complete. I will be coming back to developing the main game later

### Prototype 1

#### Objective

In this prototype I will be putting up a background image for the menu

#### Description

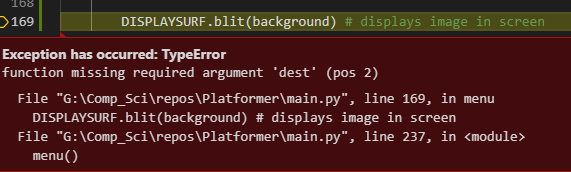


In the code created a new window for the menu and loaded the background image and changed its size to the size of the pygame window. Within the menu loop I blit the image to the screen and update the menu display.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 1 | This test is to check if the background image is displayed | * Program displays background image * Background image covers entire screen | * Error pops up | Fail |

Error image



#### Remedial Action

Blit function needs an additional argument which is the position where the image should be blitted.



#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 1 | This test is to check if the background image is displayed | * Program displays background image * Background image covers entire screen | * Program displays background image * Background image covers entire screen | Pass |

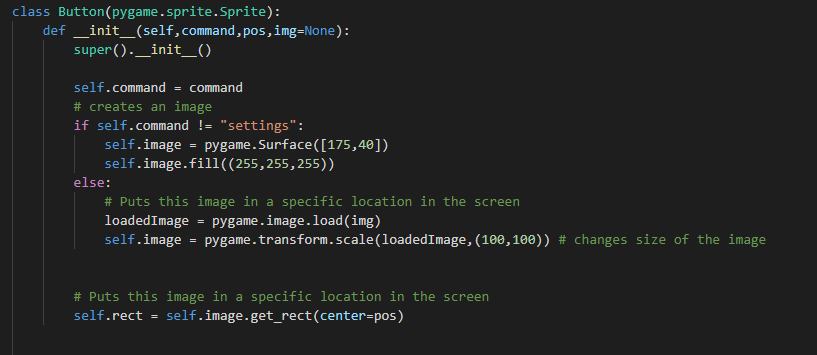


### Prototype 2

#### Objective

In this prototype I will be adding buttons into my menu

#### Description



The code creates a button class that has 3 arguments. Command is what the button does, pos determines the position of the button and img is the image location. If the command isnt settings an ordinary rectangular white box will be created a but if its any other command a gear icon will be loaded from the image folder and changed to a 100x100 pixel size image

Then in the main menu loop I create all the buttons and put them into a button group class that will be updated repeatedly to display them

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | This test is to see if all the buttons are displayed in the screen | * 3 buttons will be displayed in the middle of the screen * A gear Icon should be displayed at the bottom right of the screen | * The 3 buttons display on the screen alright * The bottom right gear icon goes off screen a little bit | Fail |



#### Remedial Action

I will reduce the x axis coordinate value for the gear icon



#### Test

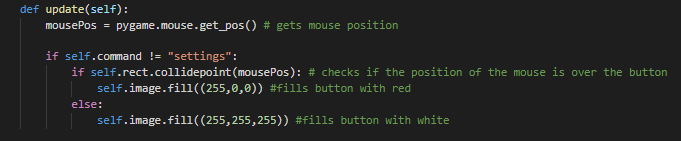
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | This test is to see if all the buttons are displayed in the screen | * 3 buttons will be displayed in the middle of the screen * A gear Icon should be displayed at the bottom right of the screen | * The 3 buttons display on the screen alright * The gear button is at the bottom right of the screen | Pass |

### Prototype 3

#### Objective

Add a feature where if the mouse hovers over the button the button turns red

#### Description



This function gets the mouse position if the button is not the settings button it then checks if the position of the mouse position collides with the button, if its true it changes the color to red else its white.

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 2 | This test is to see if when the mouse hovers over the buttons, the button turns red | * Buttons turn red when the mouse hovers on top of them | * Buttons turn red when the mouse hovers on top of them * Video of Test: <https://youtu.be/J6VEVL6x4PI> | Pass |

### Prototype 4

#### Objective

To add text into the boxes. For example I want the start button to have the words “Start Game” in it.

#### Description

\*inside buttion init method\*



This code gets the font “Arial” and then renders the text inside command and makes it that same font

\*Inside buttons update method\*



This blits the text inside the button surface in a specific location given in the surface

#### Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Num** | **Description** | **Expectation** | **Actual Result** | **Pass/Fail** |
| 3 | This test is to see if the command text is outputted inside the button | * Command text is outputted in the button | * Command text displayed | Pass |

\*Image of result\*



### Prototype 5

#### Objective

In this prototype I will be adding functionality in my menu. So when I click the start button it starts the game and does the appropriate things for the other buttons.

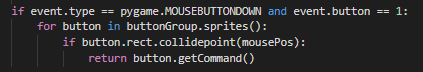
#### Description

\*In button class\*



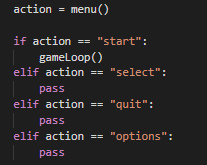
This code returns the command of the button

\*in menu loop function\*

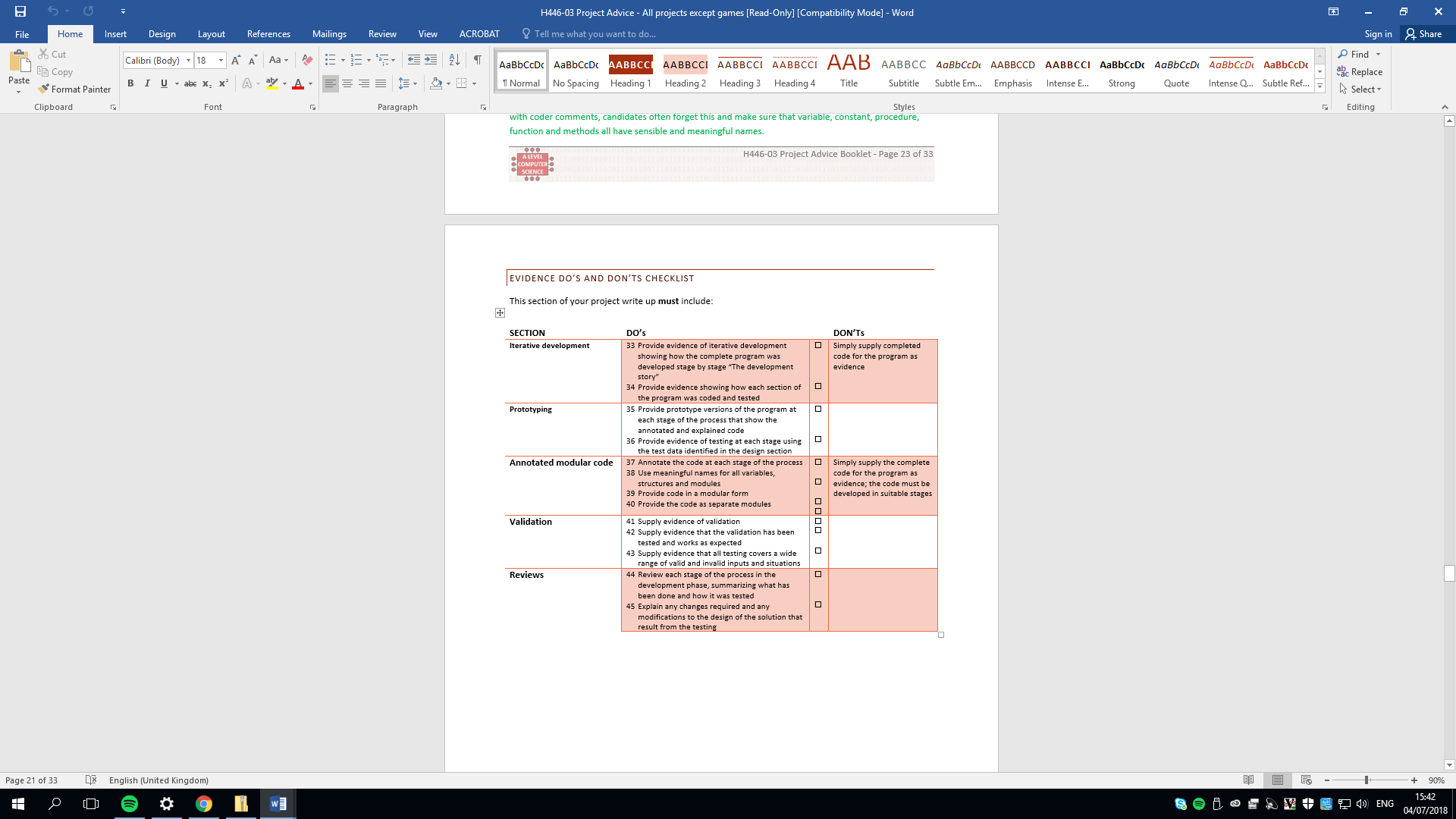


This code checks if the mouse right clicked ontop of one of the button. Then if it does the menu loop returns the command of the specific button clicked to end the function.

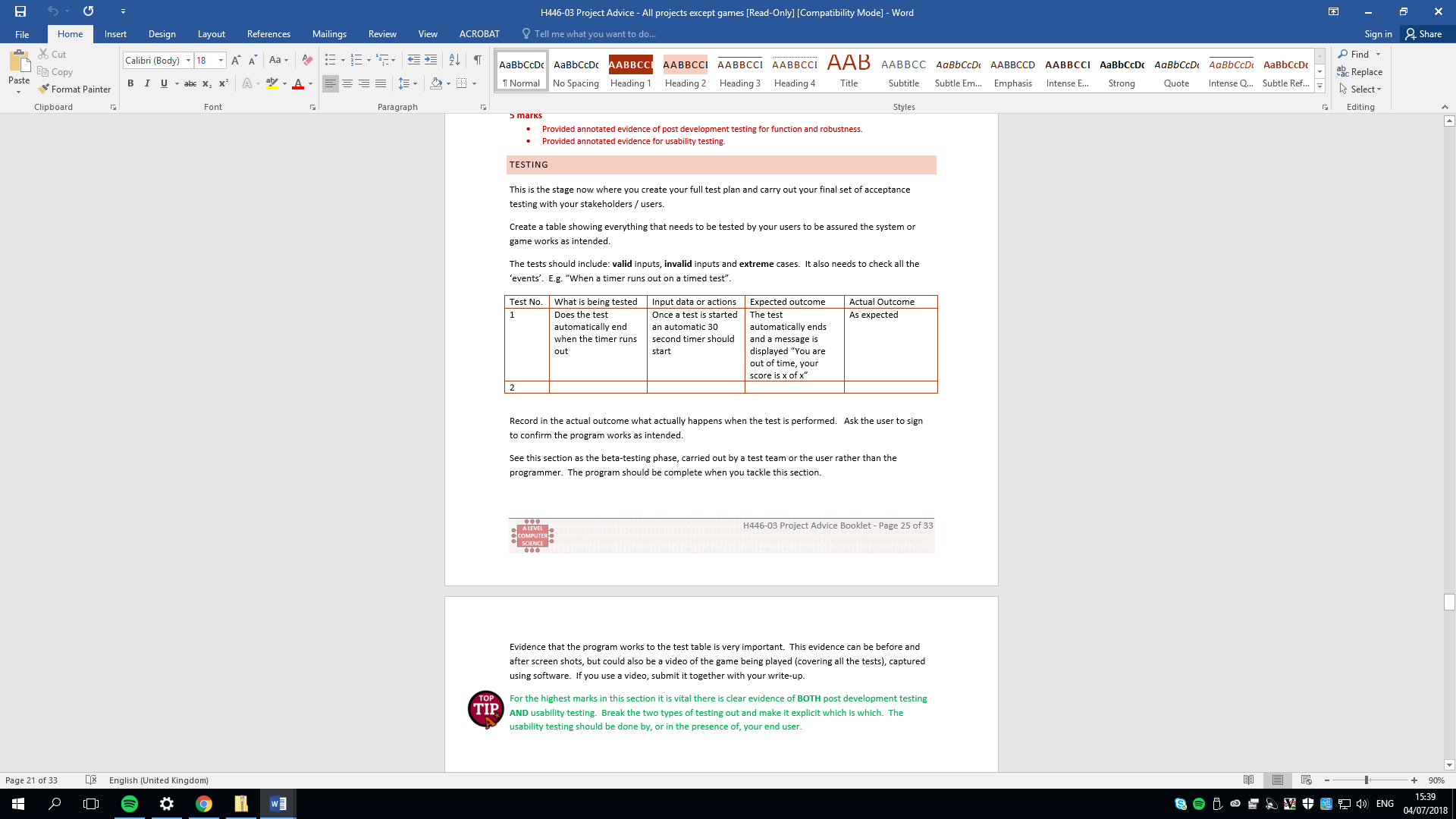
\*not in any function or class\*

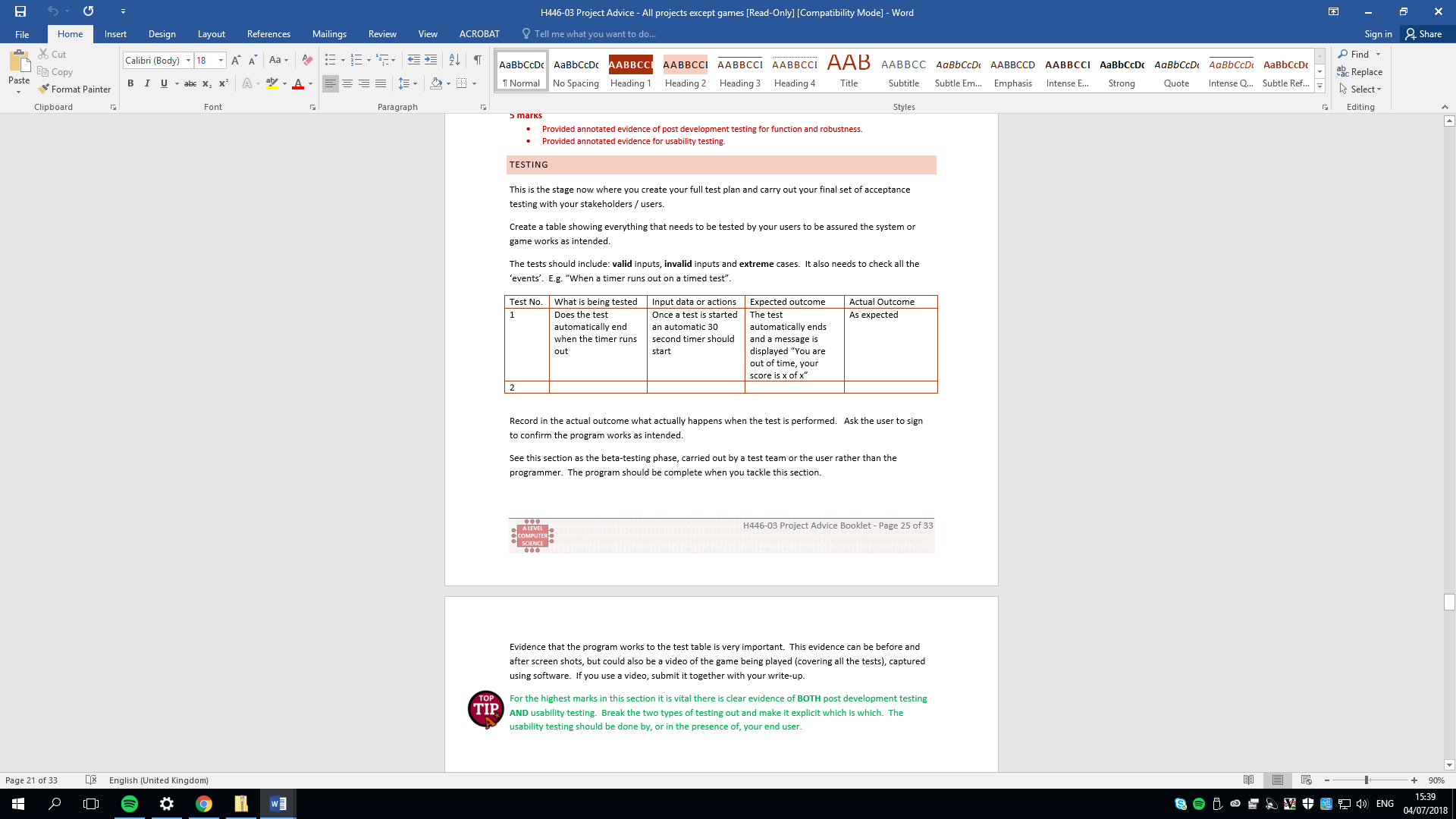


This code initiates the main menu loop when the code is runned and recieves the command of the button pressed and stores it in a variable called action. The if “start” was received ….. (finish of later)

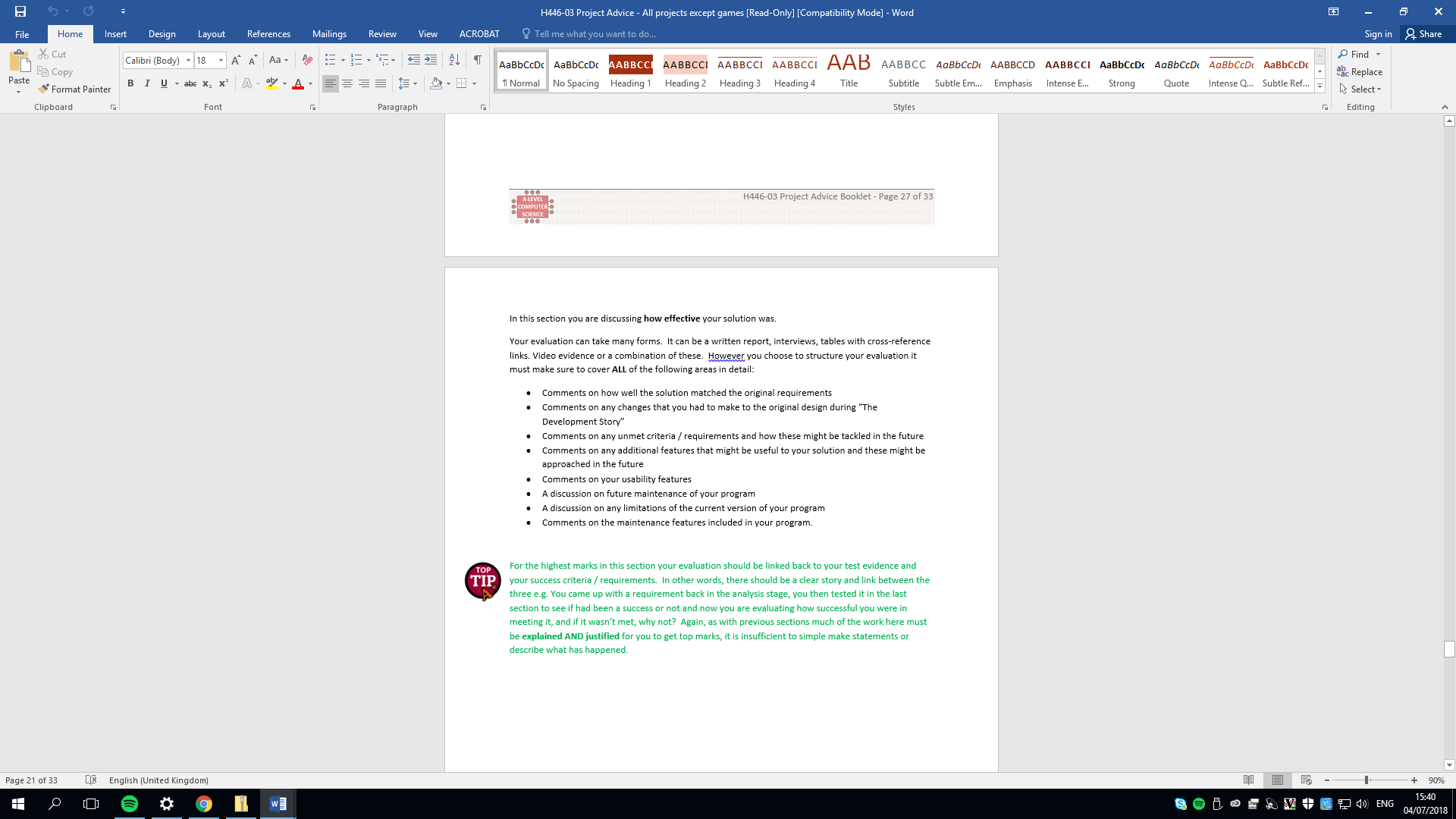


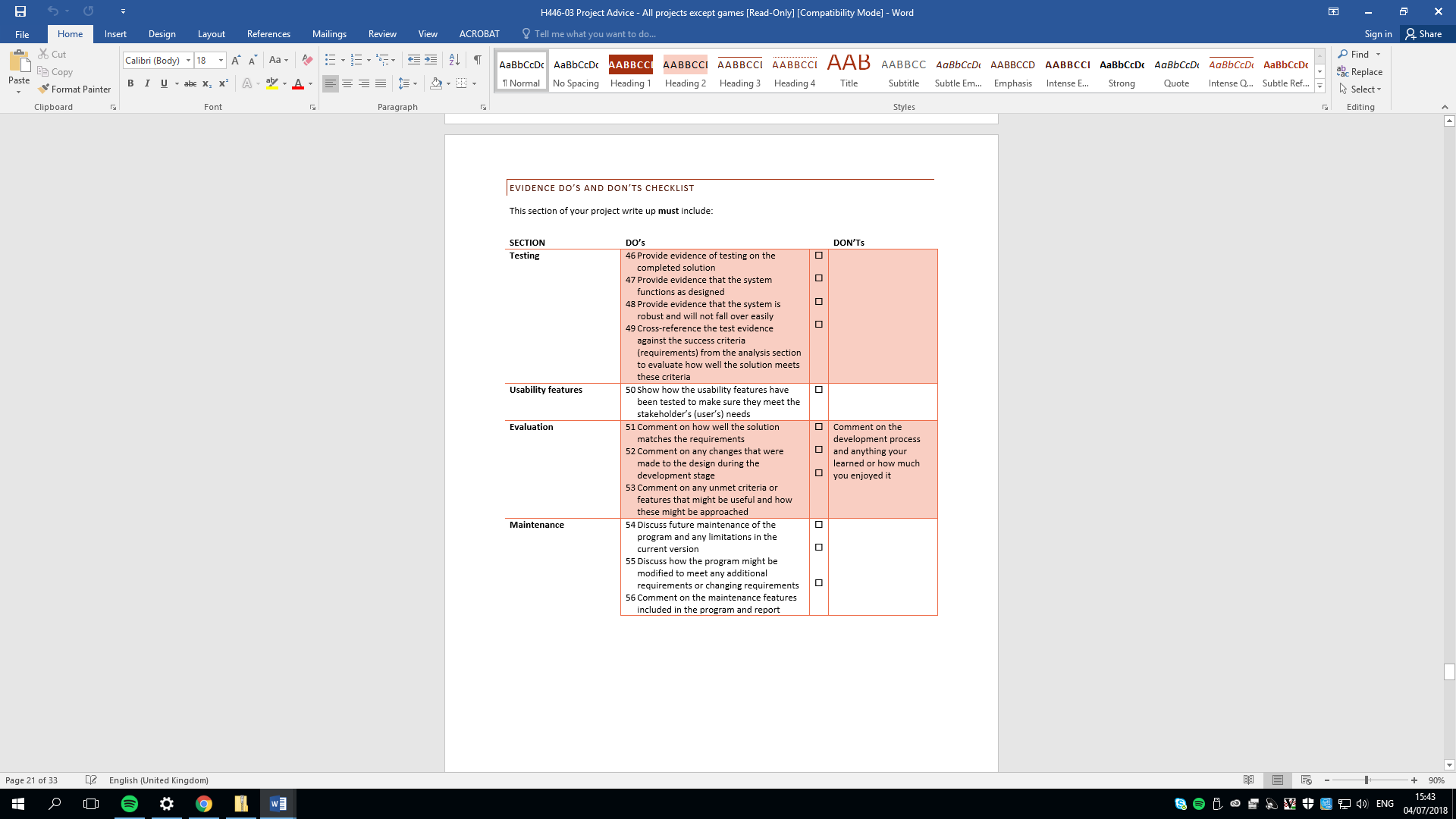
# Evaluation





## Evaluation-final solution





# Project Appendixes

Insert as many project appendixes as you need for your project.

These might include, but are not limited to:

* Complete Code Listing (ESSENTIAL)
* Interview Transcripts
* Meeting notes
* Observation notes or questionnaires