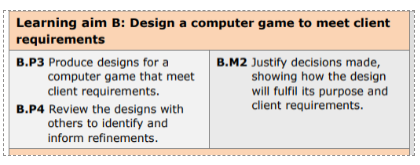
**Unit 14 designing document, Task 2**



**User requirement**

**Why are you making the game?**

I am developing a game because I have been given a job as a trainee programmer as a software developer who creates computer games. Creating this game will benefit me and challenge myself to new skills which I have not yet mastered.

**What is the game?**

I have been asked to design and construct a video game which will appeal to the age group of children ranging from 8-12. The game is a platformer game in which the user can control the character to complete various levels which get more difficult via progression. The platformer genre game insists of a 2D camera view and use of basic controls. There is also a help guide option which can be selected from the home page.

**What is the point of it and why?**

The aim of this game is to complete the levels with your lives and collect the diamonds throughout to increase your score. There is also a key on each level, the user must control the player to collect the key to progress past the door and enter the next level. The user playing must be able to avoid obstructions to complete levels and the game. There is also an enemy which moves slower than the character but will be programmed to follow the character.

**What are you trying to accomplish?**

I am trying to develop a game with various stages for the gamers to play. Game will have multiple stages which get trickier as levels pass. There are 2 gaming genres involved in this game, these include parkour and puzzles. In level 1 and level 3

**A description of the user and their needs?**

The gamer wants to be able to complete each stage without using up all their lives and make it to the end with a high score. Users need to have access to a computer desktop or a laptop. The reason for this is the game controls. Hopefully in the future I will be able to develop a version of the game available for mobile and other smart devices.

**The specific requirements of the user?**

This game does not really require any specific requirements. Gamers tend to sometimes skip out the how to play areas, however I will create an amazingly simple version and include images therefore, the target audience will easily be able to read it.

**Who is the Target audience?**

**Age**

I have decided to make my target audience base with the help of the assignment brief. This was an especially important aspect of this project. My game is very user friendly and does not involve anything above the age range. This game does not include any bad language or anything inappropriate for the age range. The decided age which my game is based on is 8-12. I will need to keep this in mind once I create my game and my user interface.

**Gender**

The game which I am creating can appeal to both genders and does not use any bias color ranges. A few ideas from my game comes from another immensely popular game which is also played even by the same age range as my target audience.

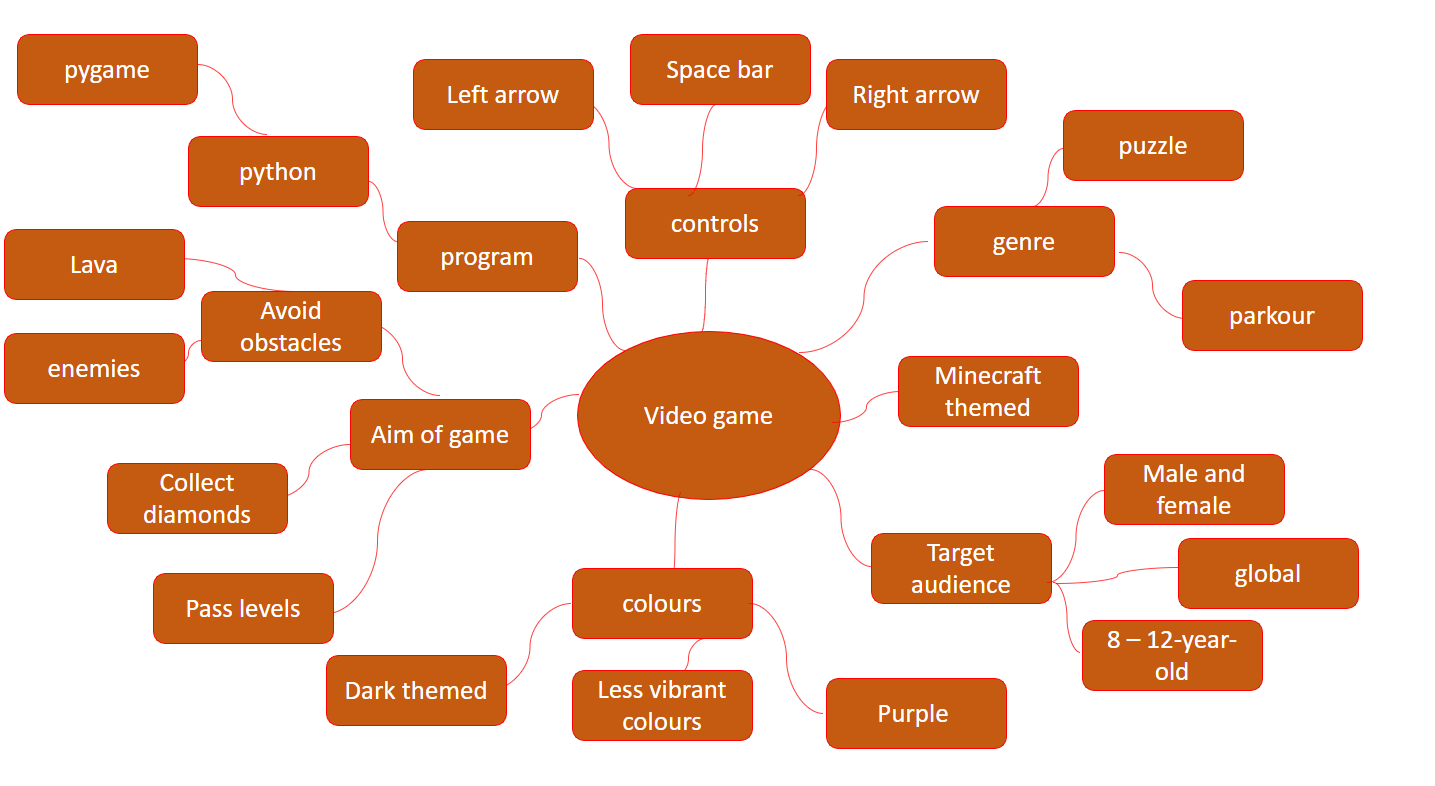
**interest**

This game does not require any interest in any other games. This game has 2 genres. I did this to attract a larger audience. For example, I decided to make just a parkour game, however as I looked over, I realized 8–12-year-olds may not all be interested in parkour games, and some may struggle if it is made too hard. Therefor I decided to add a maze which falls under the genre of puzzles. I have decided to not make the maze too hard as there will also be an entity following the player so it may make it too challenging.

**Location**

This game will be accessible globally, therefore there are no location restrictions. If you have access to a computer or laptop, you can gain access to the game. I have decided not to use a specific location for the release of the game, the reason for this is to allow more traffic towards the game I have created.

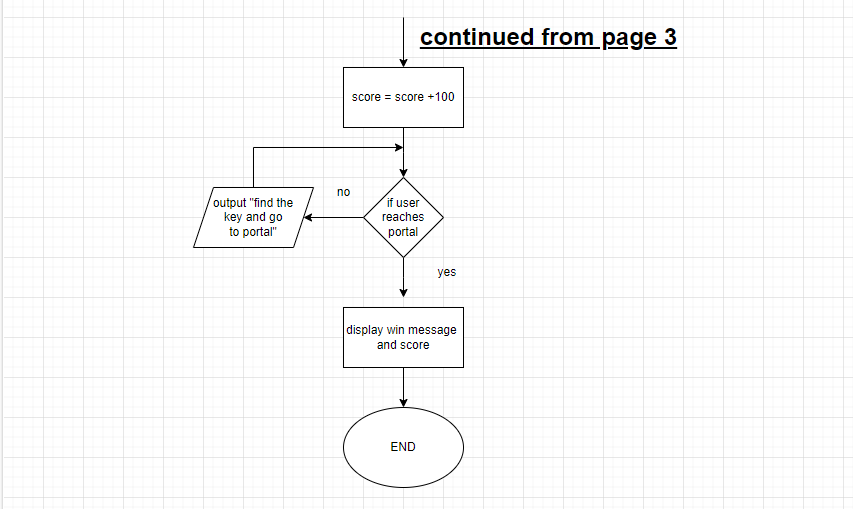
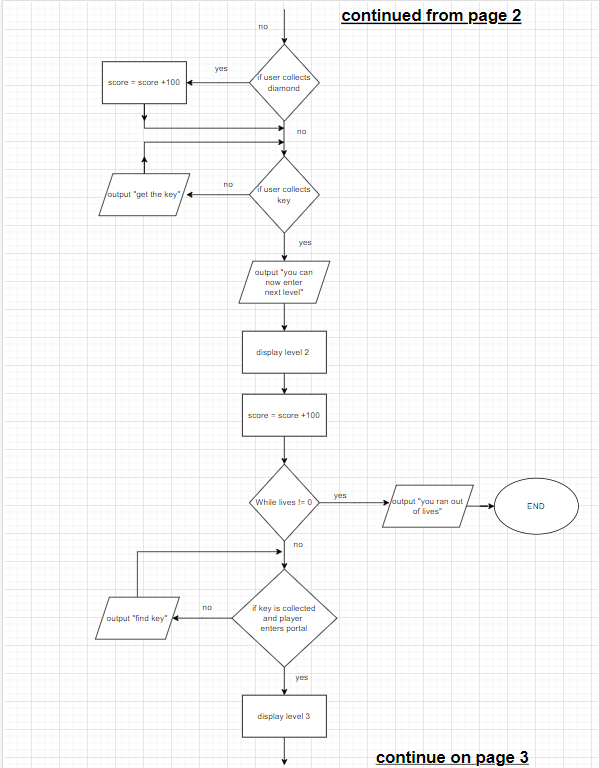
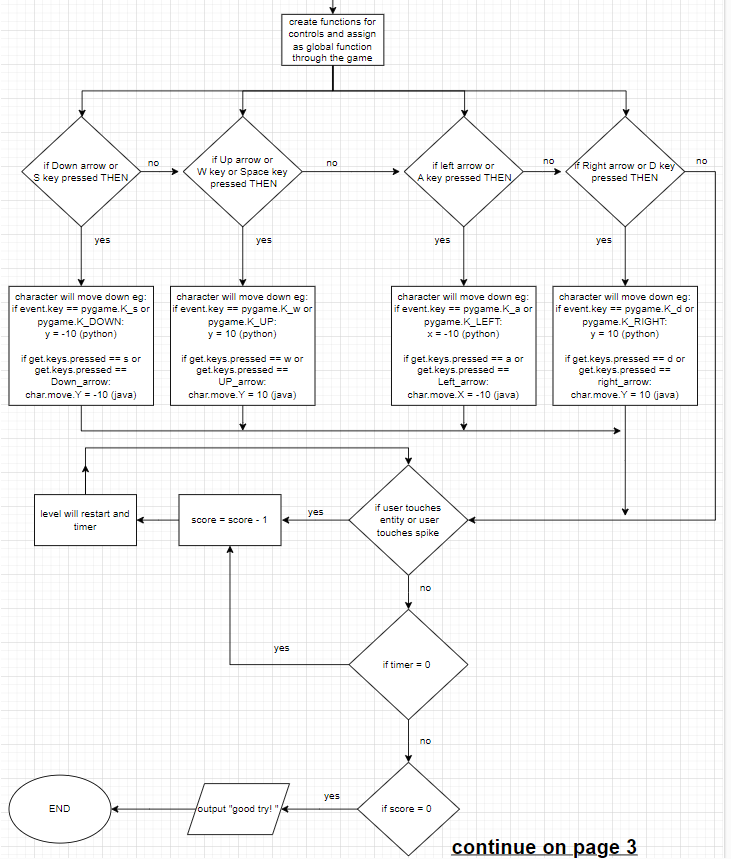
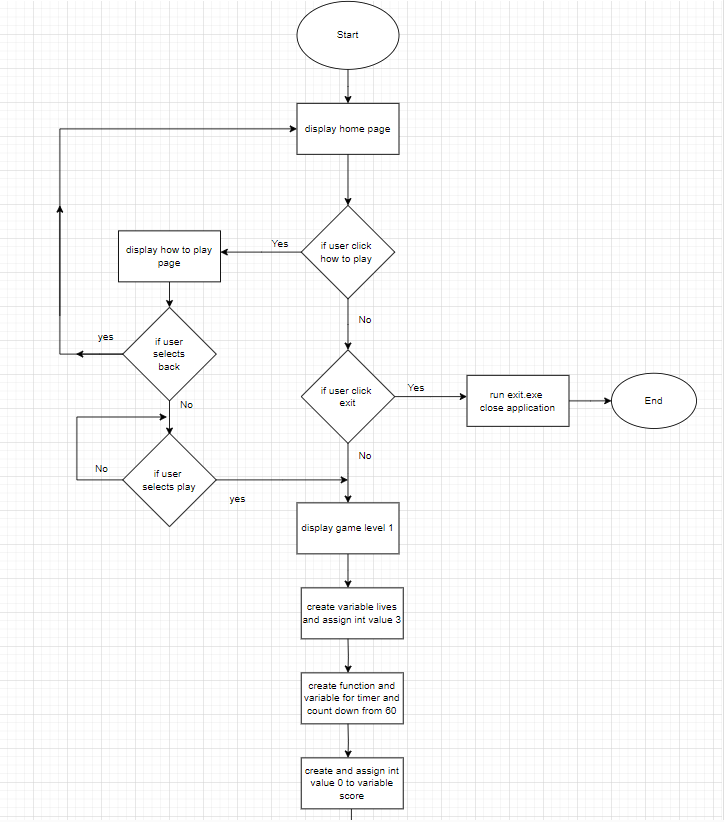
**Mindmap**



**Moodboard**



**Flowcharts of gameplay**



**Variable list (data dictionary)**

* Lives
* Score
* Timer
* Controls (W, A, S, D, SPACE BAR, ARROW KEYS)
* Character
* Enemy
* Key
* Diamonds
* Mob spike
* Platform
* Portal
* Start
* Exit

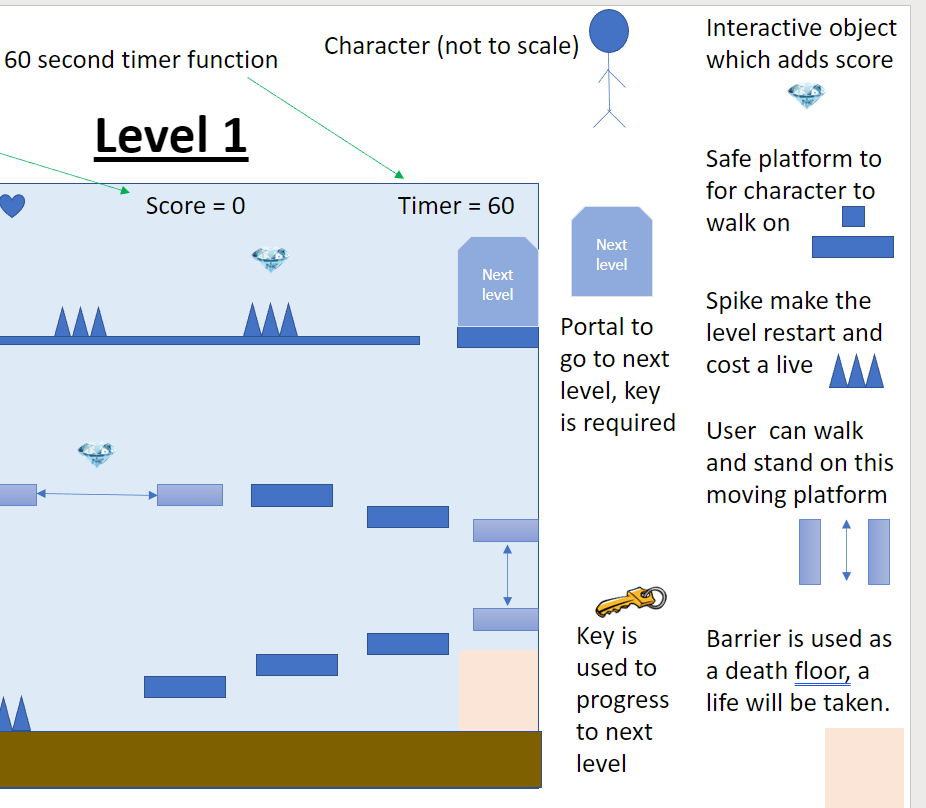
Lives = integer value = length = 1

Score = integer value = length = 3 – 4

Timer = integer value = length = 1 - 2

Start = Boolean = True

Exit = Boolean = False



**Pseudocode**

START

DISPLAY homepage

IF how to play is clicked THEN

DISPLAY how to play screen

IF user clicks back:

DISPLAY home page

Else:

IF user selects play THEN   
 Run Level\_one

Else:

Display home page

Else:

IF user clicks exit THEN

run exit.exe

Else:

DISPLAY Level\_1

DISPLAY level\_1

LIVES = 3

TIMER = 60

SCORE = 0

Def controls ():  
 IF event.key[pygame.K\_s] OR IF event.key[pygame.K\_DOWN] THEN

char\_y = - 10

IF event.key[pygame.K\_w] OR IF event.key[pygame.K\_UP] THEN

char\_y = 10

IF event.key[pygame.K\_a] OR IF event.key[pygame.K\_LEFT] THEN

char\_y = - 10

IF event.key[pygame.K\_d] OR IF event.key[pygame.K\_RIGHT] THEN

char\_y = - 10

Import time

Def countdown (time\_secs)

Countdown (60)

IF TIMER = 0 THEN

LIVES = LIVES - 1

OUTPUT “TIME UP”

IF LIVES >0 THEN

DISPLAY Level\_1

Else:

OUTPUT “YOU HAVE NO LIVES LEFT”

DISPLAY SCORE

IF user touches spikes OR IF entity touches user THEN

LIVES = LIVES - 1

OUTPUT “UNLUCKY”

IF LIVES >0 THEN

DISPLAY Level\_1

Else:

OUTPUT “YOU HAVE NO LIVES LEFT”

DISPLAY SCORE

IF user touches diamonds, THEN

Score = score + 100

ELIF user gets key THEN

OUTPUT “YOU CAN OPEN THE DOOR”

IF LIVES == 0 THEN

DISPLAY SCORE

OUTPUT “MAYBE NEXT TIME”

ELSE:

DISPLAY LEVEL 2

SCORE = SCORE + 100

IF user enters door 1 THEN

DISPLAY level\_2

ELIF user enters door 2 THEN

DISPLAY level\_3

Else:

OUTPUT “YOU WIN”

DISPLAY SCORE

END

**Test plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No** | **Test Data** | **Expected result** | **Actual result** | **comments** |
| 1 | Run code to see if home page is displayed | Main home page for game should show | “DISPLAY homepage” | The code will allow the home page of the game to be visible. |
| 2 | Does how to play page work | If user clicks on how to play, then instruction page should be shown | IF how to play is clicked:  DISPLAY how to play screen | An if statement is used, this allows the main menu page to switch to the games how to play page. |
| 3 | Can user exit the game | If user clicks exit, game shall close | IF user clicks exit THEN  run exit.exe | If statement is used to check if user selects exit button. Command run exit.exe will close the application |
| 4 | Is level displayed | Game screen should appear | DISPLAY Level\_1 | Code allows the game to show the first level |
| 5 | Set variables | 3 variables should be set and displayed | LIVES = 3  TIMER = 60  SCORE = 0 | Variables are declared and assigned int vales |
| 6 | Do controls work | When user presses on controls, character will move | IF event.key[pygame.K\_s] OR IF event.key[pygame.K\_DOWN] THEN  char\_y = - 10 | Example code shows when player presses the s key or down arrow then character y value is decreased by 10 showing that the character will move down |
| 7 | Does timer work | Timer should count down from 60 | Import time  Def countdown (time\_secs)  Countdown (60) | Code imports timer dictionary and uses functions to set a timer |
| 8 | Are lives deducted from game | The characters' lives should be lost if timer end or contact is made with enemy or traps | IF user touches spikes OR IF entity touches user, THEN  LIVES = LIVES - 1 | One of the examples shows if user touches spike or an entity then the live will be deducted by 1. |
| 9 | Is score added to the game | Score should increase if character catches diamonds | IF user touches diamonds, THEN  Score = score + 100 | Code displays that if user contacts diamonds, then the variable score will be increased by 100 |
| 10 | Does game go to next levels | When the user enters the door, the game should access the next level | IF user enters door 1 THEN  DISPLAY level\_2  ELIF user enters door 2 THEN  DISPLAY level\_3 | If statement and elif are used to elaborate how the game will display the next level if the user enters the doorway. |
| 11 | Does game have you win message | Game should say you win and show your score | OUTPUT “YOU WIN”  DISPLAY SCORE | Variable score is shown at the end and text saying you win is also displayed. |

**Design review**

I showed the design which I have made to 3 people and received their feedback. I received all positive feedback and a few areas to improve on. Most people agreed that the levels should be more challenging and there should be more levels as the game will finish too quickly

**Response 1**

* “The design of the interface was not clear. It seems like your design is resenting a black and white game. Adding some annotations to your work would improve it”.

**Response 2**

* “Your game looks good however it seems like it will finish very fast. Adding more levels can improve your overall design”.

**Response 3**

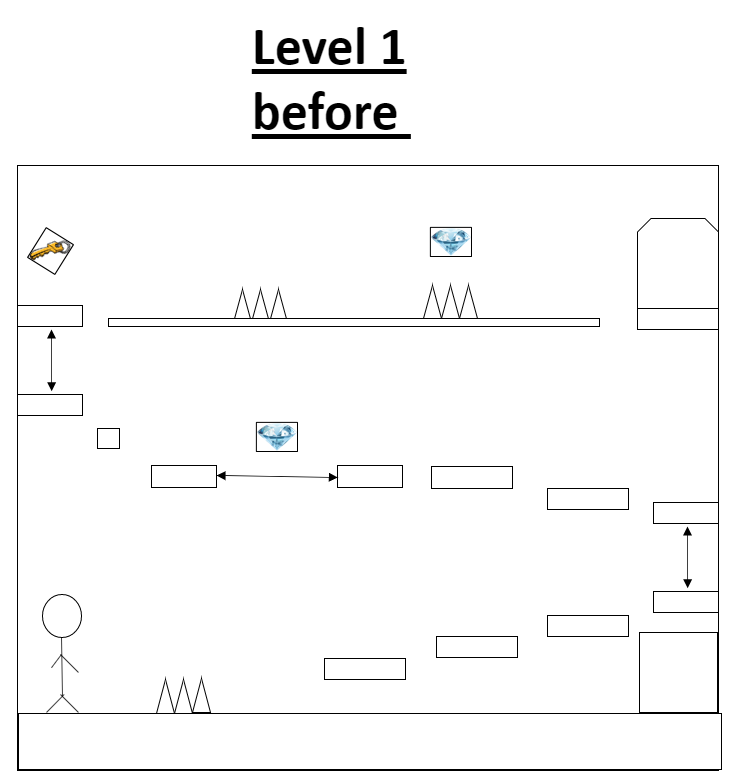
* “It was easy to identify the game from the mood board. The images clearly show an idea of how the game will be”.

**Response 4**

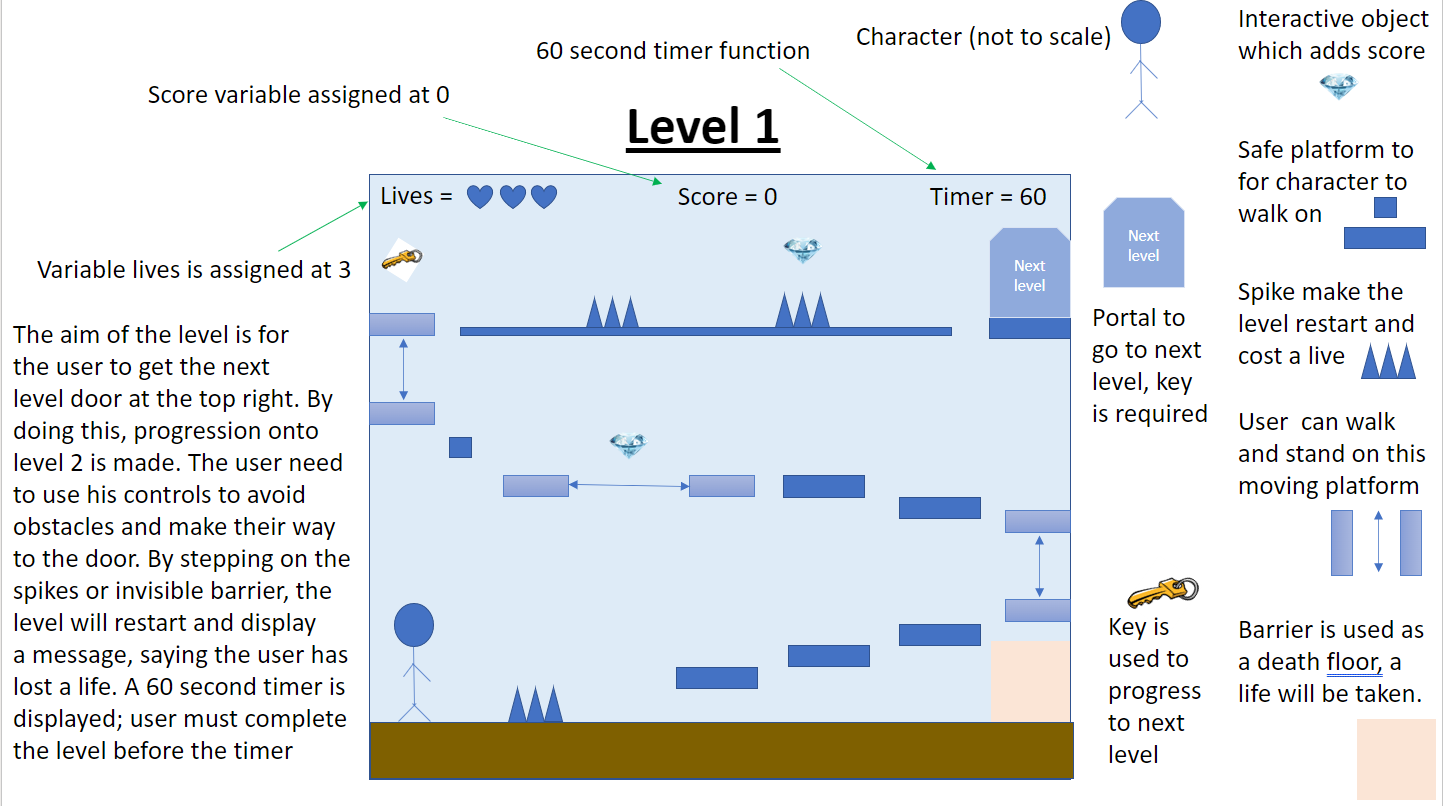
* “Your mind map makes sense but is very brief, by adding some ide to your mindmap can make your overall design better.”

Those who reviewed my game design also looked at the Pseudocode and Flowchart. They agreed that they both are similar in ways, and this should be achievable in the next stage.

**Changes made according to responses**

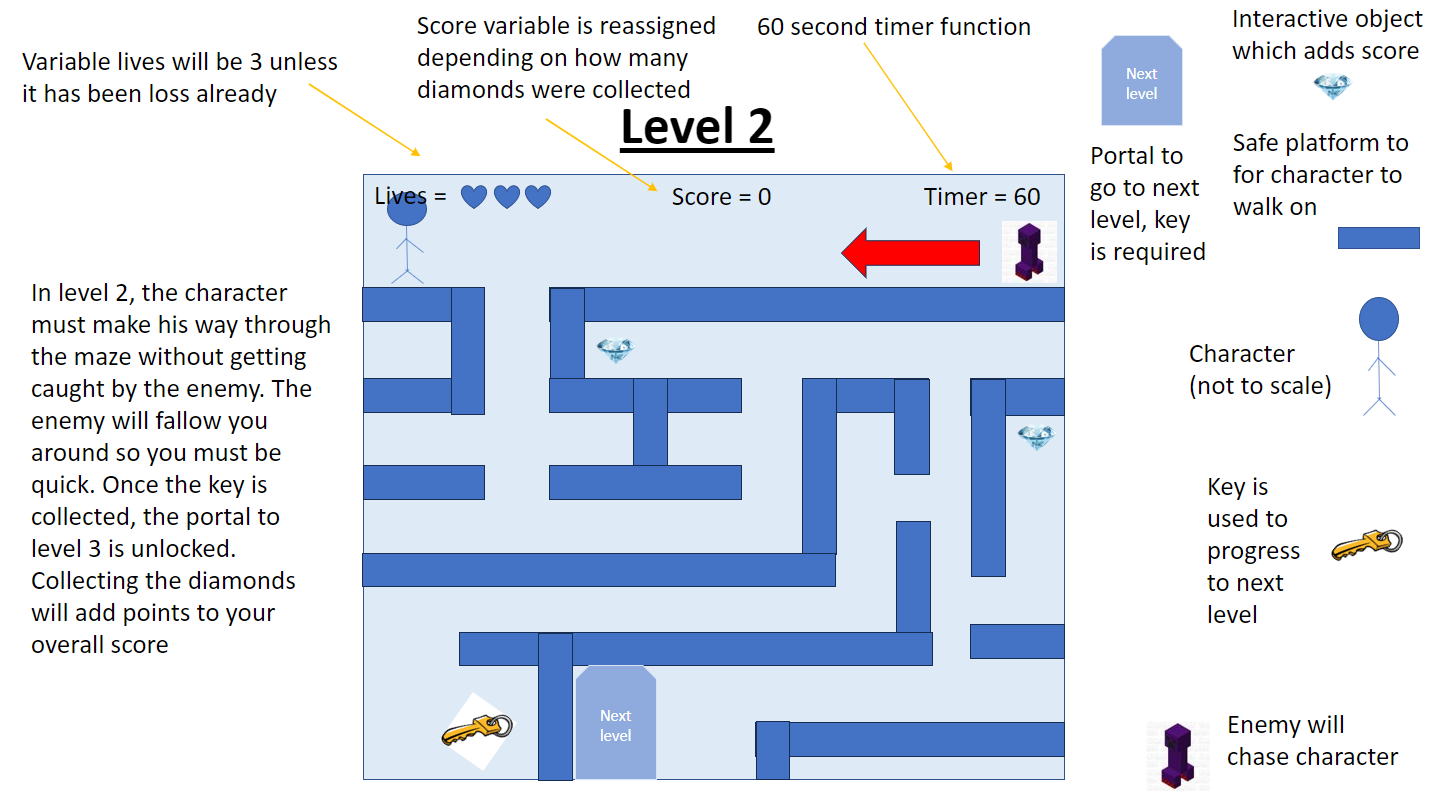
**Response 1 changes made.**

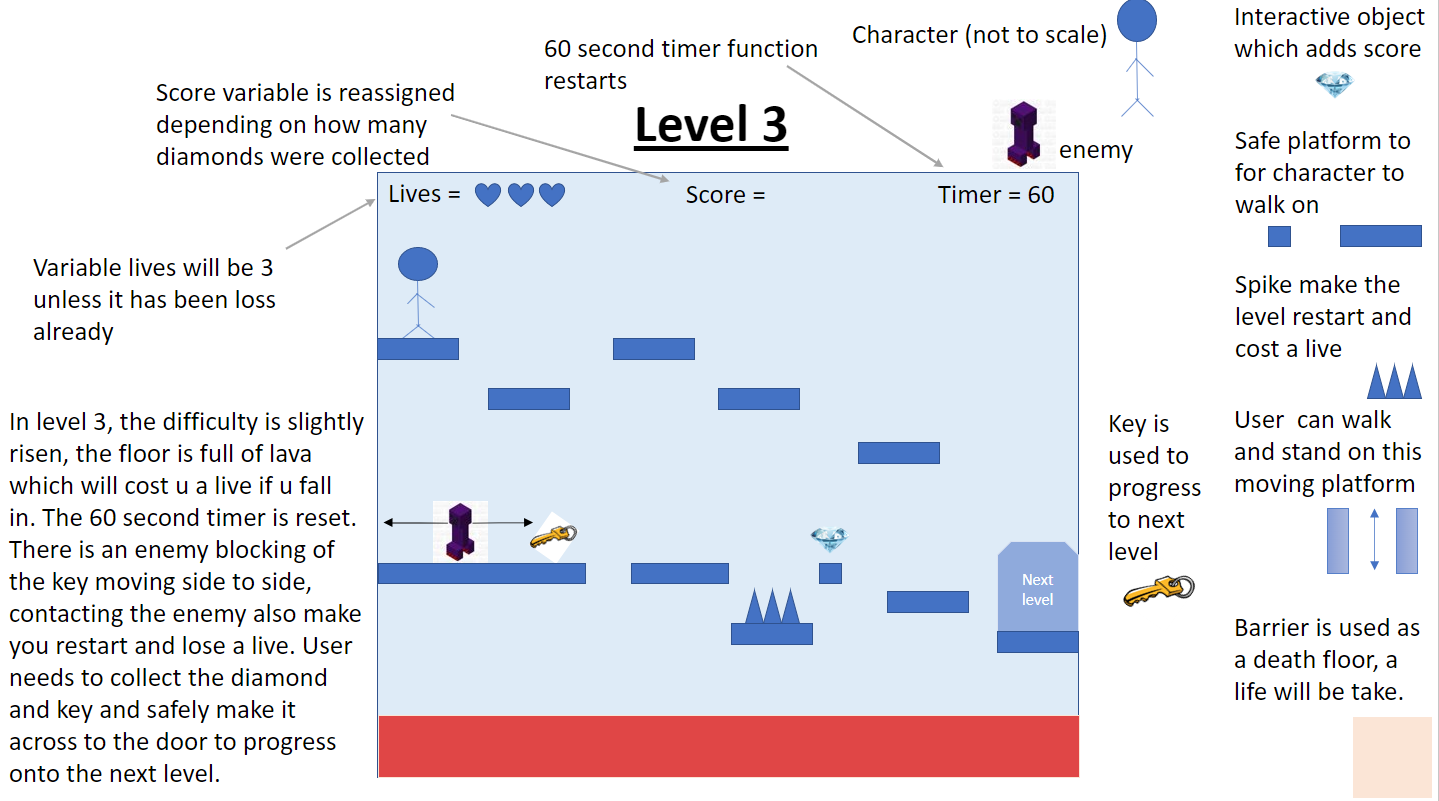
Level 1 design after feedback.



**Response 2 changes made.**

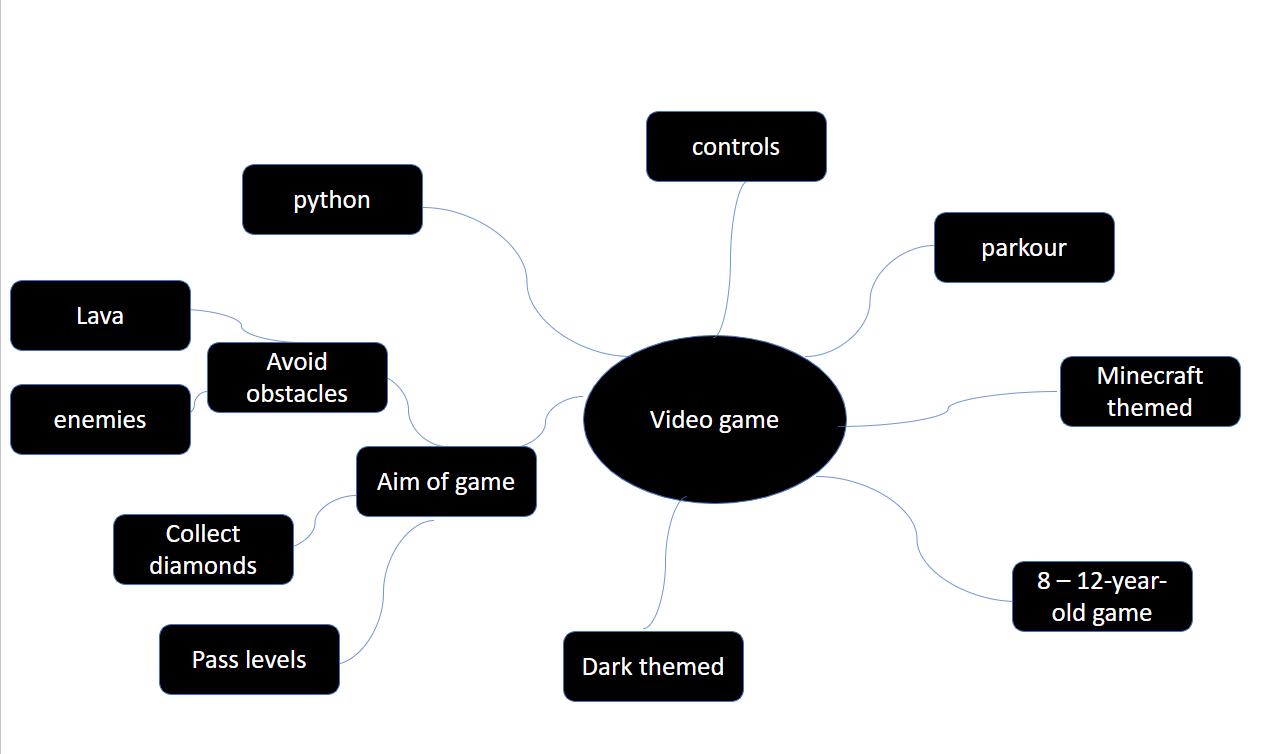
I added multiple levels to my game here are a few examples.



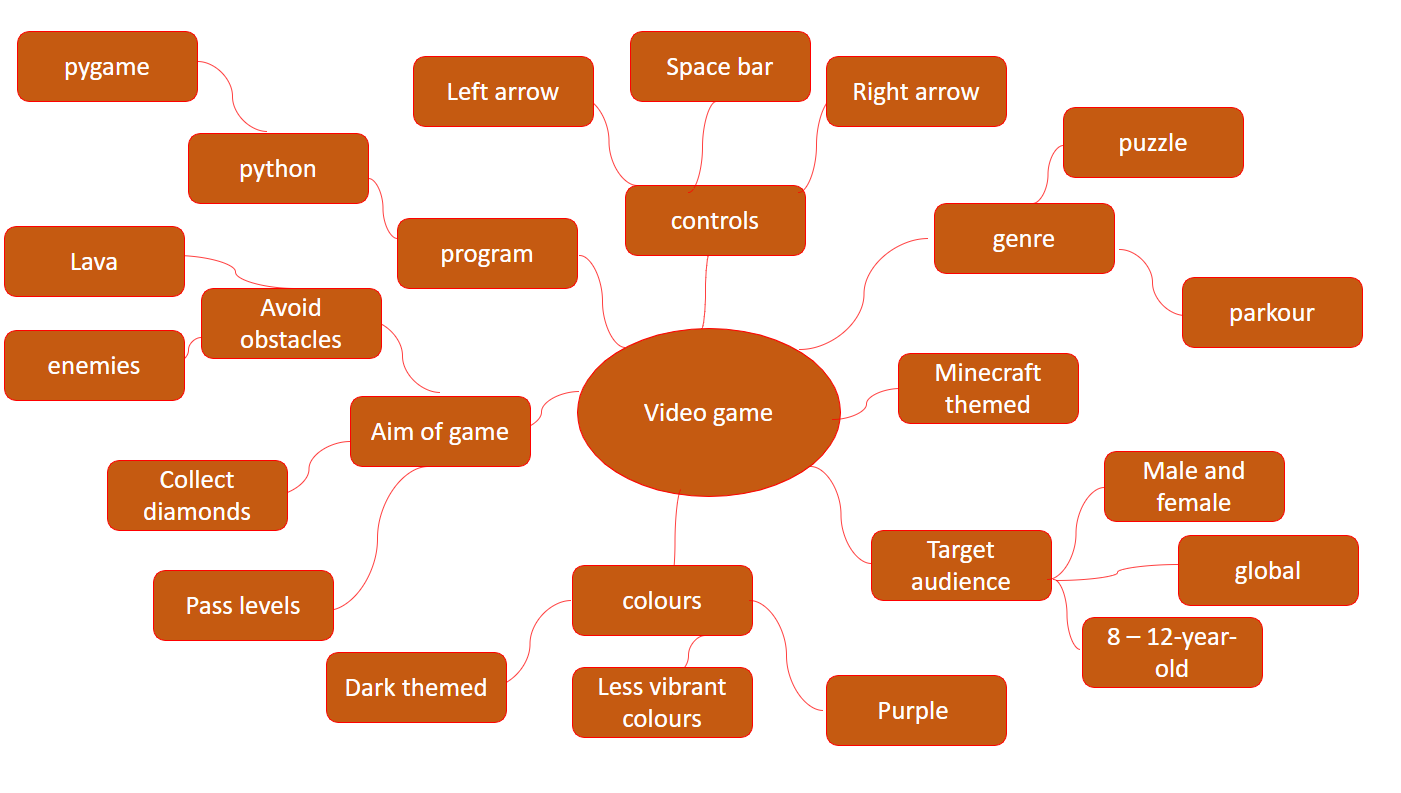


**Response 4 changes made.**

before



After



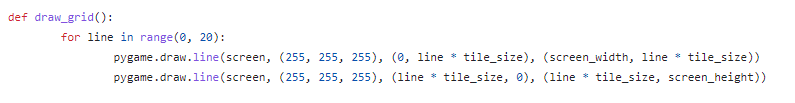
**Development review**

I began developing my 2d platformer game by installing pygame. This allowed me access the module and python game library. I decided to use python as it is my most experienced coding language.

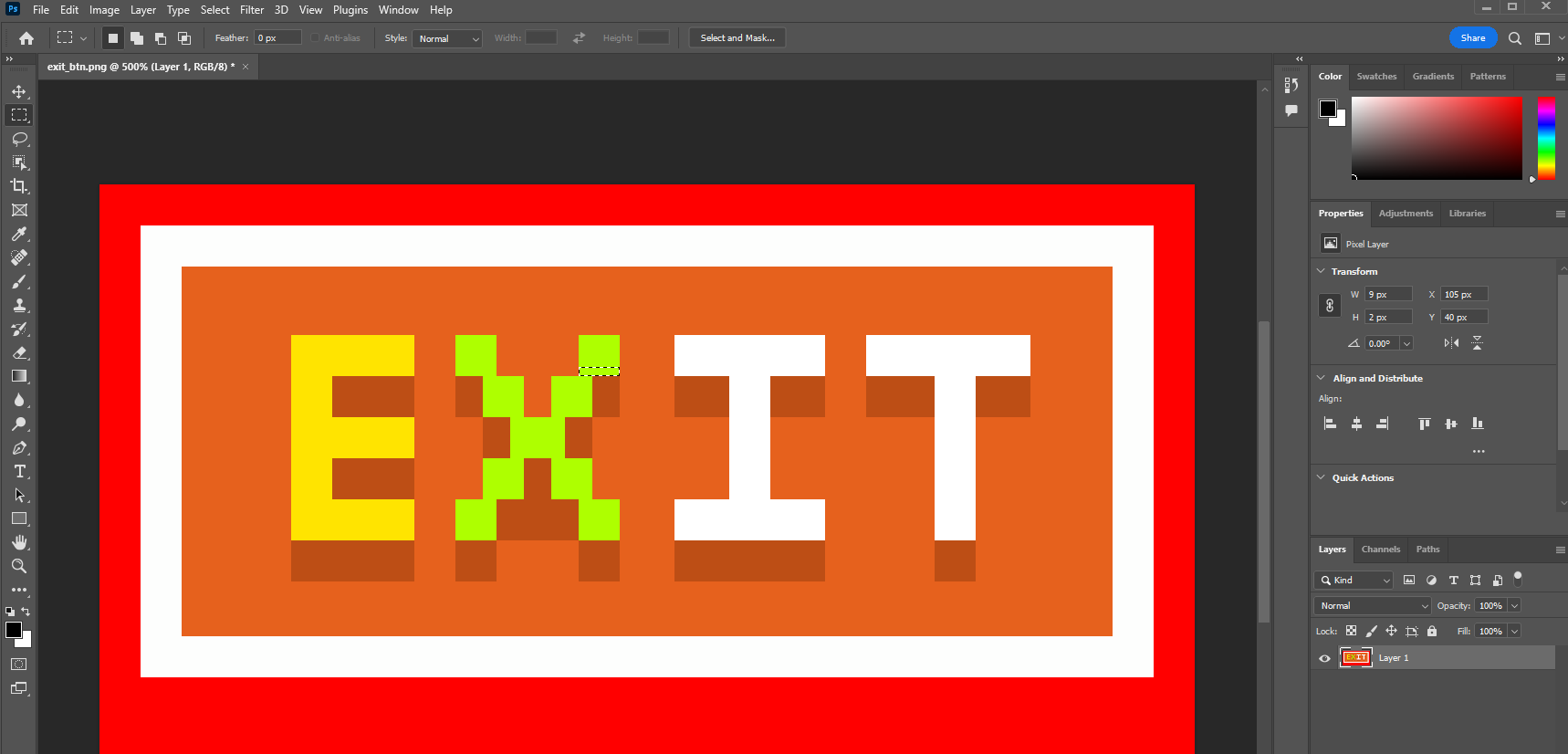


Once I ran python and pygame I began by creating the display. This was done by setting a screen width and height which can be used to display the game screen and I also created a grid which was 20 x 20 to help me see a clearer view of each pixel in my game.

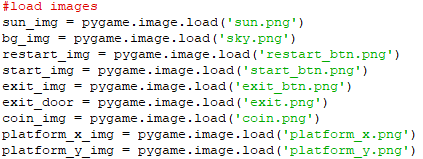




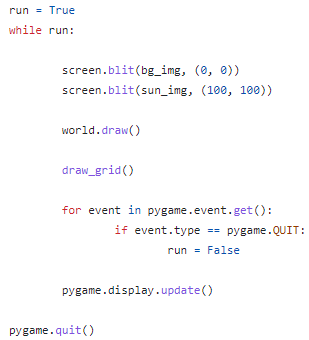
At this point in the development, I had a blank game screen. Then I decided to find some images which can correspond to my moodboard and game theme. I kept in mind the target audience and made sure the images were child friendly. Once these images were saved onto my device, I edited the images using adobe Photoshop.



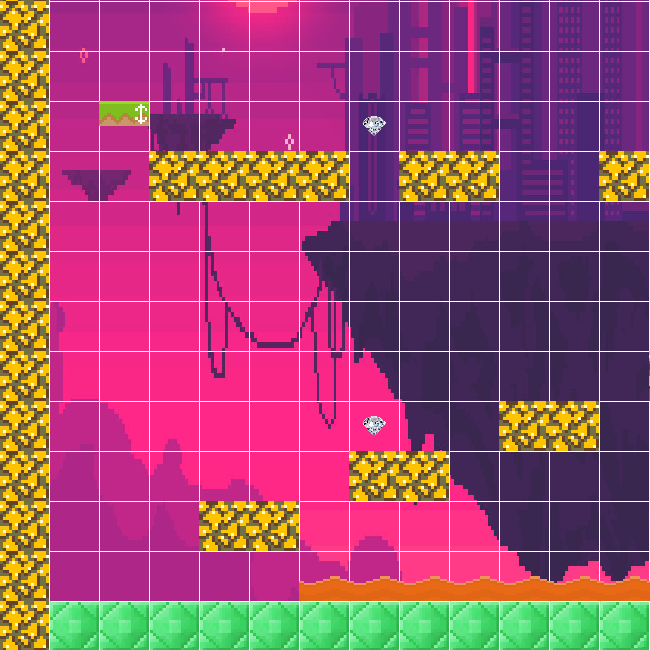
This prevented me from any sort of copyright or other legal acts. Eventually I defined my images into my code.



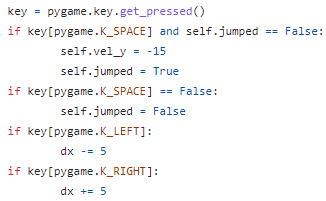
I then began coding the game world. This allowed me to have a starting level. At this point, when the code is running the first level of the game was ran. To create this, I had to use classes and define subroutine. I also made a list which defined all the world data using monochrome to change between blocks. Towards the end of the first part of my code, I assigned a Boolean value the variable run.



By running all the code which, I had created so far, I have a game page which looked like this.



This was not my original idea; however, I thought it would appeal more to the target audience. The colour scheme also stands out and appeals to the users.

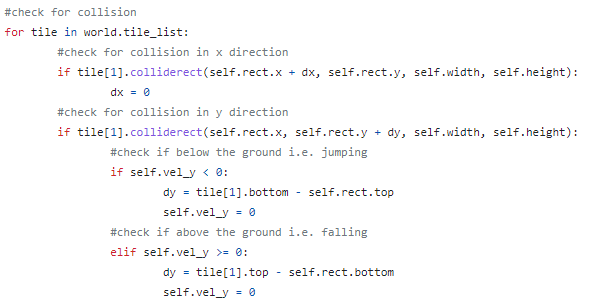
I developed some code for creating a character in my game. As I already had the images saved into my code, this stage was already half complete I just needed to call the code. Once again, I used photoshop to edit the image as I wanted the character to stand out more. Then I defined another class called player. This part was quite tricky but with some reaseach I was able to fix the problems. 



I developed some controls into my game so if the user clicked on the buttons, the character was able to move. The only problem I had was when space was clicked, the character kept on floating. This is when I had to add gravity to the y axis to allow the character to jump and land back down. I ended this part of my code by drawing out the character into my game. This was done with the following code.



By this time, I had a game which included a character and some blocks. The next problem was that the character was walking through everything. This I when I realized I needed to add collision into my game.

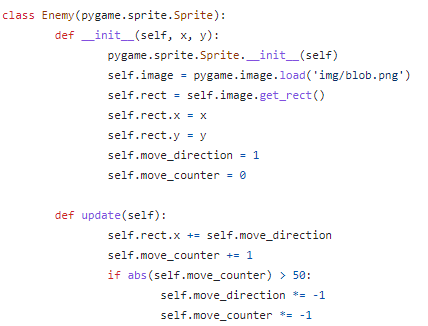


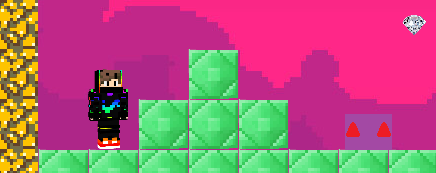
After I added the following if statements for my game collisions,



As you can see in the screenshot provided, my character can now avoid walking through blocks.

I then added an enemy into my game. This was done by the following code. I decided to change my idea from spikes to an entity. The reason for this was mainly because I wanted to base my game with Minecraft to follow my overall theme as it would be highly appealing to the target audience.





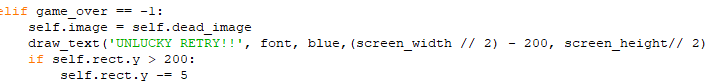
However, I then changed this enemy to a different kind. The main reason for it was because the color scheme was too close to the background color this made it hard to see, I also didn’t like the shape of a simple rectangle, so I made the enemy more circular and made it stand out more using photoshop.

I also added lava. This was simply done I creating a design using photoshop and then loading it into my game.





The next stage of this part was adding buttons into my game. For example, I created a restart button for when the user dies. By clicking on the button, the player will restart, however, they will remain on the same level.

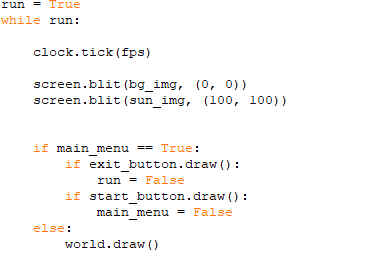




I then created a more colorful design for the restart button.



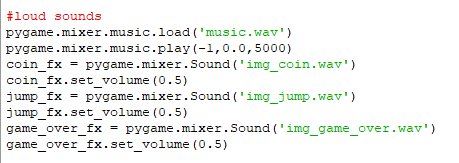
After this, I created a home page for my game. Due to time, it was most important to have a play end exit button first. Once I had these working, I would add more buttons on the main menu later.



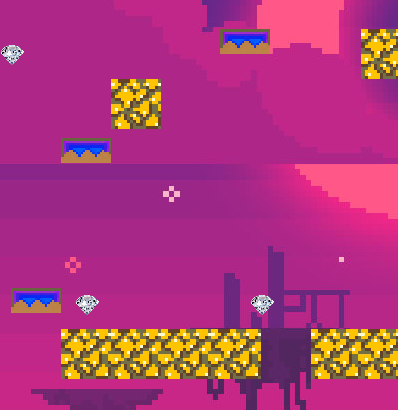
This part of code was the main part of getting the home page to be displayed.

After this, I added some sound effects into my game. For example, I have a sound effect for jumping, one for collection coins and a game which runs in the background. Below is the code which I had to research to get this.



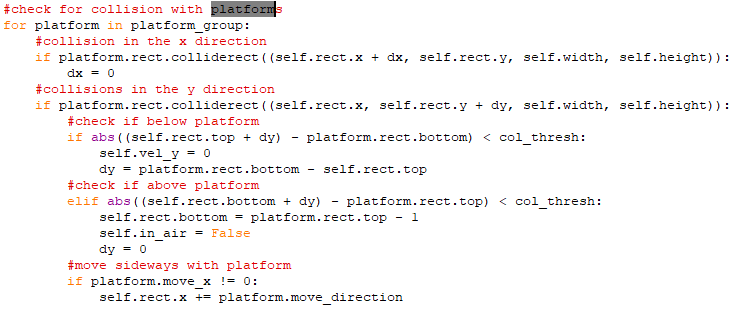


At this point my game was well working, however the only thing which was wrong was the platforms. I had platforms but they were stationary. I had to take more time to do some research and some not too long later I was able to amend this.



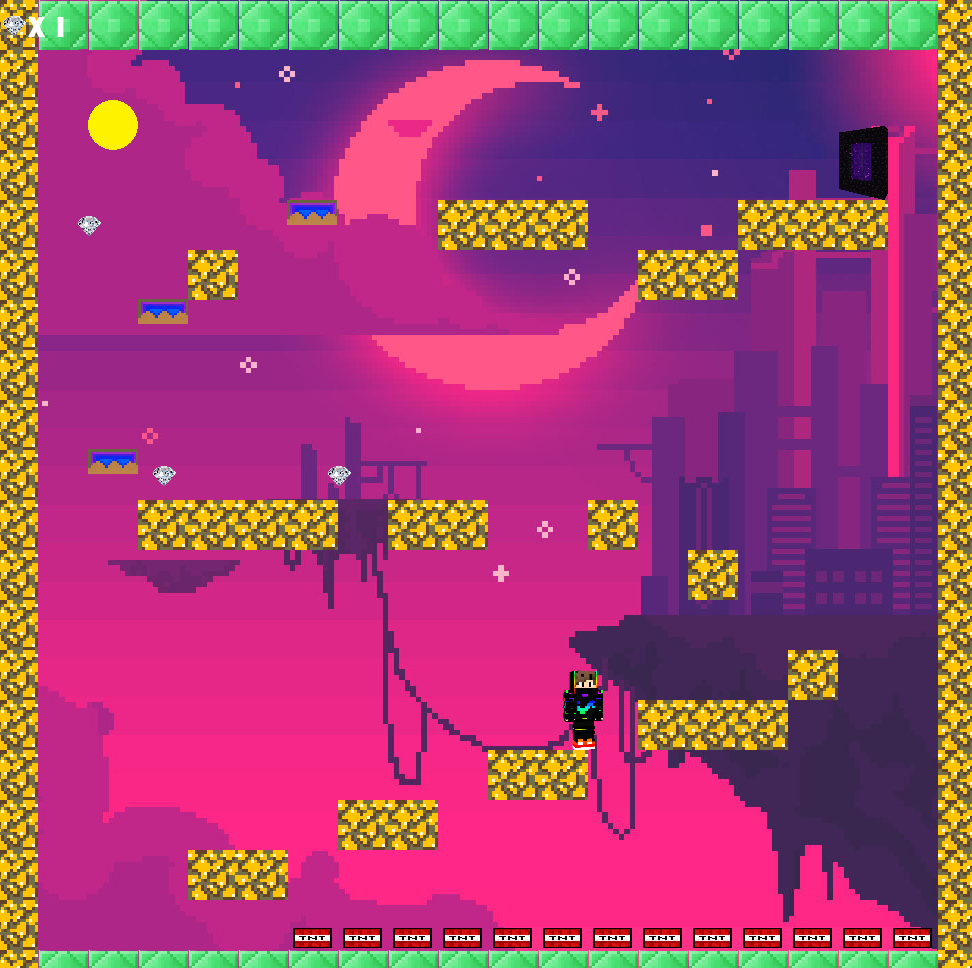
The above screenshots show proof that my platforms both move up and down.





My game has 7 working levels, and you win message at the end of the game.

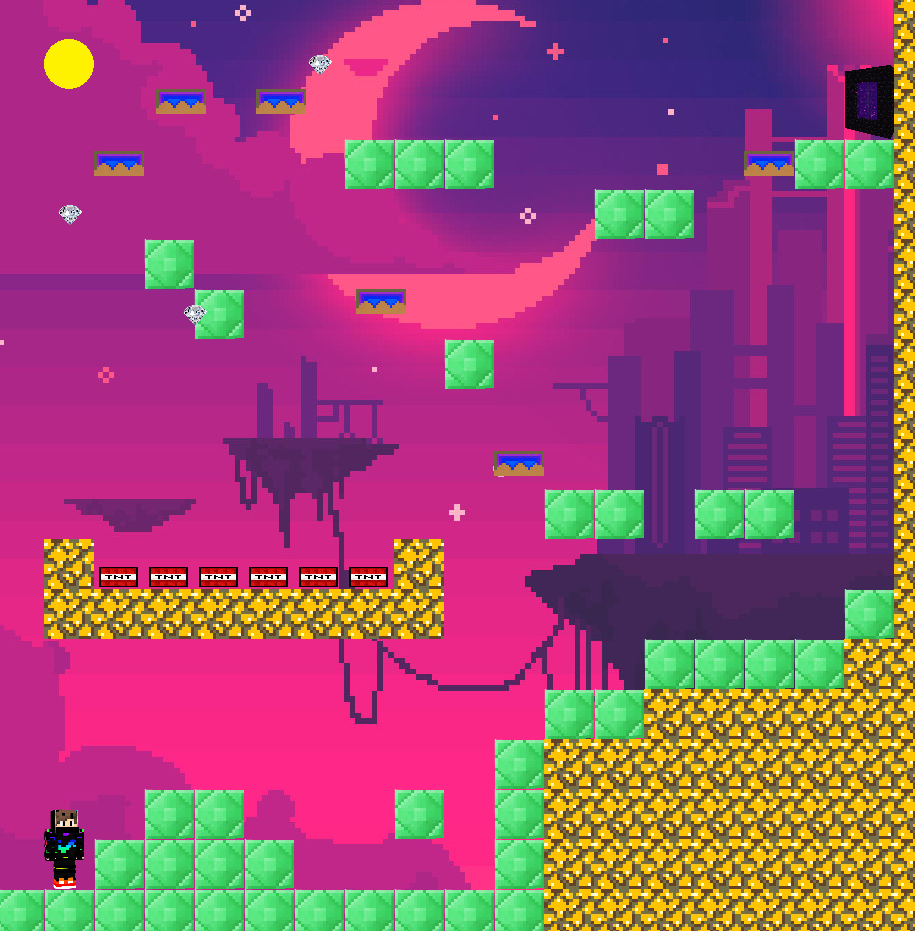
















**Conclusion**

Overall, I think I did well in this task and showed a lot of independence. One reason to justify this is because the majority of this code was done by independent research. Through this task, I have learned a lot of new skills and techniques.

I worked on this task starting of with very low knowledge of game development but over the course I have learnt a lot. I am happy with the game which I have made but I would have made more changes if I had more times. By having a limit of time, I had to make sure I completed the main game and made sure it was working.

I was hoping to set lives, get scores and a timer in the game.