

**This exemplar Unit 3 computing project was
produced by a Year 13 student**

Exemplar Project

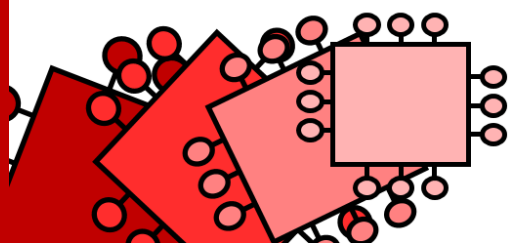
The project was marked at 61/70.

The mark was not adjusted during moderation.

Along with this document please see the following two files:

- Candidate 2 - (61 out of 70) - Additional comments.pdf
- Candidate 2 - (61 out of 70) - Mark Grid.pdf

These files show you, in detail how the best fit marking approach was applied to the actual marking criteria and provides an example of the additional commentary which was sent to the moderator to justify the teachers marks.



Candidate Name: [REDACTED]

Candidate Number: [REDACTED]

OCR GCE A COMPUTER SCIENCE PROJECT H446-03

Name: [REDACTED]

Candidate Number: [REDACTED]

Archway School: [REDACTED]

Title of Project: 2D Map Runner

H446-03 – PROJECT CONTENTS

TABLE OF CONTENTS

| | |
|---|-----|
| A. Analysis..... | 5 |
| Defining the Problems and Stakeholders | 5 |
| Justification of how the problem can be solved by computational methods | 6 |
| Research | 8 |
| Features of proposed solution..... | 14 |
| success criteria and requirements..... | 16 |
| Hardware & Software requirements | 19 |
| B. Design | 20 |
| Systems diagram..... | 20 |
| Game start | 21 |
| Main Menu | 22 |
| Exit game | 23 |
| Level selector | 23 |
| Level Viewer | 24 |
| Summary of processes including key variables and structures | 26 |
| Algorithms | 34 |
| Traditional Software development life cycle | 43 |
| Test Data for Development | 44 |
| Test data for beta testing | 45 |
| Sign off proposal | 49 |
| C. Development | 50 |
| D. Evaluation..... | 114 |
| Test table | 114 |

Candidate Name: [REDACTED]

Candidate Number: [REDACTED]

| | |
|-------------------------------|-----|
| User sign off agreement | 121 |
| Evaluation of Solution..... | 122 |
| Project Appendixes | 138 |

A. ANALYSIS

DEFINING THE PROBLEMS AND STAKEHOLDERS

My game is a level based map game where the player must complete the map whilst avoiding the obstacles, once they have completed it they will move onto a harder new map. They start at the beginning of the map, they must then use the boxes and platforms in order to proceed to the other end of the map. There will pools of lava which they mustn't fall in otherwise they will die, there are also tanks placed on platforms and other locations shooting missiles every few seconds in the direction the tank is facing; if the character is hit by one they will also die. On some maps there will be portals so that the person can complete the level quicker, going through the portal will move the character to another portal further up the map.

Due to the nature of the game it appeals to people of all ages as long as they have access to a computer and are able to play the game. The genre of my game is a platform game set in a 2 dimensional environment, where the player must control a character past objects along the platforms avoiding missiles and enemies. There wouldn't really be any extra requirements outside the game except it possibly being possible but this is standard on most games. Because my game is based on New Super Mario Bros (NSMB) the problem has already been solved, I have decided to add to it by changing the appearance of the game, there will also be different characters and enemies as well as the introduction of missiles. Aside from that the general structure of the game will be same and will resemble that of the original game. New Super Mario Bros was actually made for Nintendo DS and Nintendo Wii so by replicating the game on the computer I have solved a problem because people can play it on computer as well as the Nintendo consoles.

A stakeholder is anyone with an interest in the actions of a business or organisation, however in this case it would be someone with an interest in the game being made. Due to the origins of the game it would appeal to a wide variety of people who will be the stakeholders, however I will be aiming at people who have an interest in the original Mario game and now want to play it on computer which is what my game is addressing.

One of my two stakeholders will be Robbie Needham, he has an interest in computer and console games and being 14 spends most of his time playing video games. He was a massive fan of NSMB on Nintendo DS but he is disappointed now that he has moved away from the DS that NSMB bro isn't available on other consoles, as well as on computer. Robbie wants me to recreate new super Mario bros on computer as well as adding new things and abstracting from the original game.

My other stakeholder will be Aaron Walkley, he is 17 years old and like Robbie is a massive fan of video games and spends most of his time playing them. When he was younger he was a big fan of NSMB but outgrew the Nintendo DS and moved to Xbox and therefore couldn't play NSMB anymore. He would've liked to have been able to play it on a console such as Xbox or PlayStation or even computer, I am therefore solving his problem by making a NSMB computer version which will be of similar presence to the original game but will have things changed and things removed.

JUSTIFICATION OF HOW THE PROBLEM CAN BE SOLVED BY COMPUTATIONAL METHODS

THINKING ABSTRACTLY AND VISUALISATION

- Abstraction is the process of separating ideas from reality, removing things that are unimportant and only including things that are. Games frequently use abstraction by removing things that are important to the game whilst also adding things that don't occur in reality.
- Movement will be simplified in my game, you will be able to control the character with the up, down, left and right keys. The up key will be a jump button so when it is pressed the character will jump, left and right will move the character forward and back whilst the down button will allow the character to crouch. Despite these being standard movements in real life the character's arms and legs won't actually move abstracting and simplifying the characters movement.
- My game will have a score and timer displayed at the top of the game, this isn't shown in real life and is therefore an abstraction or visualisation.
- Similar to that of New Super Mario Bros my game will have obstacles and weapons for which the character must avoid to stay alive, this can be considered as abstraction as it is unlikely to happen in reality.
- The game will have player lives, they will be able to collect them in the maps and the player will lose a life if they are hit by a missile and other obstacles.
- The graphics will be heavily simplified, the maps will be tile based so they will be made up of tiles to make a map. So grass platforms will be green tiles and the sky or background will be made of blue tiles.

THINKING AHEAD

- Thinking ahead is the process of identifying inputs and outputs to a solution before you have tackled the problem, you can then think of potential problems and then you will have time to think of how that problem could be solved.
- It will therefore be important for me to think of inputs and outputs for my game before I have even started to code it. I can also think of potential problems which I might have when coding my game and then how to solve those problems
- One of the things I was least unsure about when thinking through my game was getting the camera to follow the character as they progress through the map. However I have looked into this and found some code that will enable the camera to follow the character like in NSMB.

THINKING PROCEDURALLY & DECOMPOSITION

- Thinking procedurally is the process of breaking a problem into a series of more manageable tasks, these work together to make the game. All games are made up sub routines responsible for doing individual tasks which count towards the overall build of the game.
- By using sub routines and procedures in my game I can make it more efficient and prevent code from being duplicated.
- The spawning of maps will be the same and the way that objects spawn onto the map, the only thing that will be different is the actual look of the map.
- The character and character movement will be the same throughout the game. The character will be the same throughout the game although adding some kind of

customization for the character would be good. Character movement will be one of the main procedures for the game and will be applicable for the whole game and will not change.

- I will also only have one piece of code for shooting missiles and enemies throughout the game, they will all do the same thing for all the maps and thus I only need to write the code for it once. The process for the character dying will also be the same for instance if they fall into water or are hit by a missile then the level will start again and they will lose a life.

THINKING LOGICALLY

- Thinking logically outlines any decisions points and then whether it results in branching or looping. Branching looks at a decision and then that decision could result in a number of different outcomes, looping redoes that section of code until a decision is met. Decisions within code are likely to affect the flow of the program and what happens.
- One of the main decision points in my game will be the menus, this is because there will be different options that result in different things happening. For instance clicking start game results in the game start bit being run where maps are called and other things happen. Another example is select level which allows you to select level and could result in different outcomes depending on the user's choice. There will be also other functions on my menu acting as decisions.

THINKING CONCURRENTLY

- Thinking concurrently is the process of implementing parts of a solution or program concurrently or side by side, e.g. by determining what parts of a problem can be solved at the same time.
- I don't think I could implement thinking concurrently throughout any parts of my solution, the only thing would be for my multiplayer mode where the movement for player 1 and player 2 will need to be done at the same time.

RESEARCH

LOOKING AT A PREVIOUS GAME: NEW SUPER MARIO BROS.

NSMB is a platform game developed by Nintendo for their Nintendo DS console, it was based on their original side scrolling game Super Mario Bros. It is mostly a single player game due to it being on Nintendo DS which only allowed one player gaming. The aim of the game is to defeat Bowser's workers in order to rescue Princess Peach, however in the maps there are power-ups he can use to gain an advantage such as being small or bigger being able to shoot fire as well as others. The game is based on a 2D platform which my game will also be based upon.



Controls: In NSMB because it is on the DS you control the character different and with the Nintendo DS controls, and therefore you jump with A and sprint with Y. The actual position or movement of the character is controlled using the arrow keys on the console. The bottom screen of the DS (touch screen) is then used to operate the stored item and use it as a power up.

Power-ups: The character has access to a wide variety of items in NSMB which are seen in the maps, they either give him a special power or change his size. The character can eat mushrooms to change the size that they are, mega mushrooms makes the character large, mini mushrooms make the character shrink to mini size and standard mushrooms make the character their standard size. The character has access to only two weapons which are fire flower and star man, fire flower allows the character to shoot fire at enemies to defend themselves whilst star man makes the character invincible.



Levels: The game is based around a number of worlds, in these worlds there are a number of levels that the player must complete. In NSMB there are 8 worlds which all have different themes making the game different. There are not necessarily the same number of levels in each world, there are also secret passages which can be unlocked with coins, and this makes completing a world faster.



Lives: In NSMB the player starts off with 5 lives and starts the first world. The levels have enemies which are trying to stop you from finishing the level, you will only lose a life if you fall of a platform and fall into lava or where there is a gap and you must start the level again. If you run into enemies and don't have a power up and are standard Mario then you will lose a life and you must start the level again, if you are big Mario and or another power up you will shrink to the size below and will not lose a life.

Things I like from the game:

| Game Feature | Justification – Could it be incorporated into my game? |
|--------------|--|
| Power-Ups | I like that the character can use power-ups in order to help them through the level faster, such as the giant mushroom which makes Mario big and the star making Mario invincible. I would like to incorporate the concepts of these power-ups into my game but this might be hard. The fire Mario would be reasonable; it would allow my character to shoot fire at the enemy in my game. |
| Graphics | In NSMB the graphics are simple yet affective this is definitely a something I would like to incorporate into my game, the levels are made of a fixed floor platform at the bottom of the screen with platforms above this which have coins and power-ups usually. I would like my game to be of similar build but also have new graphics which aren't in the original Mario Bro's. |
| Camera | The camera follows the character in NSMB meaning that a good view of the screen is available at all times, this would be hard to implement into my game but wouldn't be impossible. |

Things I don't like from the game:

| Game Feature | Justification – Could I improve it for my game? |
|----------------|---|
| Live System | I don't like the complexity of the lives system; it is too hard to understand. I like the idea of lives so I would have that in my game however I think that it would be easier to just have one level of the character so when it is hit they die and loses a life rather than shrinking to a different level but staying alive. |
| Worlds/ Levels | I like how many levels there are in NSMB and I like that the theme changes depending on the world but I feel it is too long winded and repetitive. I would have a smaller number of levels which are harder and test the user and then have some kind of high score system for the player which means they can compete against their friends. |
| Pausing | On the Nintendo DS on NSMB the pause button is the start button on the bottom part, this is often hard to reach when playing the game. I would therefore like to make it easier for players to pause my game when playing it on computer. |

FIRST INTERVIEW

Planning

I will be conducting this interview to see how much my client played NSMB the game my game will be based upon, by listening to someone who has played the game I know what they like and don't like and therefore what I can add and remove from the game. The game will also be changed to the specification they want as they might not want it to be the same as NSMB and will want changes in colours and design. This first interview will mostly find what parts the client liked and disliked from NSMB and will allow me to see things that can be built on and improved. I will mostly be getting the clients requirements in the second interview.

Interview Transcript

Dan

Rob

Did you play New Super Mario Bros?

Yes, it was one of my favorite games a few years ago.

What did you like most about it?

It was a very addictive game and I spent a lot of time playing it, it was a very polished game that was fun to play whilst also being difficult at times.

Did you like playing it on Nintendo DS?

It was good originally playing it on Nintendo DS when it was first out and when the DS was new however now I have moved onto playing XBOX I don't really play NSMB or my DS.

So would you like the game to be on computer?

Yes

What made you play it for so long?

I like that it was something you could easily just pick up and play and resume from where you'd finished because of the levels system, it was very addictive.

What made it addictive?

I dunno, the features of the game made it easy and fun to play meaning you could constantly play it, it was challenging whilst also not being too challenging.

How could I make my game addictive?

Keeping the main features of new super Mario bros and then building on the bad bits from it to make it an overall better game. Don't change it too much.

FOLLOW UP INTERVIEW – WATCHING ROBBIE PLAY NSMB

Planning

My aim for this interview is to get Robbie to play NSMB, and then get his opinion on things that he likes and dislikes from the game. This will help me better understand the things that he wants to have in the game and things he doesn't like and doesn't want in the game.

Watching him playing the game



Interview transcript after Robbie had played NSMB

How long did each level take you?

They took around 2 minutes to do, some took longer than others if they were more challenging.

Do you like the controls on NSMB?

I like that you use the arrow keys on the DS to move but find it hard to use the buttons sometime for jumping, sprint and using power-ups.

What controls would you like for my game?

The arrow keys for movement, and then space to jump and possibly enter and shift for spring and power-ups.

Did you like the backing music and effect?

The backing music is iconic to the game but I would have it muted usually, however I do like the noises for jumping and shooting.

So you wouldn't want backing music?

No, but noises for jumping and shooting would be good.

What would you consider as the limitations for playing on Nintendo DS?

It's outdated, the graphics are poor and the screen size is small its poor compared to newer generation consoles.

What screen size would you like for the game?

Roughly 900x900 pixels

I was thinking of making it so that the camera follows the character and then have a more zoomed in window just displaying a bit of the map, do you think this is a good idea?

It is a great idea that would be good.

The game will start by showing the menu, what would you like on the menu?

Start Level, a mode for seeing what the levels are and the high scores for each level and then some kind of character selection with character editor maybe.

Do you like the idea of having themes for some of the levels?

Yes

What sort of themes would you like?

Desert, Standard theme, Colourful, Dark

To start with what three power-ups would you like?

Way of shooting enemies, way of becoming giant and destroying platforms and an invincible mode.

How long should the power-ups last for?

20 seconds.

Do you like how the levels and worlds work?

No I'd prefer if there were just levels.

How many levels would you want to start with?

10

Would you like to be able to collect coins?

Yes

Do you think there should be a timer?

Yes

How long should the timer be?

I think it should count down from 300

How would you want the score system to work?

Timer multiplied by number of coins?

Yes, I think that would be a good idea

Do you want some form of enemy whether that be another character or tank shooting at the player?

Yes, that would make the game harder.

Would you prefer a tank shooting missiles, or character like on NSMB?

I think having a tank would make things more interesting and would give the game an edge.

Would you like an information bar at the top of the screen that displays coins and timer information?

Yes

Would you like a game currency, for instance as a way of purchasing locked levels?

Yes, but it should be the high score that goes into this.

If possible, do you think that it would be good to have the green pipes like on NSMB to teleport further up the map?

No, I'd rather have something like a portal but only one on each map and some kind of think blocking it making it hard to use in some way.

To make the game more competitive do you think that all the levels should be locked except the first and then you must achieve a certain score in order to unlock the next level?

Yes

How many lives should you start the game with?

5 but I think you should get 2 more every time you complete a level or maybe more or have some way of collecting lives.

What should happen if you run out of lives?

There should be the option to purchase lives if the player has game currency to do so, if not then they will have to start the game again.

Are there any other features that you would like that haven't already been agreed?

Ability to save game. Possibility of a multiplayer where its 1v1 to who completes the map quickest.

FEATURES OF PROPOSED SOLUTION

- The character will be fixed in terms of different of states it shows, so no matter the direction the direction the character will not change. The player will be able to move the character forward, backwards and then also be able to jump those will be the basic movements of the player.
- The player will be able to collect power-ups throughout the levels, to start with the game will have 3 different power-ups.
 - A shooting power-up where the player presses enter and bullets come out the character, the player will be able to use this to destroy tanks (enemies).
 - There will also be a power up which makes the character twice as big and the character will be able to destroy tanks easily and will also destroy air platforms when they walk into it.
 - And lastly there will be an invincible power-up where the character will remain the same size however he will not be killed by tank missiles.
- When you first start the game you will be met with a vibrant and colourful menu, on the menu there will be 3 different options (possibly more), such as start level, high scores / level viewer and character change.
 - Start level will play the first level, once the first level has been finished the player will be able to play the next level and so on.
 - There will then also be a way of seeing all the levels so you can look at them before you play them, it will also show which ones are locked and unlocked and the high scores for each level.
 - Finally, there will be a place where you can change the character you are playing as. With further game development there might also be a section where you can edit the player to make it more personal and involved. Due to my programming ability I think it will be hard to have a character editor.
- I have decided that in the game there will not be an enemy character, with NSMB this became too complicated. My change on the game will be that the enemy for the player will be tanks, these tanks will shoot missiles in the direction they are aiming roughly every few seconds. The tanks will move along the platforms a few tiles before going back and forth, the character cannot disable these enemies except with the power-ups. So when a missile hits the character they will lose a life and will have to start the level again.
- There will be 10 levels, there will be about 4 with the standard theme and then the rest will have special themes such as desert, dark and colourful but also the possibility for others. All levels will be built in similar ways; they will have a base platform which will be fixed. There will then be floating platforms with the possibility of some moving, these will be objects and thus can be destroyed if invincible/ bigger character runs into them.
 - The desert levels will have orange colours with desert related objects and the background will be desert linked. Platforms will also have sand on like in the desert.

- The standard theme levels will resemble that of new super Mario bros so the base platform will be grass the floating platforms will be boxes and the background will be a blue cloudy sky.
- The dark theme will be similar to the standard theme however the lighting will be reduced to make it seem darker, the background will also be a dark night sky rather than a blue sky.
- I have also decided to have colourful levels where the level will be the same as the standard theme however there will be random colours to make the game more colourful.
- If I have time available then I will look into adding a portal system which will be similar to that of the green tubes in NSMB, going through a portal will send the character up-to the next portal and therefore further up the map.
- As requested by Robbie the controls will be; LEFT arrow for moving left, RIGHT arrow for moving right, SPACEBAR will be for jumping. Other controls may be used if there is a need for more things to be controlled.

SUCCESS CRITERIA AND REQUIREMENTS

| Requirements | Justification |
|--|--|
| Minimum of 10 levels | Robbie has requested that the game start by having 10 levels (interview page 9), there is the possibility to have more levels if there is enough time. |
| Have a menu that has 4 options | When the game first starts it will go to the menu, Robbie has requested that the menu has 4 options outlined below. |
| Have a minimum of 3 power-ups | Robbie has requested that to start with there be 3 different power-ups; one that makes the character giant or bigger, another that makes the character be able to shoot a weapon of sort and finally a power-up that makes the character invincible. |
| Power-ups should last for 20 seconds | Once a player has received a power-up it will last for 20 seconds before returning the character to the original state as requested in the interview. |
| Controls: Arrows for movement, spacebar for jump. | Robbie has requested that the controls for the game be; arrows (left and right) for moving the character and then spacebar for the character jumping. |
| There will be a timer counting down from 300 throughout the level. | Like in NSMB Robbie thinks there should be a timer counting down, he thinks it would be reasonable to have this at 300 (5 minutes). This might be susceptible to change depending on the difficulty of the levels. |
| There should be around 100 coins on each level. | Robbie feels that there should be about 100 coins on each level similar to that of NSMB, this means that the maximum coin amount for a level will be 100. |
| High Score system (Timer x Coins) | There will be a high score system which works by multiplying the time left on the timer at the end of the level by the coins the person collected during the level. For instance, if the player finished with 150 seconds left on the timer and collected 88 coins they would have a high score of 13,200. |

| | |
|--|--|
| On the menu have a section which displays the levels. | Robbie has requested that one of the options on the menu be a section where you can check the levels possible view them in more detail. It will also display high scores for that level, this will also show the levels locked and how many scores (from high scores) the player will need to unlock the next level. This also means that the player won't necessarily have to play levels in order, the harder levels will require more coins to unlock them however. If a player doesn't have enough coins they can play an unlocked level again to gain more coins. |
| On the menu have a section for character selection. | To make the game more personable to the player, the possibility to play as a wide variety of different characters and sprites could be implemented onto my game fairly simply. NSMB doesn't allow this as you can only play as Mario and Luigi with a hack, this would be a problem solved and improvement on NSMB. The game will not have any characters from NSMB. |
| Appeal to ages 3+ | My game ideally should appeal to a wide variety of ages, those above 3 who have a computer. There is roughly an age of 3-10 who use a computer rather than a console so this would be my main intended audience. However, Robbie has also outlined his desire for the game as well as Aaron – people who played the original NSMB but stopped playing it as Nintendo DS became out dated. I think my rebooted game will therefore appeal to a wider variety of people. |
| Levels 2-10 will be locked to start. | When you first play the game all the levels will be locked except the first one, players will have to reach a certain number of game currency before they can unlock other levels. Player might end up having to play a level numerous times before being able to unlock the next level. |
| Level 1 should take no more than 2 minutes to complete. | The first level should be the easiest to complete, having this as a maximum time will allow me to determine the score required for the next level. Score required to unlock levels will get progressively harder and may require the player replaying a level. |
| The first few levels will not have a theme and will use the basic theme. | The first few levels will resemble the colours of NSMB, this means a blue sky background with possible boxes for floating platforms and grass for base fixed platforms. Aside from this the other levels will have special themes. |
| Character Editor option on menu | Robbie has requested there be a character editor built into the game however I don't think my programming skills would be strong enough to implement this. |
| Sound for jumping and shooting. | Robbie has requested that there should be no backing music however he wants there to be a nice for when the character |

| | |
|--|--|
| | jumps and shoots a weapon. This could possibly be expanded so that the power-ups make noises when in use. |
| Menu should be easy to navigate. | The menu will only have 3 or 4 options on it, meaning that it will be also really easy to navigate and use and it will be fairly self-explanatory. I will also look into being able to use the arrow keys and enter button to use the menu making it easier. |
| Have a special dark theme. | The levels will be split so some are standard basic theme and then the others will be a special theme of sort. Robbie has requested that one of these themes be a dark theme, it will be similar to the standard theme except lighting will be reduced and the background will be a dark night's sky. |
| Have a special desert / sand theme. | The levels will be split so some are standard basic theme and then the others will be a special theme of sort. Robbie's has requested that another of these special themes be a desert theme; this theme will have orange colours and sand on the platforms. The enemy will also have a desert theme and shoot orange missiles opposed to black. |
| Have a special colourful theme. | The levels will be split so some are standard basic theme and then the others will be a special theme of sort. The final special theme will be the same as the basic theme however it will be completely multicolored and will have a retro rainbow feel to it. |
| Save Game | Robbie has requested that there be the ability to save the game, he feels that it would take too long to complete all the levels so it would be good to be able to save your progress and come back to it. This also gives the option of having more levels if time is available to continue to develop the game. If the game were to be released then it will need more than 10 levels, maps won't necessarily take than long to build so adding more and more levels would be easy enough. |
| Ability to play 1v1 multiplayer. | Robbie has requested that one of the additional features be the option of having multiplayer, it would be a simple 1v1 where the aim is to complete the level in the shortest possible time, and they would also have access to all the levels. One person would be able to use the arrow keys whilst the other could use WASD letters. |
| Score bar displaying information at the top of the screen. | Robbie has requested there be a score bar or information bar at the top of the screen; this will contain number of coins collected in the level as well as time remaining and any power-ups the player has to use. |
| Player will have 5 lives. | Players will start on the first level with 5 lives as requested by Robbie. He also suggested that if the player runs out of lives they should be able to buy more with game currency. The |

| | |
|--|--|
| | player can gain lives normally by completing a level normally by which they will receive 2 lives extra. If a player runs out of lives completely and is unable to buy some they must start the game again. |
|--|--|

HARDWARE & SOFTWARE REQUIREMENTS

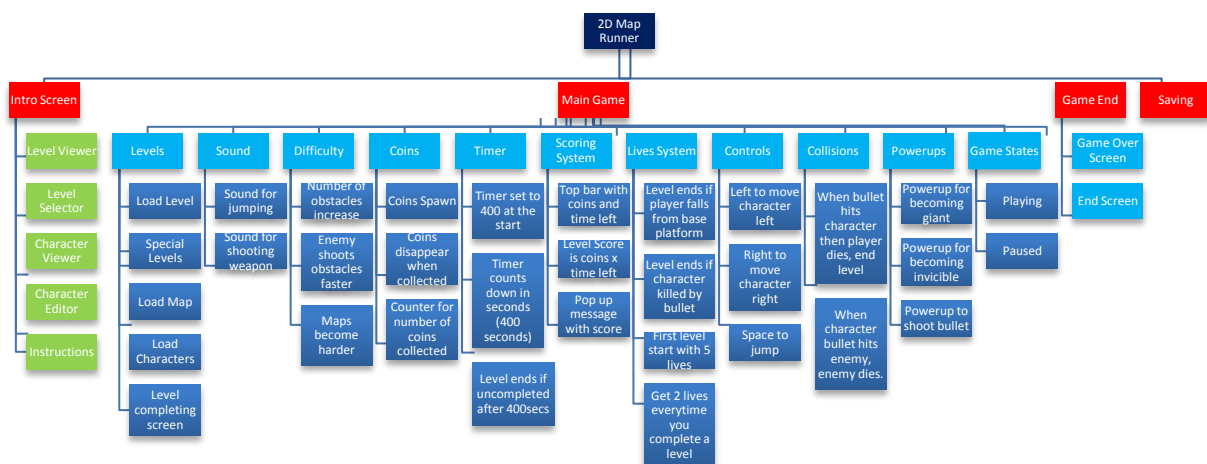
My game will have certain hardware and software requirements in order to play the game, however because it will be made using Monkey X and will compile with Monkey X, the requirements will be the requirements to run Monkey X rather than to run the game. However, I have provided some advisable software and hardware requirements.

| Requirements | Justification |
|--|--|
| Software | |
| Operating system: WINDOWS XP/VISTA/7/8, MACOS 10.7/10.8 OR LINUX | These are the operating systems that Monkey-X will work with, you might not be able to download Monkey X without one of these operating systems. |
| Monkey X programming language application. | You will need this in order to actually have the code and run the game. |
| Monkey X libraries | These are used as a way of having improved graphics and special features of monkey x, it would allow me to build a more advanced game. |
| Hardware | |
| Processor: 2GHZ+ | Will need around 2GHZ in order to do the graphics and animations of the game as well as the tasks. |
| Memory: 1GB+ RAM | |
| Graphics: 12*MB+ | You will require minimum graphics to run this game. |
| Hard Drive: No more than 250MB | The game will take up no more than 250mb |
| Computer Keyboard | Required to control the character. |
| Monitor | Way of being able to view the game and play it. |
| Speaker | There will be sounds for jumping and shooting. |

B. DESIGN

After my interview with Robbie I have collected all of the requirements for the game which means that I am able to start designing it. His requirements will change the way in which I design the game which means that it is important for me to think about everything, I have therefore broken the game down with a systems diagram. The systems diagram consists of smaller processes which will need to be designed, this diagram allows me to break the problem down into more manageable sections and processes.

SYSTEMS DIAGRAM



GAME START

- Introduction screen to the game loaded.
- Player will have to press space or enter to get to the main game menu.
- The user should be able to click space so that the menu is loaded.



- As you can see this is my first design for the loading screen which is the first thing the user will see, I have shown the two main parts to the graphics of the game being the sky and the wood platforms at the bottom which is open to change. There will then be a picture of the game actually working displayed below the title.

MAIN MENU

- Unloads loading screen / game start screen.
- In the original design of the game, the main menu will actually have 5 options.
- Game originally being built with no method of saving thus the game doesn't need to check for save game file or other.
- It will have 5 options: level selector, view levels, character viewer, character editor, instructions.
- When clicked these will go to corresponding page.
- Possibility of it being controlled with the arrow keys on the keyboard making it easy to use.



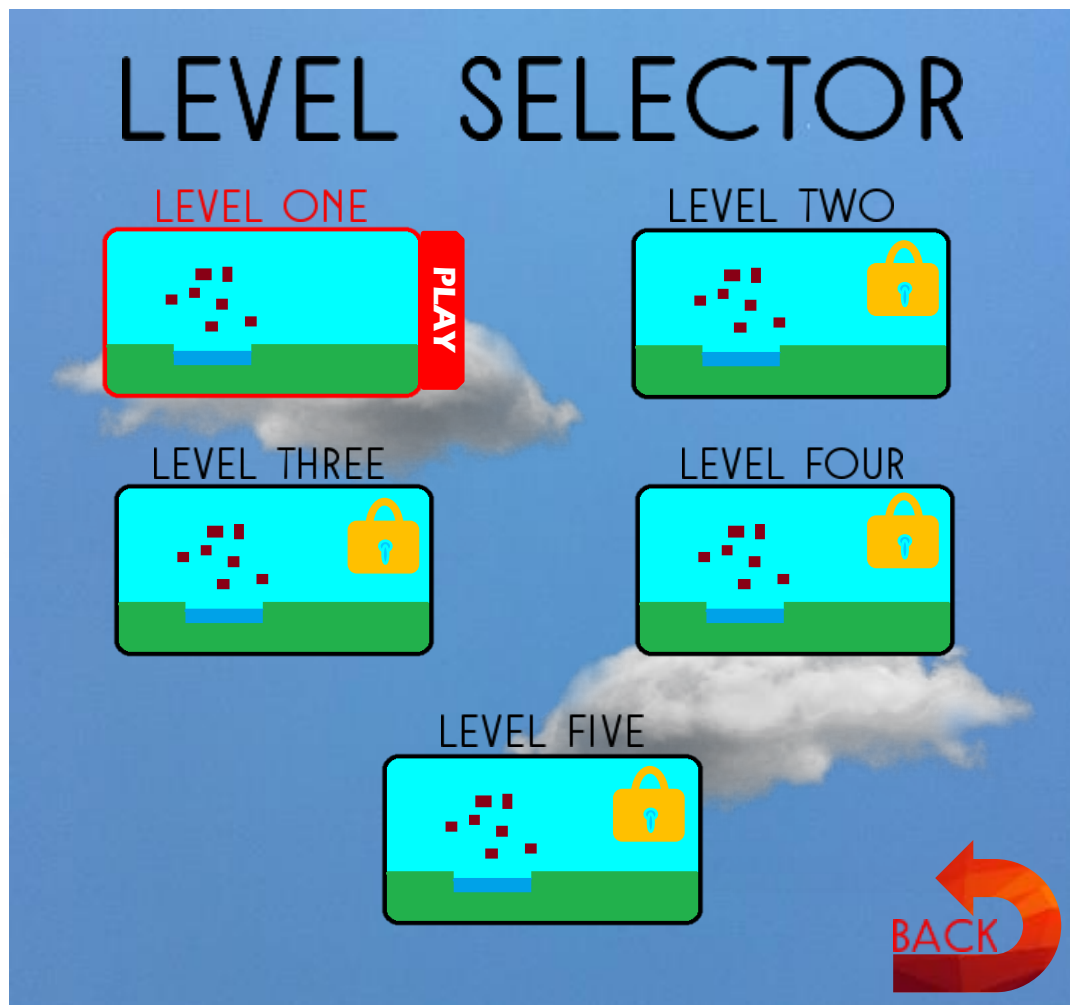
- These buttons are in the colours that my base platform will be, a grassy looking texture.
- When the button is selected there is a red outline around the button showing that it is currently selected.
- The exit button in the bottom right will close the game.
- When either of these are selected these will load the next screens.

EXIT GAME

- When exit button pressed on menu screen unloads all screens.
- Closes all scenes and screens.
- Goes to desktop.
- This won't be the same as game over which will be when a player dies during a level.
- This button on the menu screen will be the only way of closing the game.

LEVEL SELECTOR

- Loads after being selected on main menu.
- Main menu unloaded.
- This screen design will only have 5 levels to start with to give an example of how the screen should look.
- It will differ from the level viewer.



- Here is a rough idea of what my level sector will look like, it displays all of the levels.
- The current level selected (level 1) will go red to show it is selected and a play button will appear beside it which they can click to play.

Candidate Name: [REDACTED]

Candidate Number: [REDACTED]

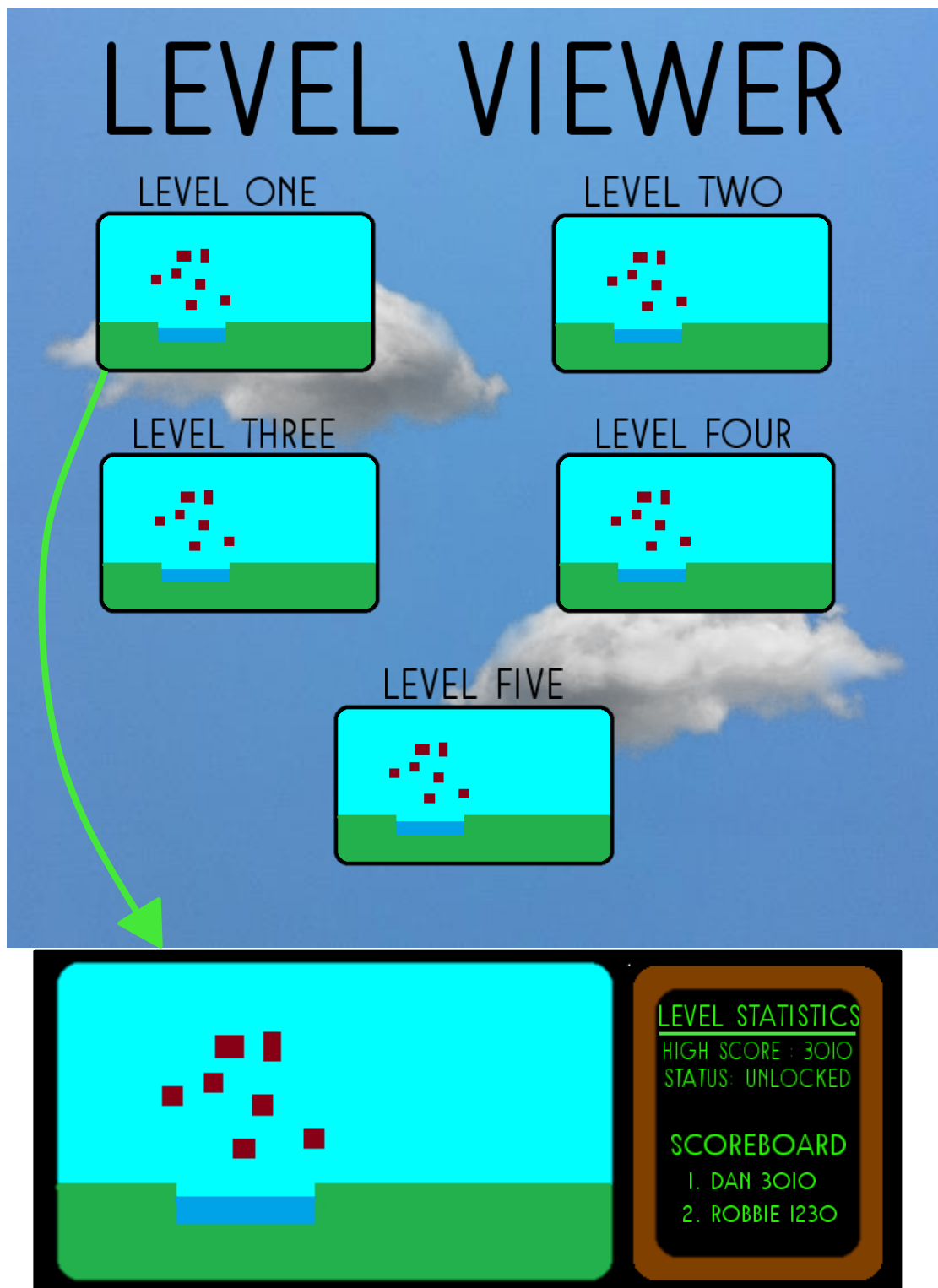
- The levels which are currently locked have a lock symbol next to them whilst unlocked ones have nothing, completed ones will have ticks in place of them.
- Back button goes back to main menu.
- These are just mock images of the level and are not like the real ones, actual level images will be added at a later date.

LEVEL VIEWER

- Loaded after being selected on menu.



- Menu unloaded



- The level viewer is useful for looking at a level in more detail, when the level is clicked it will display similar to above, the level will be shown in a bigger view so the player can see it.
- There will also be a level statistics section which displays high scores and other information.
- High scores will be updated once the level has been done again.
- Status is whether the level is unlocked or locked. Locked will have how many coins to unlock it.
- Back button bottom right goes to the main menu.

SUMMARY OF PROCESSES INCLUDING KEY VARIABLES AND STRUCTURES

MENU'S

WHAT HAPPENS?

- When game is started the loading screen will appear and be loaded, the player will then have to click space to go to the main menu.
- On the main menu there are a number of options, consisting of 5 buttons: Level selector, Level Viewer, Character selector, Character editor, instructions.
- Clicking on the level selector is the more important button because this is the only thing that leads to a further part of the game. Level sector lists all the levels that you can play within the game, in this case I have made it 5 to start with, with the hope to grow this to 10. When a level is selected that is unlocked
- Clicking on level viewer will display all the levels, clicking on them will display a further in depth image of the level, it will also show statistics for the level such as high scores and whether it is unlocked.
- The character viewer is a way of viewing all the characters available within the game and selecting which the player wishes to use.
- Character editor is a way of editing the character such as colours for instance, in the final game this is likely to be combined with the character viewer selection.
- Instructions will include controls and what the aim of the game is.
- There is also an exit button at the bottom right which will close the game.

KEY VARIABLES / DATA STRUCTURES AND CLASSES

- Loading screen example code;

```

Method OnCreate()
Select loading_screen = LoadImage ("loading_screen.png")

Method OnUpdate()
Select GameState
    Case "LOADING SCREEN"
        If KeyHit (KEY_SPACE) Then GameState = "MAIN_MENU"

Method OnRender()
Select GameState
    Case "LOADING SCREEN"
        DrawImage loading_screen, 0,0
  
```

- Main Menu example code;
- ```

Select main_menu = MainMenu ("main_menu.png")

 Case "MENU"
 Button inputs to use menu screen
 If KeyDown (KEY_L) Then GameState = "LEVEL_SELECTOR"
 If KeyDown (KEY_V) Then GameState = "LEVEL_VIEWER"
 If KeyDown (KEY_C) Then GameState = "CHARACTER_VIEWER"
 If KeyDown (KEY_E) Then GameState = "CHARACTER_EDITOR"
 If KeyDown (KEY_I) Then GameState = "INSTRUCTIONS"

```

## LEVEL SELECTOR

- When this button is clicked on the main menu it will display a screen consisting of 5 squares, these will be labelled as level 1 and so on to level 5.
- It will also be displayed which levels are playable and which aren't which will be obvious because of the lock symbol in the image of the level, the player is unable to do these levels until they have enough score to unlock the levels. Player also knows which level is currently selected because there will be a red outline around the level.
- They can also go back to the main menu by clicking the back button in the corner.
- When the player clicks on a level they can play, a red button will appear beside the level they can click on this and it will lead to the level.
- If a player clicks on one of the levels that is currently locked it should not display a play button as the player shouldn't be able to play it.

## LEVEL VIEWER

- When this button is clicked on the main menu it will display a screen consisting of 5 squares, these will be labelled as level 1 and so on to level 5.
- When the player clicks on one of these boxes it will go to a different screen, this screen will have a bigger full image of the level that has been clicked on.
- Beside this will be statistics of the current level, it will say whether it is locked with what is required if it is locked, it will also display the highest scores for the level.
- You will not be able to play a level through this you must go to the level selector.

## CHARACTER VIEWER

- When this button is clicked on the main menu it will display a screen consisting of a central character who will be big and the one currently selected, there will then be arrows either side so they can change between the available characters.
- My game will hopefully have a variety of characters which they can play with, these will also be editable in terms of colours however this will be quite complicated to code.
- The screen will have a central character big on the screen, this is the currently selected character either side of this character will be a left and right button so that the player can switch between the characters.
- To play with this character they just have it selected then click the back button in the bottom right and that character will be the character currently used, the player then just has to select the level they want and play and they will be able to use the character.
- To change the character they go in the character viewer scroll to the character they want click the back button and play a level and the character will change.

## CHARACTER EDITOR

- When this button is clicked on the main menu it will display a screen showing a broken apart image of the currently selected character.
- You will be able to change accessories, hair, skin colour as well as other things to personalize the character, to save the changes the player just has to click the 'make changes' button and then the back button to go back to the menu.
- You will be editing the character that you currently have selected.

- This will ideally be built in to the character viewer selection further through development of the game.

## INSTRUCTIONS

- When this button is clicked on the main menu it will display a screen with a large section of text.
- This will be an in depth guide on how to play the game such as;
- Controls
- How the lives system works
- How the score system works
- There will be a back button in the bottom left which will take them back to the main menu.

## LEVELS

---

- Level needs to be loaded, level graphics need to be loaded into array as well as characters and enemies.
- Characters need to be loaded and set in the right position.
- Count down from 3 before starting the level.
- Coins need to spawn.
- Coins score needs to be set to 0.
- Timer needs to be set to 400.
- When count down finishes and level starts timer counts down from 400.
- Coin score is incremented by 1 when a coin is collected.
- When player reaches finishing area level completing screen appears with the option to go back to main menu. Will also display players score for that level.

## SOUND

---

- Robbie doesn't want any backing sound so there won't be any.
- However he requested that there be a sound for jumping, this will work so that when the spacebar key is pressed the jumping sound will play.
- He has also requested there be a sound for shooting a weapon, there could possibly be a different sound for character bullet and enemy bullet to make it distinguishable. Having a trigger for this sound will be harder because the power up button won't necessarily just be when the character is able to shoot so it can't make a sound when the power up button is pressed. However there could be some kind of if statement which checks if the power-up button is pressed as well as having the shooter power-up. Alternatively there could be a distinct shoot button which they can use to shoot when they have the shooter power up.

## DIFFICULTY

---

- The first level will be the easiest level to complete, it will be important to have variable difficulties for my level, first level will be easiest and last level will be the hardest. They can be made harder by making jumps harder having more obstacles and having more enemies shooting at the player.
- Level 1 will have minimal jumps to avoid water and fire obstacles, the platforms over the obstacles will be easier to use and jump between so that there is less chance of the character falling in and losing a life.
- Throughout the levels there will be gradually more and more obstacles consisting of water and fire, if the character falls in they will have to restart the level and they will also lose a life. The enemy will also shoot their weapon more and more often, in the first level they might shoot every 5 seconds in

higher levels this will reduce to 1 second – this means the character will constantly have to avoid these weapons.

- The enemy will also become more agile where they are placed so they move faster towards the character, the easiest way for the character to get past the enemies at this stage will be via using power-ups.
- It is important in the earlier levels for the character to do the level without losing lives and gaining the two when they pass the level so that they have loads of lives in the latter stages of the game.

## COINS

---

- The coins will need to spawn at the start of the level, this will be on platforms and other hard places so that it isn't too easy to collect all the coins, the more coins the player collects the higher there score will be however they have to complete the level in the shortest time too.
- Coins will be collected by having an if statement which compares the coordinates of a coin with the coordinates of the character, if they match the coin will disappear and the coin score will increase by 1.

## TIMER

---

- When the level initializes the timer will need to be set to 400 seconds.
- It will then need to count down from 400 seconds, this gives the player 400 seconds to complete the level.
- When the player runs out of time it will need to say level / game over and go back to the main menu.
- It will need to stop once the player completes the level and then be used in calculating the players score.

## SCORING SYSTEM

---

- To start with when the player is completing the level it will not give the score, instead it will give a breakdown of the school so it will show you the number of coins collected and how long is left to complete the level.
- The coin score and timer will need to be reset at the beginning of each level.
- Coin score goes up one every time a coin is collected.
- Timer counts down from 400 until the player completes the level or runs out of time.
- At the end once the level is completed there will be a popup saying the score.
- The score is calculated by multiplying the number of coins collected (coin score) by the remaining time (timer value), so the higher the two values the higher the score will be.
- There will be a different popup message if the person achieves a high score, saying 'congratulations on getting the highest score'.
- This score will also need to be added on level viewer where it lists scores for that level, must give the scores in the right order. It would probably be appropriate to only store the 3 highest scores and no more.

## LIVES SYSTEM

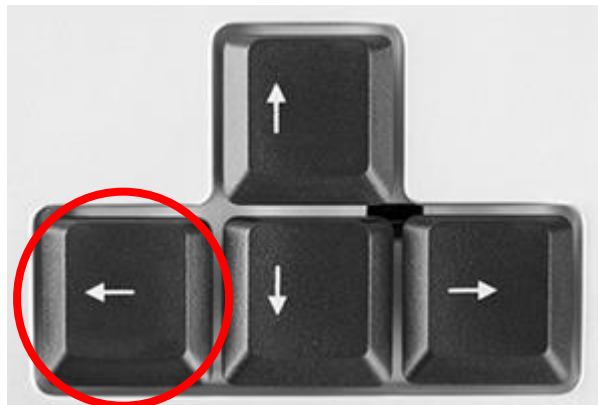
---

- My lives system is very similar to that of the original NSMB and is something that I have reused.
- When the player starts the game they will be given 5 lives.
- The level will end if player falls from base platform and they will lose 1 life.
- The level will end if player is shot by one of the enemy characters and will lose 1 life.
- When the player completes a level will receive 2 lives to their overall live count.
- In the event that a player loses all their lives the game will be completely over and they will lose all the progress they have made and start with 5 lives on level 1 and all the other levels locked.
- If the character has the invincible power up and are shot they will not die, and if they fall in an obstacle such as fire and water they will also not die and will be able to jump out.
- If the character has the giant power up and are shot they will return to their original size, if they fall into an obstacle they will not die and will stay giant.
- This means that it might be a good idea for the character with minimal lives to do easier levels again in order to get lives to allow them more chance at the level.
- However with further development of my game it might be possible to have a section where you can purchase more lives with the game currency.

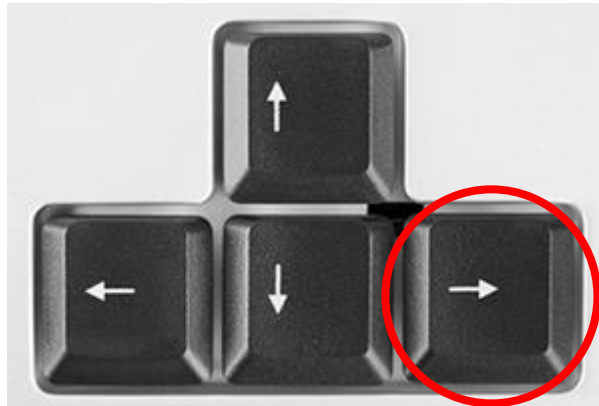
## CONTROLS

---

- Left arrow button used for moving the character to the left this will be done in terms of coordinates when I code it, for instance clicking left should move the character x coordinates to the left or change the x coordinate by a certain amount.



- Right arrow button used for moving the character to the right this will be done in terms of coordinates when I code it, for instance clicking right should move the character x coordinates to the right or change the x coordinate by a certain amount.



- The spacebar button will be used for jumping this will be done in terms of coordinates when I code it, for instance clicking spacebar should move the character up x coordinates or change the y coordinate by a certain amount.



- The enter button will be used for using the powerups this will require code to distinguish between the powerups because they do different things. Simply when the enter button is pressed the power up that was collected will be used.



- The p button will be used for pausing the level, a pause screen will appear, pressing p again will un pause it with a 3 second countdown.

## COLLISIONS

---

- There will need to be a collision for when the enemy shoots a bullet at the character, this will be relatively simple to implement. For instance when bullet coordinates match the coordinates of the player then the player will die.
- There will also need to be a collision detection for when the character falls into one of the obstacles, such as when the character falls into the water or fire.

## POWER-UP'S

---

- There will be a power up for becoming giant, this will entail an If statement for if the giant power up actually has been collected.
- When the power up has been collected and the enter button has been pressed to trigger the use of the power up the player size will change to 3x what it was.
- It will last for 20 seconds and it gives the player the ability to charge through platforms destroying them in the process.
- During this time the character will also be immune to enemy weapons.
- They will also be immune from dying to static obstacles such as the water and fire traps, in this event the player will just be able to jump out of the obstacle.
- When power up finishes player will return to original size.
- There will be a power up for becoming invincible, this will entail an if statement for if the invincibility power up has actually been collected.
- When this power up has been collected and the enter button has been pressed to trigger the use of the power up a flash text screen saying 'You are Invincible' on will appear.
- The character will also become a flashing color.
- This power up will last for 20 seconds and gives the character complete invincibility to every way of dying, the person should try get as far as they can in the 20 seconds to advance quickly through the level.
- When the power up finishes, player will return to normal.
- The final power up will enable the character to be able to shoot bullets, this will entail an if statement from if the shooter power-up actually has been collected.
- When the power up has been collected and the enter button has been pressed the character will be able to shoot every time they click the enter button.
- This will work the same way as the enemies will however shot rate may be increased to provide more of an advantage.
- Shooting a bullet / missile at the enemy will result in the enemy dying, however this power up doesn't make the character immune from being killed in any form.

## PAUSING

---

- Robbie has requested that he should be able to pause the game whilst playing the level, all he will have to do is click the p button and a pause screen will appear and everything will stop.
- Unpausing will result in a 3 second countdown giving the player time to get ready.

## GAME ENDING

---

- The game will not end if the player dies in the level however it will start the level again and they will lose a life.
- If the player loses all their lives, then it will go back to main menu and everything will reset as if they have just started the game.



Candidate Name: [REDACTED]

Candidate Number: [REDACTED]

- The player can close the game by clicking the exit button on the main menu, it will take it back to the computers desktop

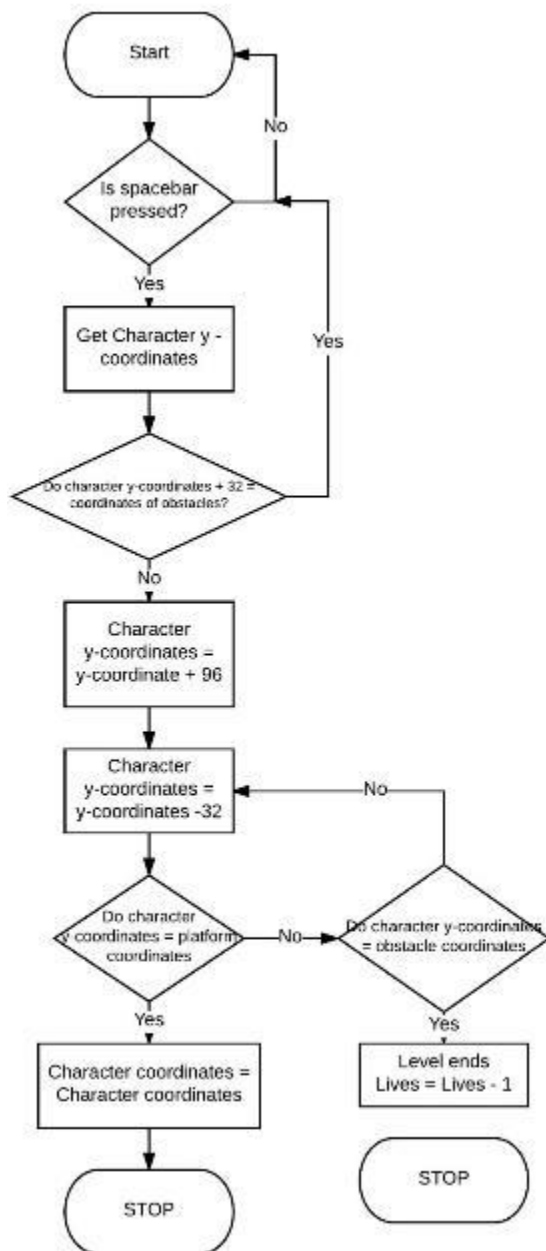
## MOVEMENT

---

- If the left button on the arrows is clicked, then the character will move to (current coordinate - 32) along to the left
- If the right button on the arrows is clicked, then the character will move to (current coordinate + 32) along to the right.
- If the spacebar button is clicked, then the character will move to (current coordinate + 32) above.

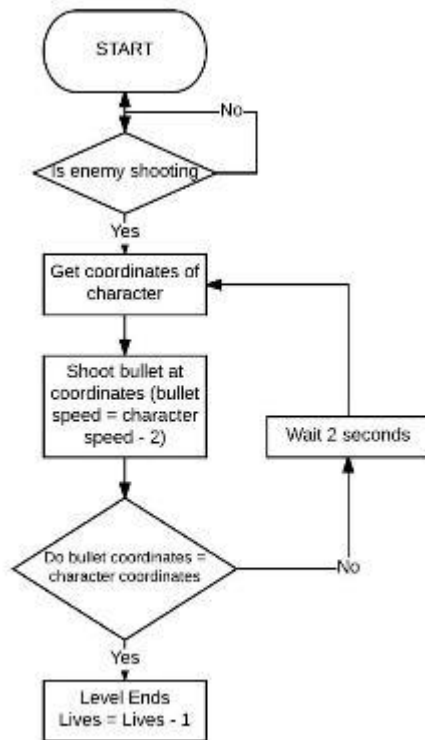
## ALGORITHMS

## MOVEMENT FOR JUMPING (SPACEBAR)



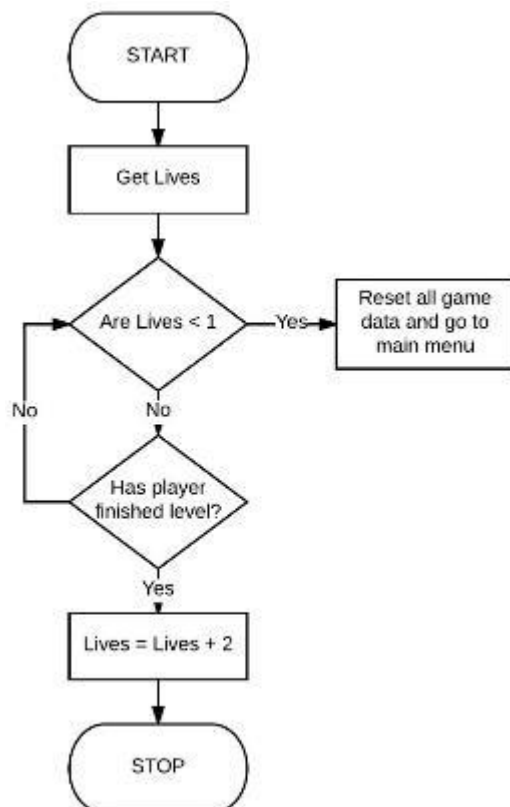
## ENEMY SHOOTING COLLISIONS

---

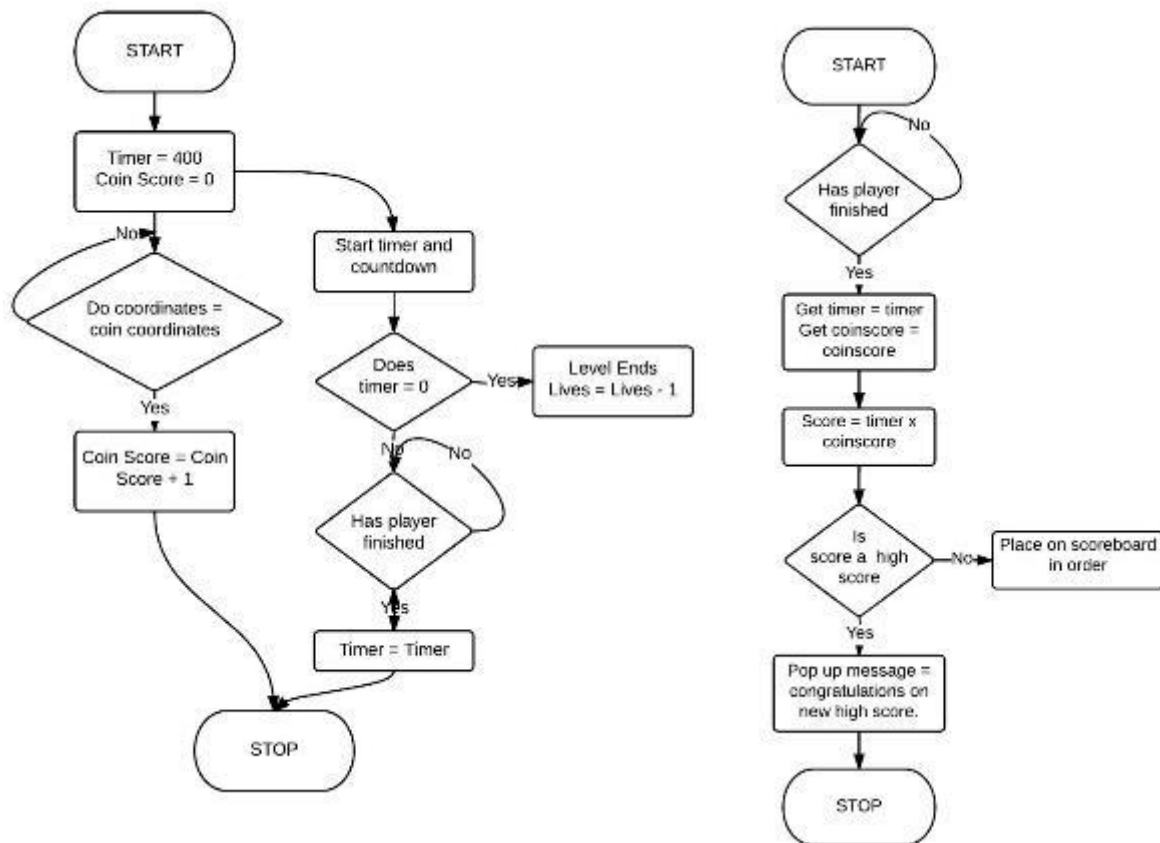


## LIVES SYSTEM

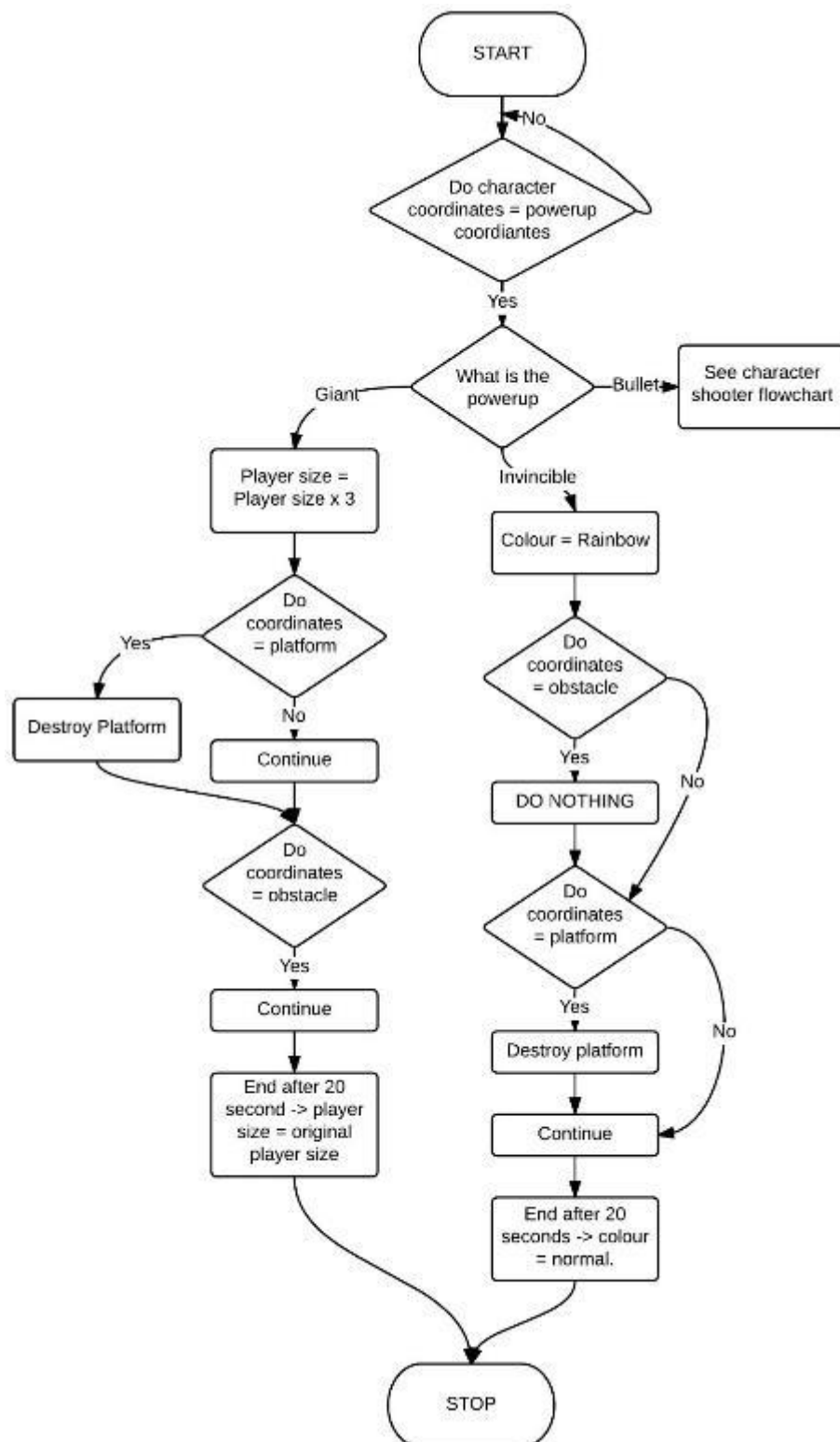
---



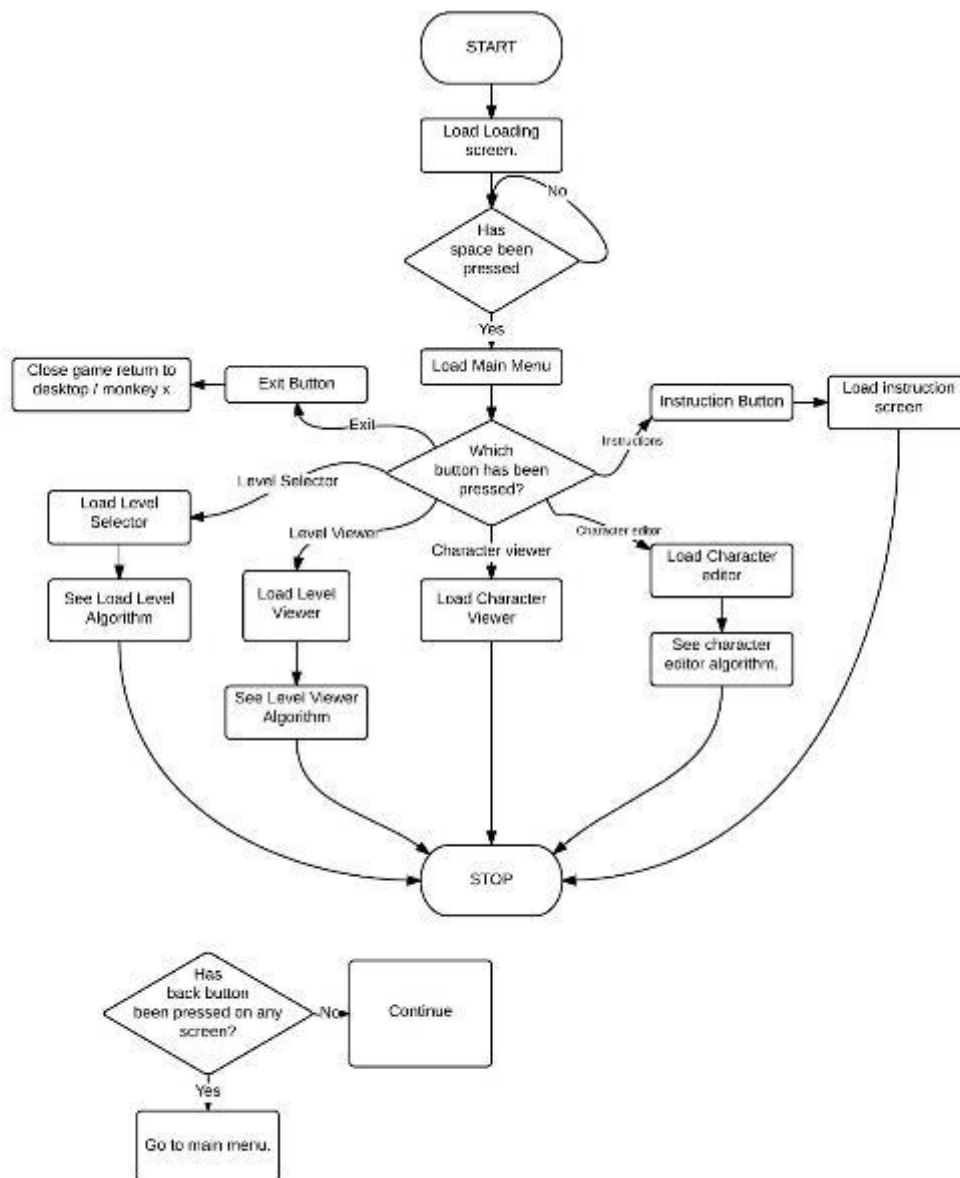
## SCORING SYSTEM



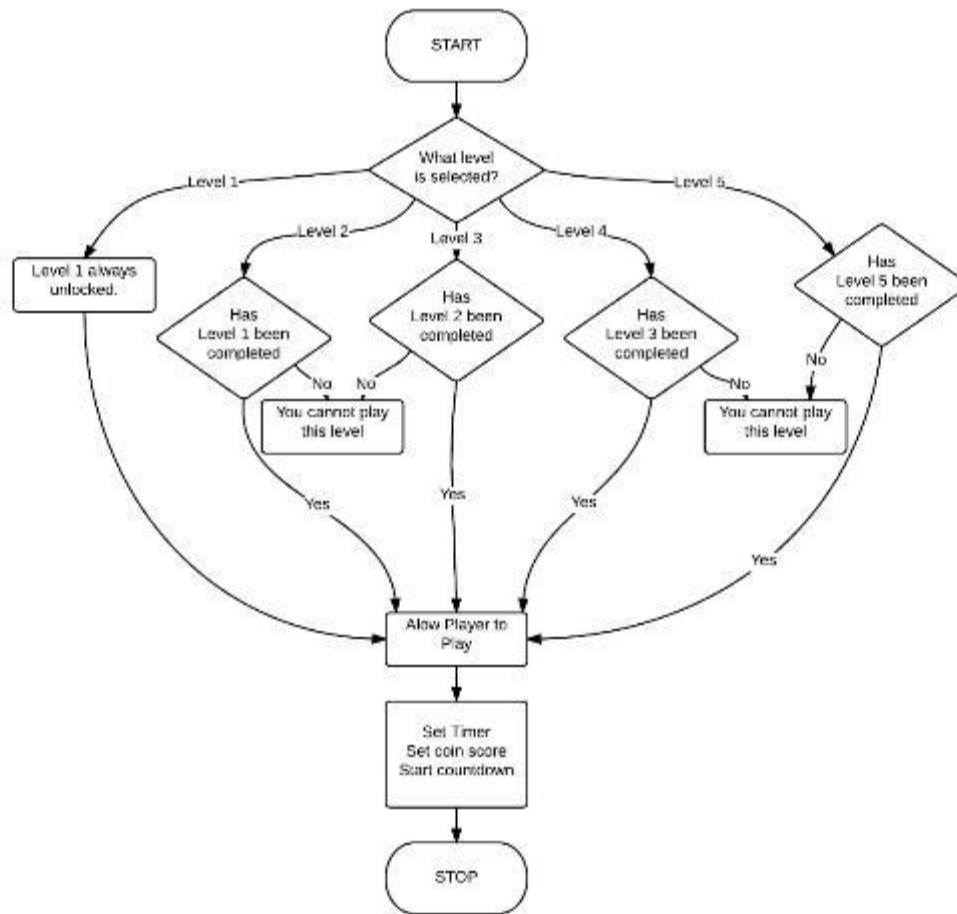
## POWER - UPS



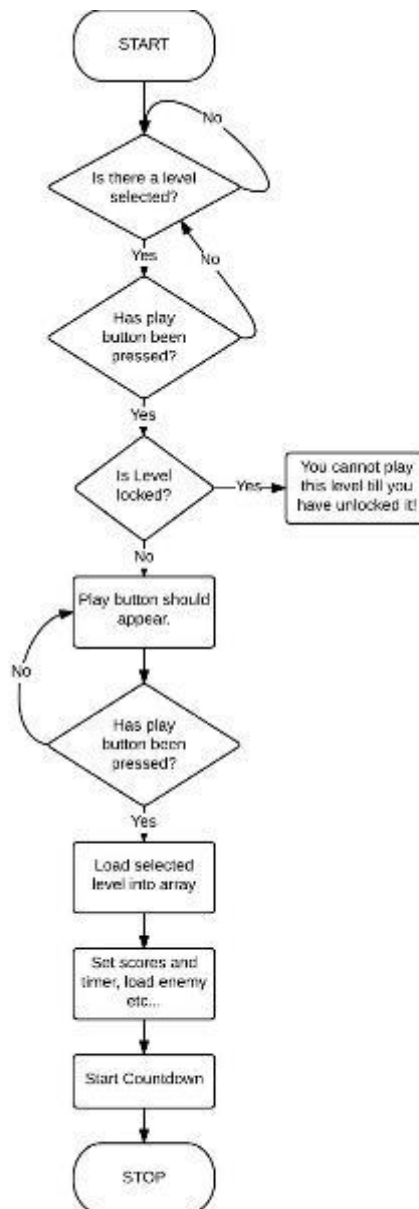
## MENU SYSTEM



## LEVEL LOADER VERIFICATION



## IN DEPTH LEVEL VALIDATION

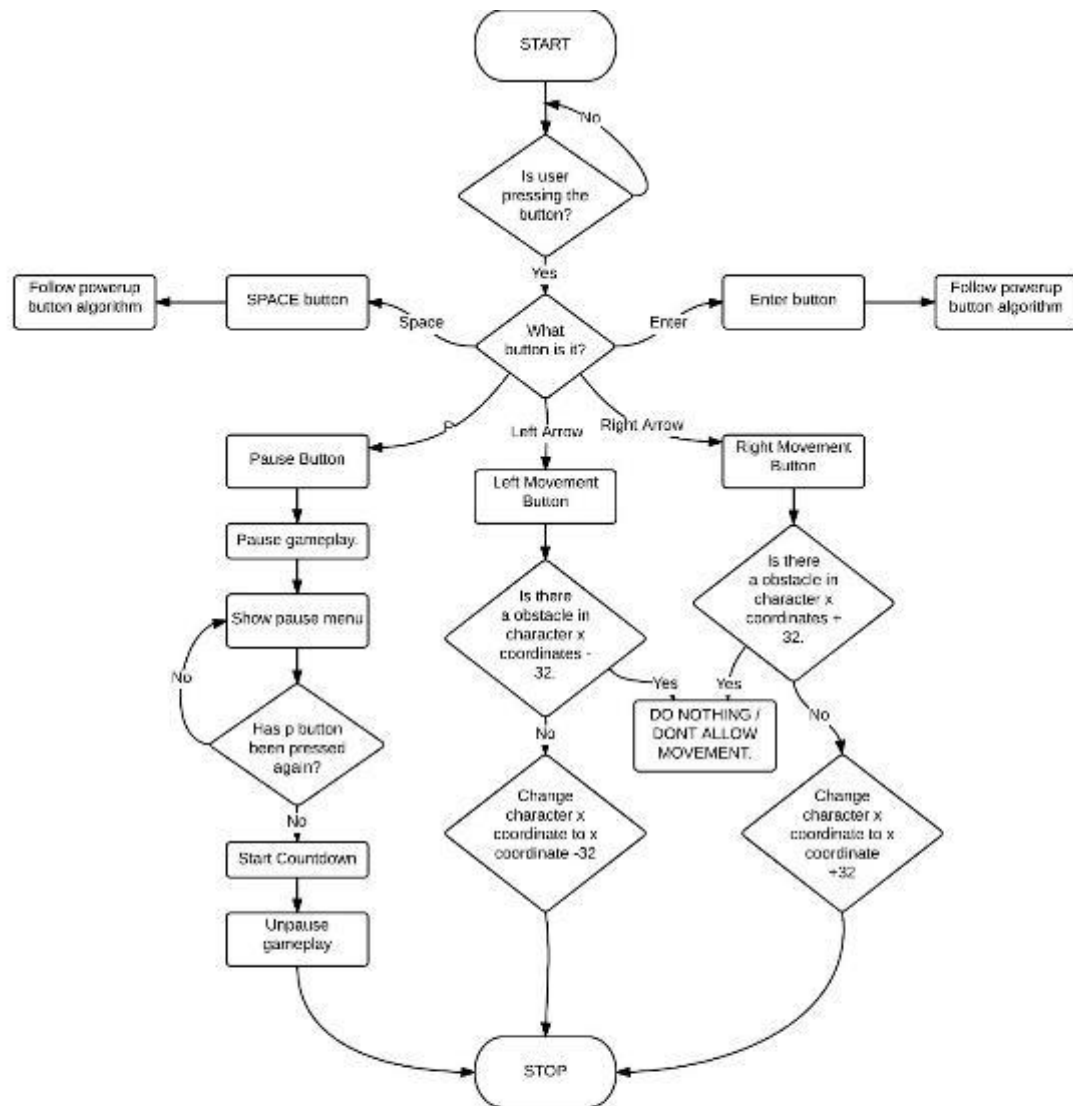


## LEVEL VALIDATION EXPLAINED

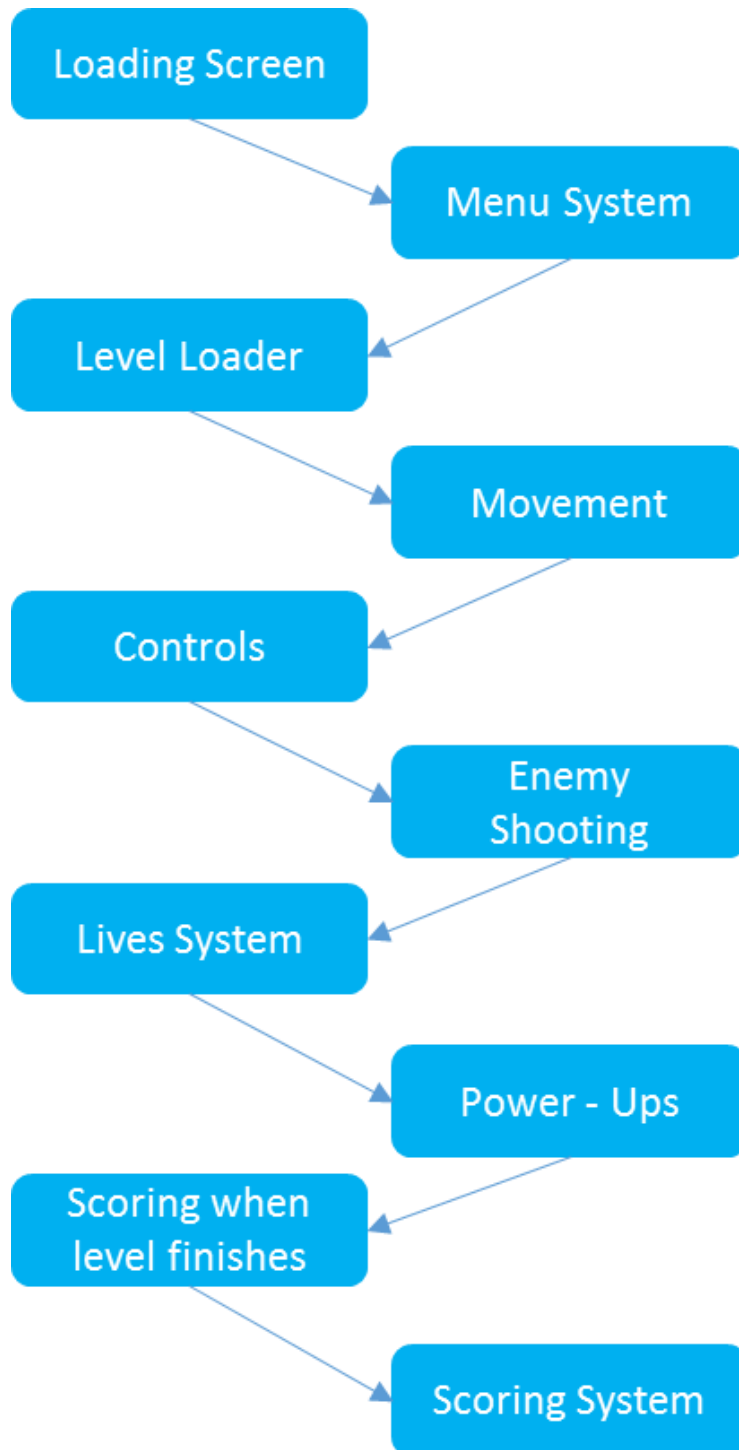
I will need to have some level validation which either stops or allows the player to play a certain selected level, my game works by having 1 unlocked level (level 1) the player then progresses through the game by unlocking the next levels until they have unlocked and completed every level. This therefore means there must be some form of validation in place throughout the game to stop the player from playing a level they are yet to have unlocked. I have decided to make it so that it works through checking to see whether the level before has been completed, if the level before hasn't been completed the next level can't be unlocked. However if the level before has been completed by which it will have a high score then the level directly after should become unlocked so the player can play it and so on.



## CONTROLS

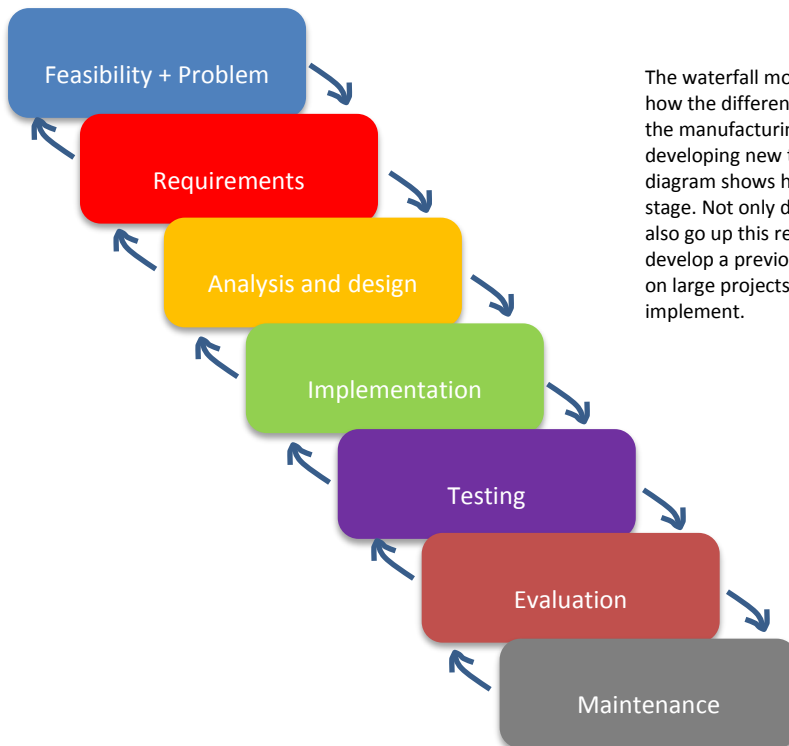


## HOW DO THE ALGORITHMS LINK?



## TRADITIONAL SOFTWARE DEVELOPMENT LIFE CYCLE

### WATERFALL



The waterfall model is called the waterfall model because of how the different stages are represented. It is mostly used in the manufacturing and construction industries as a way of developing new things and not necessarily new software. The diagram shows how the program flows down through each stage. Not only do the arrows go down the model but they also go up this represents how programmers often have to develop a previous stage. The waterfall model is often used on large projects because it is easy to understand and implement.

### AGILE



Concentrates on the fact that the requirements will change frequently whilst the program is made. The program is made in an iterative way. This means that with each iteration the requirements change or there are more requirements. This also means the program can constantly be shown to the user who can change what they want. However with this method the price might increase constantly as the requirements increase.

### HOW WILL THE GAME BE DEVELOPED?

With most of the development lifecycles they all start with analysis and design, the game will follow this development process however it will follow an agile development cycle. This works by coming up with the backbones for your idea and brainstorming (analysis) and then designing game prototypes such as screen designs (design). Agile has an emphasis on game iterations which will be best for my game, an procedure or iteration is developed for the game, it is then checked for bugs and if there are errors then a new iteration is coded. This development methodology will be the best for my game, it will enable me to produce bug free thoroughly thought through code.

---

**TEST DATA FOR DEVELOPMENT**

The program will be tested as I go along, when I program a new thing I will test it if possible to see whether what I have programmed has worked, this means the program will be tested as it is developed.

**Movement**

| Test Data     | Type    |
|---------------|---------|
| 'Left Arrow'  | Valid   |
| 'Right Arrow' | Valid   |
| 'Spacebar'    | Valid   |
| 'Enter'       | Valid   |
| 'P'           | Valid   |
| 'W'           | Invalid |
| 'A'           | Invalid |
| 'S'           | Invalid |
| 'D'           | Invalid |

**Power-ups**

| Test Data     | Type    |
|---------------|---------|
| 'Left Arrow'  | Invalid |
| 'Right Arrow' | Invalid |
| 'Spacebar'    | Invalid |
| 'Enter'       | Valid   |
| 'P'           | Invalid |
| 'W'           | Invalid |
| 'A'           | Invalid |
| 'S'           | Invalid |
| 'D'           | Invalid |

**Menu Controls**

| Test Data     | Type    |
|---------------|---------|
| 'Left Arrow'  | Valid   |
| 'Right Arrow' | Valid   |
| 'Spacebar'    | Invalid |
| 'Enter'       | Valid   |
| 'P'           | Invalid |
| 'W'           | Invalid |
| 'A'           | Invalid |
| 'S'           | Invalid |
| 'D'           | Invalid |

## TEST DATA FOR BETA TESTING

I will be testing my game as it is developed, so that when I code something new and get to a complete section of my program that I can run it and see if it works. This therefore doesn't really follow a testing approach. However I will follow a black box approach to testing the final beta, final beta is when a program is tested in regards to the specification

## ENEMIES AND CHARACTERS

| What is the test for?                                               | Explanation                                                                                                                                   | Expected Outcome                                                               |
|---------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| Enemies should appear.                                              | Enemies will appear where declared in level code, harder levels will have more enemies.                                                       | Enemy should appear.                                                           |
| Character does not fall through the floor.                          | Character should not fall through floor when they move.                                                                                       | Character will stay fully above the platform.                                  |
| Character shouldn't fall through if they land on elevated platform. | If character jumps and lands on one of the elevated boxes / platforms they shouldn't fall through, should only fall if nothing is below them. | Character should stay in position on platform.                                 |
| Enemy shooting bullets                                              | Enemy should shoot bullets horizontally from current position at a fire rate of a bullet every 2 seconds.                                     | Enemy shoots bullet                                                            |
| Character collisions with enemy bullet.                             | If character in collision with bullet, then level ends. Unless giant and invincible power up detected.                                        | If collision detected between bullet and character level ends and lose a life. |
| Character falling in obstacle (water or fire)                       | If character falls into obstacle, then level ends. Unless giant or invincible power up detected.                                              | Level ends and player loses a life if character falls into an obstacle.        |

## MENUS

| What is the test for?                   | Explanation                                                                 | Expected Outcome                                       |
|-----------------------------------------|-----------------------------------------------------------------------------|--------------------------------------------------------|
| Game starts and loading screen appears. | The loading screen for the game should appear when the game is compiled.    | Loading screen appears.                                |
| Main Menu appears                       | Main menu should appear when the space button on loading screen is pressed. | Main Menu with options appears.                        |
| User can navigate menu                  | User should be able to use arrow keys to navigate menus.                    | Arrow keys allow different options should be selected. |

|                                         |                                                                                                                    |                                                         |
|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| Red box around current selected option. | When the arrow keys are used to select different options then a red box should appear around current selected one. | Red box around currently selected option.               |
| Exit button should close the game       | The exit button on the main menu should close the game.                                                            | The exit button on the main menu should close the game. |
| Display level selector screen.          | Clicking enter on level selector button should display the level selector screen.                                  | Level selector screen appears.                          |
| Display level viewer screen             | Clicking enter on level viewer button should display the level viewer screen.                                      | Level viewer screen appears.                            |
| Display character selector              | Clicking enter on character selector button should display character selector screen.                              | Character selector screen appears.                      |
| Display character editor                | Clicking enter on character editor button should display character editor screen.                                  | Character editor screen appears.                        |
| Display instructions                    | Clicking enter on instructions button should display the instructions screen.                                      | Instructions screen appears.                            |
| Back Button                             | The back button on all of these screens should go back to the main menu.                                           | Back button goes back to main menu.                     |

## LEVELS

| What is the test for?                            | Explanation                                                                              | Expected Outcome                                                    |
|--------------------------------------------------|------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| Level selector shows 5 levels, 2 – 5 are locked. | When the game starts the first level should be unlocked and the others should be locked. | Level 1 unlocked<br>Level 2 to 5 locked                             |
| Selected locked levels results in error.         | It should not be possible to play any of the locked levels.                              | Clicking locked level should do nothing, level shouldn't be loaded. |
| Clicking level 1 should load level.              | Level 1 which is unlocked when game starts should load.                                  | Level 1 loaded into array.                                          |

## POWER-UPS, TIMER, COINS

| What is the test for?                                     | Explanation                                                                                                                                       | Expected Outcome                                                                    |
|-----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Coins are loaded and appear randomly.                     | Coins should appear in random places on base platforms and floating platforms. (Around 100 will appear).                                          | Coins appear when level starts                                                      |
| Coins should disappear when they are collected.           | Coins should disappear when character goes through them to stop them being collected again.                                                       | Coin should disappear.                                                              |
| Power ups should be loaded and appear randomly.           | Power ups should appear in random places on the level, should be a level of difficulty collecting them.                                           | Power ups appear.                                                                   |
| Power up should disappear when collected                  | The power ups should disappear when the character goes through one and collects it.                                                               | Power ups should disappear.                                                         |
| Timer set to 400                                          | Timer should count down from 400 (1 = 1 second).                                                                                                  | Timer set to 400 and counts down.                                                   |
| Giant power up increases size of the character.           | When giant power up collected and used the character's size will increase.                                                                        | Character becomes giant with giant power up.                                        |
| Invincible power up makes character invincible.           | When invincible power up collected and used the characters colour changes and player will be immune to all ways of dying.                         | Character becomes invincible.                                                       |
| Shooter power up allows character to shoot bullets.       | When shooter power up collected and used the character can shoot bullets by pressing enter button. At increased fire rate in comparison to enemy. | Character is able to shoot bullets by pressing enter.                               |
| Cannot collect more power ups if one is stored or in use. | If the character has a power up or is using one then they shouldn't be able to collect, character should go straight through it.                  | Character cannot collect another power up if they have one to use or are using one. |
| Power up lasts for 20 seconds when collected.             | When the power up is collected it will last for 20 seconds giving the character the extra power.                                                  | Power up lasts for 20 seconds before ending.                                        |
| Coin collected                                            | When a coin is collected validly then coin score incremented by 1.                                                                                | Coin collected = Coin Score + 1                                                     |

|                                                                  |                                                                                                                                                                |                                                                         |
|------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| Timer runs out then level should end.                            | When the timer runs out and the person has not completed the level then level will be kept as uncompleted, it will go back to level selector and live is lost. | Timer runs out level ends and life is lost.                             |
| Timer and coins should be displayed changing in the scoring bar. | The scoring bar should update to show the coin score and the time left (counting down).                                                                        | The coin score and timer should update in the score bar.                |
| Finishing level, timer stops and score calculated.               | When the player completes the level then timer stops and overall score is calculated with calculation -> score = timer x coin score.                           | Timer stops when level finishes and overall score correctly calculated. |
| Scoreboard updated                                               | The scoreboard should be updated once a level is correctly completed. Scores will need to be compared to see which is biggest.                                 | Scoreboard correctly updated when level completed.                      |
| Player gets 2 lives if level completed.                          | If the level is completed within time, then the player gets two lives.                                                                                         | Player gets two extra lives and it should be added to live count.       |

## SOUNDS

| What is the test for?                                          | Explanation                                                                   | Expected Outcome                                             |
|----------------------------------------------------------------|-------------------------------------------------------------------------------|--------------------------------------------------------------|
| When player jumps there should be a jumping sound.             | As requested by Robbie there will be a sound for jumping.                     | Sound plays when character jumps (presses spacebar).         |
| When character shoots bullet there should be a shooting sound. | As requested by Robbie shooting with the bullet power up should have a sound. | Sound plays when enter button pressed with shooter power up. |

## MOVEMENT / CONTROLS

| What is the test for?                     | Explanation                                                                                         | Expected Outcome                          |
|-------------------------------------------|-----------------------------------------------------------------------------------------------------|-------------------------------------------|
| Left arrow moves character to the left.   | Left arrow button moves character a tile to the left as long as there is no obstacle like a wall.   | Left arrow moves character to the left.   |
| Right arrow moves character to the right. | Right arrow button moves character a tile to the right as long as there is no obstacle like a wall. | Right arrow moves character to the right. |



Candidate Name: [REDACTED]

Candidate Number: [REDACTED]

|                                                  |                                                                       |                                                        |
|--------------------------------------------------|-----------------------------------------------------------------------|--------------------------------------------------------|
| Jump button (spacebar) causes character to jump. | The jump button is pressed characters' y coordinates should increase. | The character will jump and fall if no platform below. |
|--------------------------------------------------|-----------------------------------------------------------------------|--------------------------------------------------------|

---

## SIGN OFF PROPOSAL

## INTERVIEW WITH CLIENT

---

Dan

Robbie

Q. Are you happy with what has currently been shown in terms of the menus and screen designs?

A. Yes the current screen designs are very good, so I'm happy with that.

Q. I can't directly copy the screen when I go to code the game in monkey x but will try and keep it as close to the screen designs that I have at the moment.

A. Yes that is fine, I would like to see these designs and actual screens as we go along.

Q. Is there anything else that you would like to be added to the screens?

A. Nothing that I can think off but I wouldn't mind adding ideas later on.

## SIGNATURE OF APPROVAL

---

[REDACTED]

Robbie Needham

## C. DEVELOPMENT

### 07/12/2016 INTRODUCTION

I had all my requirements outlined so that I knew exactly what the game had to offer when I started programming and the features I would have to think ahead and understand how to incorporate. Because I was new to the Monkey X programming language and TED IDE I decided the best step in order to start my game was to better understand the language – I did this by looking at some pre made (bare bones) Monkey X programs. This will be a good way of giving me an insight into the structure, functions, methods, subroutines I could expect to use.

### 10/12/2016 LOOKING AT MONKEY X PROGRAMS

*Single comment used for making comments*

*Used to import a library of functions, lightweight framework of routines.*

*This is where the program starts.*

*Class game-app is where the main routines for the game operate here.*

*Field is the keyword to declare a variable that can be accessed only by the methods in the class.*

*On create is called once when the program first runs.*

*Menu is loaded with the image menu.png from the shocker class folder.*

*A class describes an object in the program, a method is what the object can do.*

*This is a global variable*

*A variable called game is created that will hold an instance of the game.*

*It assigns the variable game to be a new instance of a class called game-app effectively starting the game.*

*Keyword for inheritance*

*Game-app contains all the methods from the app class that is part of the mojo framework.*

*Will have the image data type.*

*This is good place to put all the initial object declarations, data structures and loading images.*

*sets the frame rate to 60 frames per second (FPS)*

```

'Libraries and globals
Import mojo
Global Game:Game_app

'Main program starts here:
Function Main ()
 Game = New Game_app
End

'All game code goes here:
Class Game_app Extends App
 Field menu:Image
 Global GameState:String = "MENU"
 Global board:Int[26]
 Field difficulty:5
 Field puzzle:Image
 Field square:Image
 Field square_selected:Image
 Field solved_pic:Image
 Field tile:Image[25]
 Field selected:Int = -1

 Method OnCreate ()
 'All the initialisation for the game goes here:
 SetUpdateRate 60
 menu = LoadImage ("menu.png")
 Seed = Milliseconds()
 puzzle = LoadImage("puzzle1.png")
 square = LoadImage("square.png")
 square_selected = LoadImage("selected.png")
 solved_pic = LoadImage("solved.png")
 Local x:Int = 0
 Local y:Int = 0
 For Local num:Int = 0 To 24
 tile[num] = puzzle.GrabImage(x,y,100,100)
 board[num] = num
 x=x+100
 If x = 500 Then
 x=0
 y=y+100
 End
 Next
 End
End

```

## 22/12/2016 CREATING A GENERAL STRUCTURE FOR MY GAME

---

The first part is creating the global variable for the game, it is called 'MazeRunner' which the identifier and then the variable type is declared as 'Game' which is its type.

```
Global MazeRunner:Game

Function Main()
 MazeRunner = New Game
End
```

The next part is the 'Function Main()' this is where the actual game runs, without this the game would do nothing, 'MazeRunner = New Game' simply starts a new game which is the maze runner.

```
Class Game Extends App

End
```

This is the first stages of object orientated programming – using classes. This class will hold methods; a class is basically a blueprint for an object stating what its characteristics will be like. This will be the main class for my game and will be where all of the objects are drawn and methods used.

```
Global MazeRunner:Game

Function Main()
 MazeRunner = New Game
End

Class Game Extends App

 Method OnCreate()

 End

 Method OnUpdate()

 End

 Method OnRender()

 End

End
```

The next parts of a normal Monkey X program are the three methods; 'OnCreate' are where the object declarations are and are how objects are created in the program. 'OnUpdate' is where all the game logic is situated such as what happens when certain buttons are pressed for instance. Finally, 'OnRender' is where all the objects are drawn to the screen and how they appear.

The next part that I will be trying to code is coding all of the screens, they will have simplified navigation between them and will just be blank screens to show how the game will function in regards to the menus.

```

Class Game Extends App

 Method OnCreate()

 SetUpdateRate 60

 End

 Method OnUpdate()
 Select GameState
 Case "LOADING_SCREEN"

 Case "MENU"

 Case "LEVEL_SELECTOR"

 Case "LEVEL_VIEWER"

 Case "CHARACTER_VIEWER"

 Case "CHARACTER_EDITOR"

 Case "INSTRUCTIONS"
 End
 End

 Method OnRender()
 Select GameState
 Case "LOADING_SCREEN"

 Case "MENU"

 Case "LEVEL_SELECTOR"

 Case "LEVEL_VIEWER"

 Case "CHARACTER_VIEWER"

 Case "CHARACTER_EDITOR"

 Case "INSTRUCTIONS"
 End
 End
End

```

So far I have only added the set update rate line so far to the on create method, this simply sets the update rate to 60 fps. However, I have added the different game states to both on update and on render methods, these cases are basically all the screens there will be, so in total there will be 7 different screens excluding levels. On update will basically have what happens when I click keys on certain pages for instance which is my next step for programming the general structure of my game. On render for the moment will contain all that is drawn on the screens, for instance text and back ground colours.

```

Method OnUpdate()
 Select GameState
 Case "LOADING_SCREEN"
 If KeyHit (KEY_SPACE) Then GameState = "MENU"

 Case "MENU"
 If KeyDown (KEY_I) Then GameState = "LEVEL_SELECTOR"
 If KeyDown (KEY_V) Then GameState = "LEVEL_VIEWER"
 If KeyDown (KEY_C) Then GameState = "CHARACTER_VIEWER"
 If KeyDown (KEY_E) Then GameState = "CHARACTER_EDITOR"
 If KeyDown (KEY_I) Then GameState = "INSTRUCTIONS"

 Case "LEVEL_SELECTOR"
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"
 Case "LEVEL_VIEWER"
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"
 Case "CHARACTER_VIEWER"
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"
 Case "CHARACTER_EDITOR"
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"
 Case "INSTRUCTIONS"
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"
 End
End

```

Here is the current navigation for my game, this will not be how it is done on the final game but I have decided to simplify it to start with to concentrate more on the bigger parts of the game. Ideally it will be done by the user being able to click on the text options on the main menu. At the moment in the current build the screens are controlled by pressing certain keys, for instance on the loading screen pressing the spacebar will result in the game state changing to the menu, here there are 5 different choices for what will happen and I have

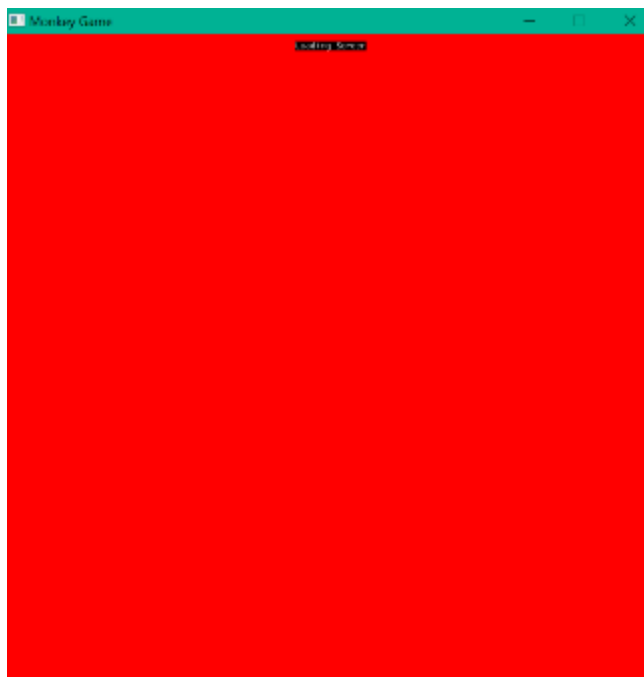
decided to control this through letters, e.g pressing the I key will change the game state to instructions as specified earlier this is only to simply navigation to start with. With all of these screens pressing enter will take you back to the main menu where you can select a different page. At the moment the game will just display black screens so you won't be able to tell you are actually navigating between the pages, this is the next thing I need to implement.

```

Method OnRender ()
 Select GameState
 Case "LOADING_SCREEN"
 Cls (255,0,0)
 DrawText("Loading Screen", 450, 10, 0.5)
 Case "MENU"
 Cls (0,255,0)
 DrawText("Menu", 450, 10, 0.5)
 DrawText("Press L for level selector", 450, 410, 0.5)
 DrawText("Press V for level viewer", 450, 430, 0.5)
 DrawText("Press C for character viewer", 450, 450, 0.5)
 DrawText("Press E for character editor", 450, 470, 0.5)
 DrawText("Press I for instruction", 450, 490, 0.5)
 Case "LEVEL_SELECTOR"
 Cls (0,0,255)
 DrawText("Level Selector", 450, 10, 0.5)
 Case "LEVEL_VIEWER"
 Cls (0,0,255)
 DrawText("Level Viewer", 450, 10, 0.5)
 Case "CHARACTER_VIEWER"
 Cls (0,0,255)
 DrawText("Character Viewer", 450, 10, 0.5)
 Case "CHARACTER_EDITOR"
 Cls (0,0,255)
 DrawText("Character Editor", 450, 10, 0.5)
 Case "INSTRUCTIONS"
 Cls (0,0,255)
 DrawText("Instructions", 450, 10, 0.5)
 End
End

```

All of the code for this has been put in the on render method, I have added titles to all of the pages as well as instructions for moving between the pages with code such as 'DrawText("Character Viewer", 450, 10, 0.5)'. Draw text draws a standard monkey x text to the screen, it takes 4 parameters – the actual text, the x position which I have set for the middle of the page, the y position which is 10 of the top and 0.5 centers it at the x point so it will be in the middle of my 900 width page. In each of the game states I have used the Cls command which changes the background colour, I have made the background colour red for the loading screen, green for the menu screen and blue for the other screens.

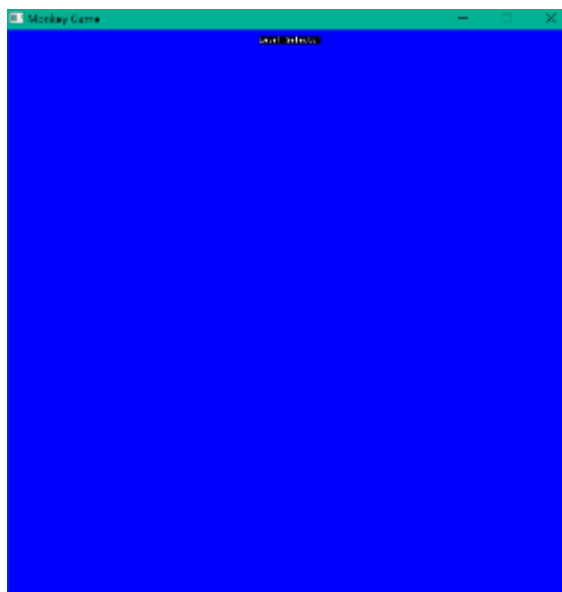
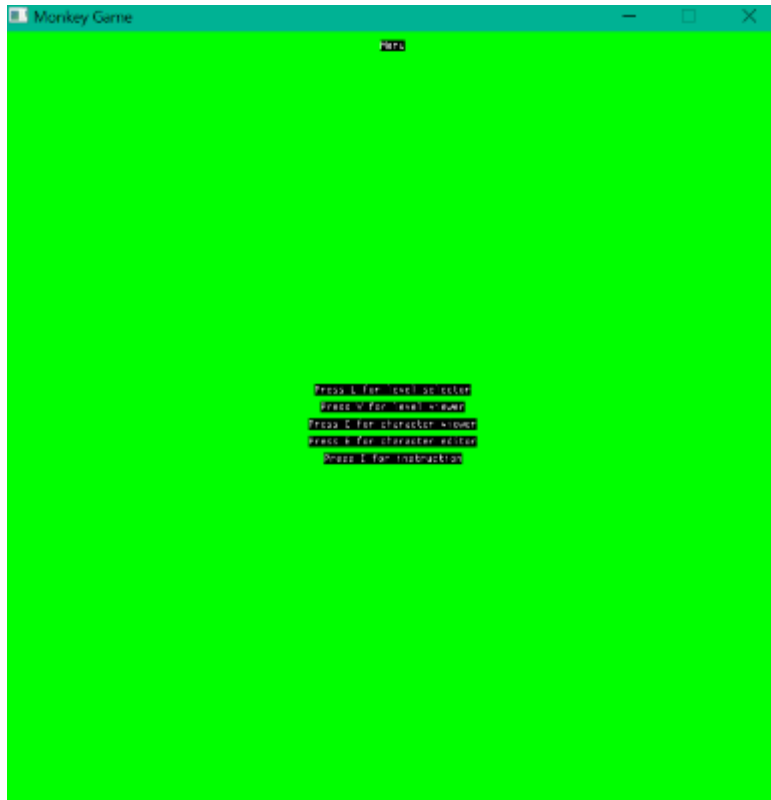


When the game is run, this is the first screen that is shown, it will have my loading screen.



Candidate Name: [REDACTED]


Candidate Number: [REDACTED]



Example of one of the blue screens, pressing escape will go back to menu, where you can choose one of the other

## 23/12/2016 ADDING GRAPHICS TO THE LOADING SCREEN

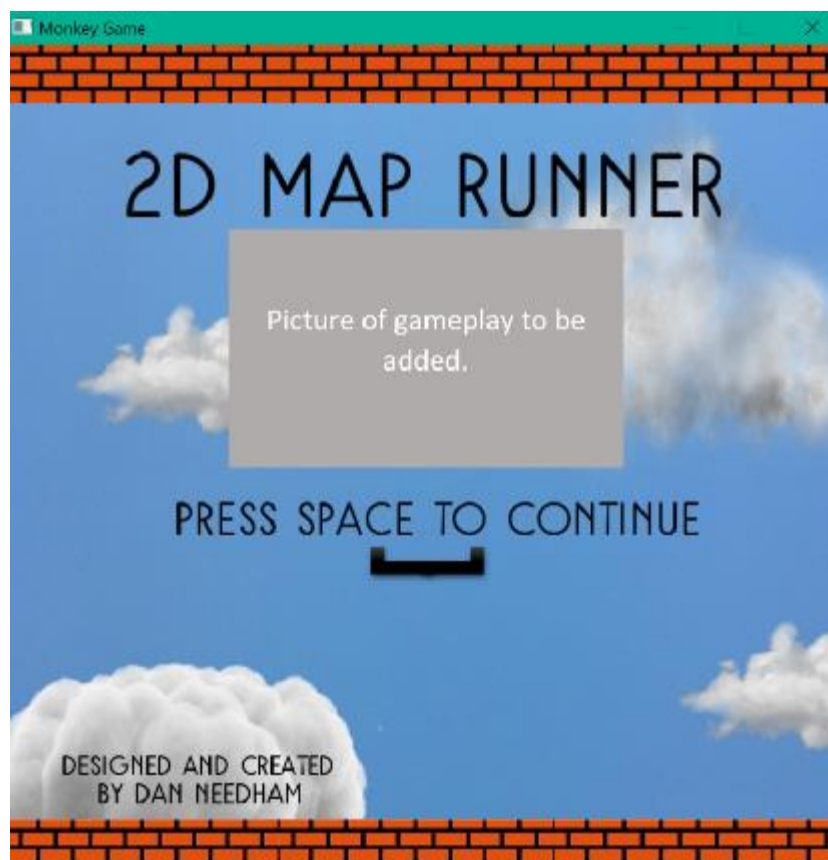
My loading screen graphics will be really simple, it will just be a picture and then the player can press enter to move onto the next page, no interactions are required. The first stage of this is to add the actual image that will be shown to the data file next to the actual program, this is an image I designed myself.

| KINGSTON (D:) > A2 > Computing > Computing Project > Starting project > MazeRunner > MazeRunner.data |                  |          |          |
|------------------------------------------------------------------------------------------------------|------------------|----------|----------|
| Name                                                                                                 | Date modified    | Type     | Size     |
|  loadingscreen      | 31/12/2016 13:32 | PNG File | 1,177 KB |

In the main game class, you create a variable, this calls the variable loadingscreen and declares its type as an Image.

```
Class Game_app Extends App
 Global GameState:String = "LOADING_SCREEN"
 Field loadingscreen:Image
```

In the on create method you simply add the code 'loadingscreen = LoadImage ("loadingscreen.png")', this simply initialises the loadingscreen variable or image so that when the loadingscreen is called it is ready to displayed. Finally in the on render method I added the code 'DrawImage loadingscreen, 0,0' which draws the image loadingscreen at the coordinates 0,0; which looks like this...



## 24/12/2016 TESTING THE MENUS

### On the loading screen...

| Input    | Output                               |
|----------|--------------------------------------|
| Spacebar | Moves to the menu screen as expected |
| 'L'      | Null                                 |
| 'V'      | Null                                 |
| 'C'      | Null                                 |
| 'E'      | Null                                 |
| 'I'      | Null                                 |

### On the menu screen...

| Input         | Output                                             |
|---------------|----------------------------------------------------|
| 'Left Arrow'  | Null                                               |
| 'Right Arrow' | Null                                               |
| 'Spacebar'    | Null                                               |
| 'Enter'       | Null                                               |
| 'P'           | Null                                               |
| 'W'           | Null                                               |
| 'A'           | Null                                               |
| 'S'           | Null                                               |
| 'D'           | Null                                               |
| 'L'           | Moves to the level selector screen as expected     |
| 'V'           | Moves to the level viewer screen as expected       |
| 'C'           | Moves to the character selector screen as expected |
| 'E'           | Moves to the character viewer screen as expected   |
| 'I'           | Moves to the instruction screen as expected        |
| 'Esc'         | Null                                               |

## 28/12/2016 LEVEL 1 GRAPHICS

We first start by creating another class, this will be my level 1 class where all of the graphical code and creating code goes for the level. I had spent ages trying to perfect this class, and had problems consistently trying to implement the levels and graphics into the game. I already knew that my game would use a tile based system where maps are created using tiles to make up a whole map, I found it hard to understand how this worked and how it broke a jpg with tiles in to determine which tile would be drawn. However, I found the game maze from Craig 'n' Dave which has the system implemented into the game so I analysed it to better understand the system and how it worked.



29/12/2016 MAZE GAME ANALYSIS

---

```

Class Level
 Field tiles:String[21][]
 Field tileset:Image

 Method New()
 tileset = LoadImage ("tiles.png",32,32,3)
 For Local i:Int = 0 To 20
 tiles[i] = New String[16]
 Next
 End

 Method load()
 Local level_file:FileStream
 Local level_data:String
 Local data_item:String[]

 level_file = FileStream.Open("monkey://data/maze.txt","r")
 level_data = level_file.ReadString()
 level_file.Close

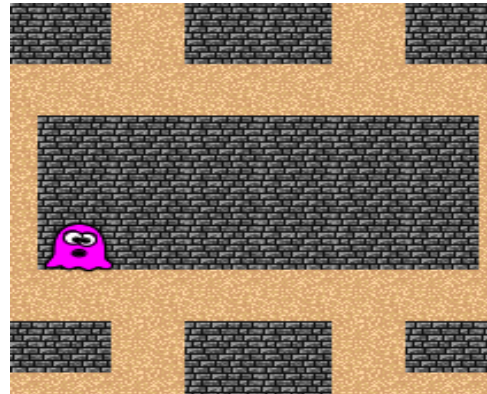
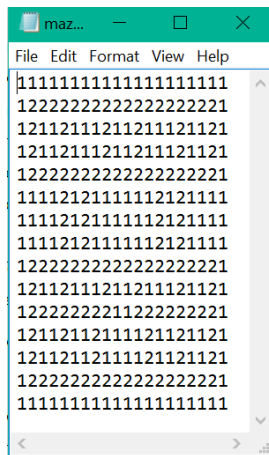
 data_item = level_data.Split("~n")
 For Local y:Int = 0 To 14
 For Local x:Int = 0 To 19
 tiles[x][y]=Int(data_item[y][x..x+1])
 Next
 Next
 End

 Method draw()
 Local tile:String
 For Local y:Int = 0 To 14
 For Local x:Int = 0 To 19
 tile = tiles[x][y]
 If tile = Wall Then DrawImage tileset, x*32, y*32, Wall
 If tile = Grass Then DrawImage tileset, x*32, y*32, Grass
 Next
 Next
 End

```

This is the section of code responsible for taking in the tile map and breaking it down into separate tiles which can be used to draw the map. Tile set is the tiles that will be drawn depending on the text file, in reality we are drawing a map that is 20 tiles wide and 15 tiles high. But this isn't so simple, for Monkey-X we do this by creating an array making it 20 tiles by 15 a 2 dimensional array. We load the tileset with the code 'tileset = LoadImage ("tiles.png",32,32,3)'; 32, 32 is the size of the tile (32 pixels by 32 pixels) and 3 is how many tiles there are in the png file. We use our for loop to initialise our array for the world so that it is 20x15, this is all inside our method new which is what is done when a new instance of the level class is created.

The load method is essentially responsible for sorting out the array and loading the data into the array, there are three main parts to it; 'level\_file' this is the actual data file and helps connect the text file, 'level\_data' is the contents of the data file consisting of our integers, finally 'data\_item' is one row of tiles. The aim is to convert the tiles stored as strings to integers, the result of this manipulation technique is used for the draw method: 'tile = tiles[x][y]'. This works along with the constants declared at the beginning of the program: 'Const Wall = 1' and 'Const Grass = 2'. It basically looks at the text file and if there's a 1 it draws a wall tile and if theres a 2 it draws the grass tile.



As you can see the data file corresponds to what is drawn to the screen, we can change the data file to display a different map and allows us to customize things. For my game the map will be bigger and I will also have more tiles.

### 03/01/2017 STARTING MY LEVEL CLASS

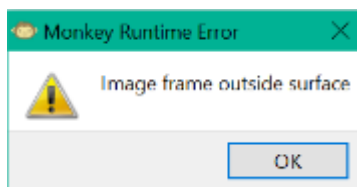
I decided that the best idea would be to create another program before implementing this into my main program, I could perfect my level class and then add it to my game, it would also make it much easier to make more levels. I first off started my importing the two monkey modules, mojo responsible for graphics and brl responsible for data and text files.

Before I had even started to create my game I spent quite a while using tiles and getting to understand them and how they worked, however this resulted in consistent errors as explained below. My first resulted when I was adapting the tile program from Craig 'n' Dave in the hope it would teach me how tile programs work as explained earlier.



*'TRANS FAILED: Error updating target project – can't find block end tag 'TRANSCODE'*

This was one of the easiest errors to solve, I didn't completely understand as to what the error was but after researching the error on the Monkey-X forums I found that a simple solve would be to delete the build file and then re-run the program which would create a new build file and solve the problem which it did.



*'Image frame outside surface'*

This was another error that didn't really hinder the development of my game because it was a very easy fix. It resulted when I tried using and creating my own tile png which I could use to make my own custom maps, the tile image which I created myself using word has overlapping tiles which resulted in the image frame being

outside the surface, despite not solving this problem I decided to use a proper tile map creator for my actual game which later on worked.

#### *'Array index out of range'*

This was another error which resulted from dodgy code to implement my tile png, when declaring the size of my tile map I made the array too big not taking into consideration that an array starts at 0. This was easily solved and furthermore I didn't make this mistake when programming my level 1.

### 04/01/2017 CONTINUED DEVELOPMENT OF LEVEL 1 CLASS

---

As stated earlier I decided to start my level 1 class in another program from main game so that it would be easier to diagnose problems, it will also make it easier to create more and levels using this level class.

I first started by creating another general game structure consisting of the main game function and global variable along with my game class. As you can see this is usually the general structure for a monkey x game.

```

Import mojo
Import brl
Global Game:Game_app

Function Main ()
 Game = New Game_app
End

Class Game_app Extends App

 Method OnCreate()

 End

 Method OnUpdate()

 End

 Method OnRender()

 End

End

```

I have added the same code as from my main game program to speed up the process of developing my level.

```

Method OnCreate()
 SetUpdateRate 60
End

Method OnUpdate()
 Select GameState
 Case "MENU"
 If KeyHit (KEY_SPACE) Then GameState="LEVEL1"
 Case "LEVEL1"
 End
End

Method OnRender()
 Select GameState
 Case "MENU"
 DrawText("Menu", 320, 10, 0.5)
 DrawText("Press space for level 1", 320, 240, 0.5)
 Case "LEVEL1"
 Cls 0, 0, 0
 End
End

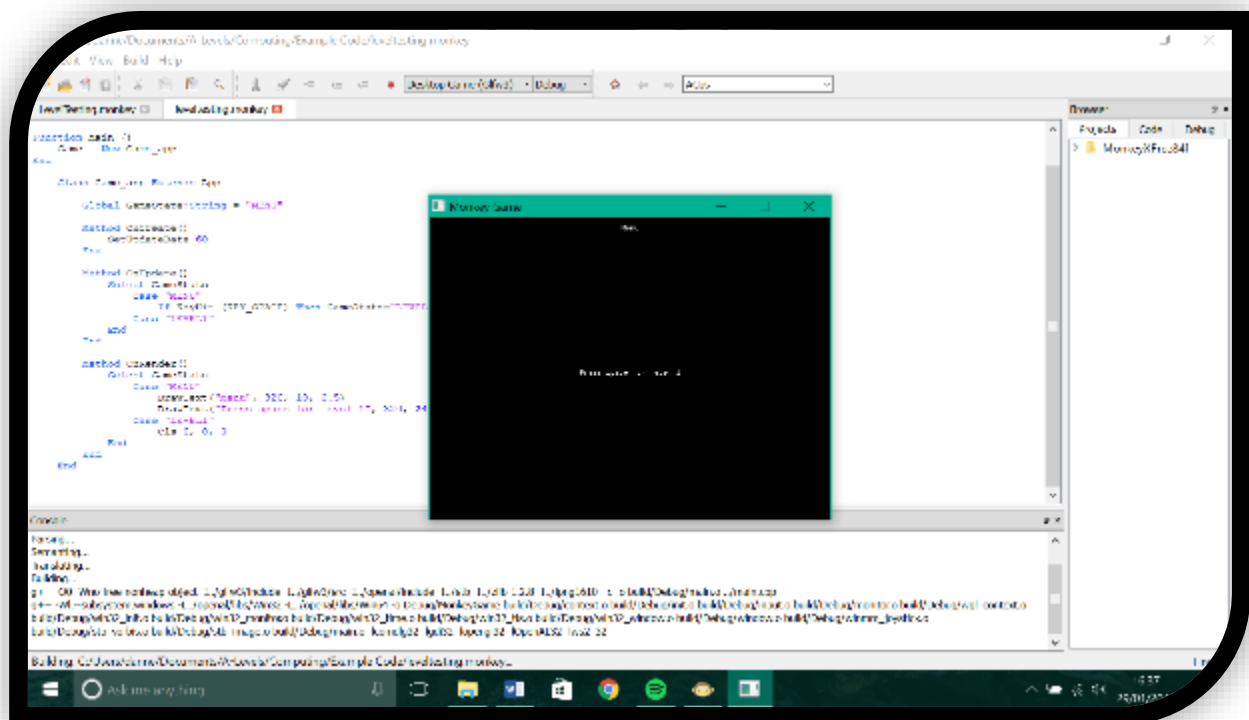
```

This has included adding basic code such as 'SetUpdateRate 60' which are all part of the general structure, I have then also created a fairly basic menu system shown above which displays the menu when the game first starts along with the title. Pressing space will then change the game state to level 1 which will then display the level 1 when I have coded it.

\_\_\_\_\_

© 2005 Blackwell Publishing Ltd

However, this section of game or this game app class is unimportant and is just a simple menu class which I will use in order to test the level game that I am coding in this program. When I have finished the level class I will implement just the level class in to my main program and add the appropriate code to make it work.



As you can see so far my level game is fairly simple consisting of just the titles and how to navigate the program, at the moment pressing space just results in a black screen because there is nothing in the 'LEVEL1' game state at the moment.

```

Class Level

 Method New()

End

Method load()

End

Method draw()

End

End

```

This will be the general structure for my level class, it will consist of three methods called new, load and draw. New is called when a new instance of the level is created, load is responsible for loading the data into array so this is where the array information is held. Finally draw is what actually draws the tiles from the array.

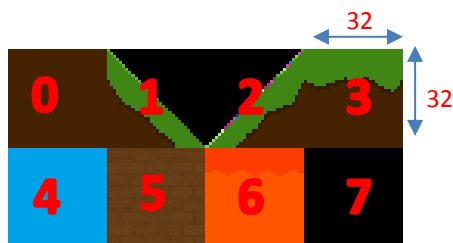
At the beginning of the class we declare two local variables – 'Field tiles:String[51][]' and 'Field tileset:Image'. The first one declares the size of the tile world so it will be 50 wide, using 51 makes things easier. I have decided that my game will be '1600x480' this means that my game will be 50 tiles wide and 15 tiles high, these will therefore be the number I will use when declaring my array.

My level class consists of two variables:

```
Field tiles:String[51][]
Field tileset:Image

Method New()
 tileset = LoadImage ("level1tilemap.png",32,32,7)
 For Local i:Int = 0 To 50
 tiles[i] = New String[16]
 Next
End
```

I have first of added this section of code into my new method, tile set is the image containing all my tiles, it loads my tile map image called "level1tilemap.png" shown below. It then declares the size of the tile as 32 x 32 and 7 is how many tiles there are in the file.



Then there is a loop which responsible for creating an initializing the array, my game will be 50 tiles by 15 tiles so an array of size 50 x 15 is made.

### 13/01/2017 - AMENDMENT TO DESIGN - TEST DATA

Now I have started to program my game, I've started by simplifying the navigation for my game this for the mean time has resulted in me changing for the controls for things such as the menus. I have therefore created a test data table for this.

| Test Data     | Type    |
|---------------|---------|
| 'Left Arrow'  | Invalid |
| 'Right Arrow' | Invalid |
| 'Spacebar'    | Invalid |
| 'Enter'       | Invalid |
| 'P'           | Invalid |
| 'W'           | Invalid |
| 'A'           | Invalid |
| 'S'           | Invalid |
| 'D'           | Invalid |
| 'L'           | Valid   |
| 'V'           | Valid   |
| 'C'           | Valid   |
| 'E'           | Valid   |
| 'I'           | Valid   |
| 'Esc'         | Valid   |

As you can see these bottom controls weren't in my original test data plan, these will be valid controls for the meantime whilst I am developing my game, however I will change to arrows and mouse clicks as I further develop my game. Also note the other controls will result in invalid functions.

### 17/01/2017 - AMENDMENT TO DESIGN - KEY VARIABLES

Now that I have actually started to program my game I have found it easier to address and understand the key variables I will be using in relation to my game.

**'Field loadingscreen:Image':** This is responsible for creating a variable for which the loading screen will be called, this is declared as type image because the loading screen will be an image.

**'Global Game:Game\_app':** This is the main game variable, it is responsible for storing the game whilst it is being run, this is a requirement for every game in Monkey - X.

#### GAMESTATES

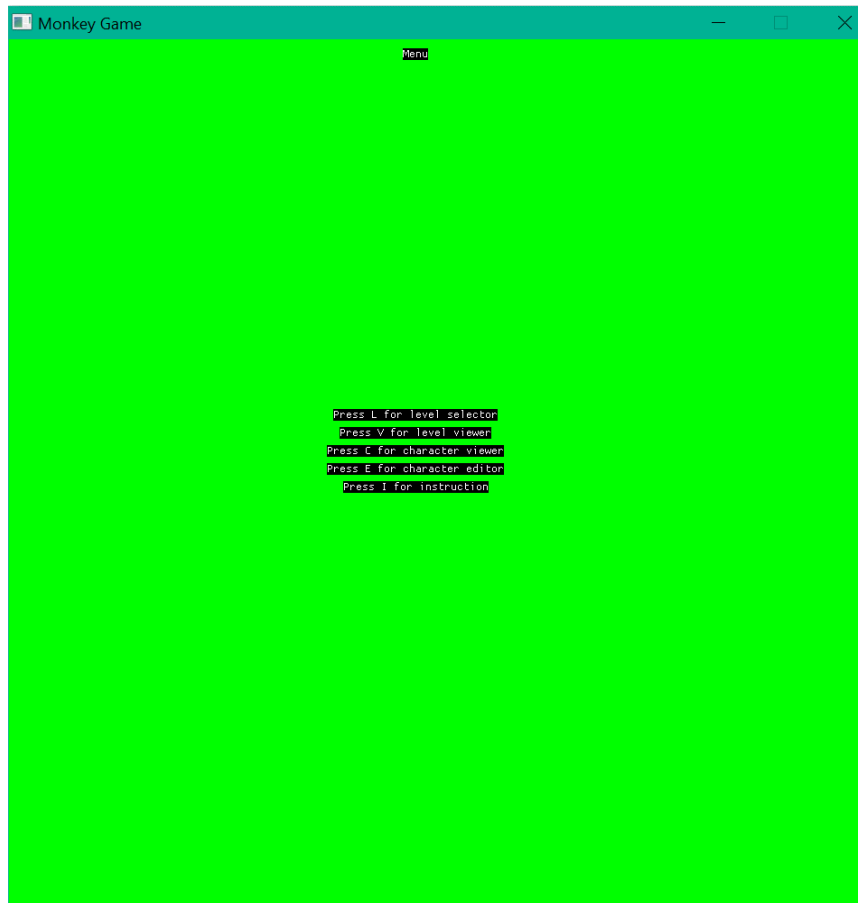
Game states are an important part of my game, they are basically the different screens that will appear and what will make up my game. Game states enable to program the specific functions and actions that will make up my game. So far after developing my game for a few weeks I have around 7 game states all requiring different types and levels of functionality.

**Loading screen:** Loading screen has minimal functionality and is responsible for displaying the loading screen image that will show when the game is first run. This has one control in the way of space which changes the game state from loading screen to menu.

Candidate Name: [REDACTED]

Candidate Number: [REDACTED]

Menu: So far with my development there is minimal navigation and graphics for my menus, the menu has a list of controls for which will change the game state depending on what is pressed.



The next screens all have very similar functionalities, and as of so far they don't have anything on except the simple monkey x font displaying the title. On any of these pages clicking escape will result in returning to the main menu.

## 18/01/2017 LEVEL CLASS CONTINUATION

After looking at the maze tutorial game I had been studying and after understanding it I decided that it would be the most convenient way of having tiles in my system so have copied parts from the code because I'm not aware of how to program the tile map features myself.

In our load method we need three variables, map, map\_data and data\_item. Local map is declared as a FileStream, this is our way of using our data file responsible for drawing the map. Local map\_data is what's actually in the data file so the digits that decided what is to be drawn. Finally, data\_item is one row of the tiles.

```
Method load()
 Local map:FileStream
 Local map_data:String
 Local data_item:String[]
```

We use the map variable to open the data file and data\_file is assigned as the contents.

```
map = FileStream.Open("monkey://data/level1map.txt", "r")
map_data = map.ReadString()
map.Close
```

The next section of code is responsible for separating the tiles as well as other things. 'data\_item = map\_data.Split("~n")' this splits the rows at the carriage return which basically means it resets the position to the beginning of the line again. The next are two loops; a nested loop is used to iterate through all the tiles basically a loop, loops inside another loop. As they are stored as strings in a text file they are converted to integers.

```
data_item = map_data.Split("~n")
For Local y:Int = 0 To 14
 For Local x:Int = 0 To 49
 tiles[x][y] = Int(data_item[y][x..x+1])
 Next
Next
End
```

## 22/01/2017 LEVEL 1 CONTINUATION

Finally, we can start programming our last method which is responsible for actually seeing what the number is in the text file and then drawing the corresponding tile to the screen. This method contains one local variable called tile which is a string. The method uses a nested loop to go through the array so it can draw them to the screen, it must know what tile to draw and also where to draw the tile. In my level 1 class there are 7 different tiles that make up my map. The numerous if statements say which tile needs to be drawn, then using the tile set variable that tile is drawn with size 32x32.

```
Method draw()
 Local tile:String
 For Local y:Int = 0 To 14
 For Local x:Int = 0 To 49
 tile = tiles[x][y]
 If tile = BrownTile Then DrawImage tileset, x*32, y*32, BrownTile
 If tile = SlopeDown Then DrawImage tileset, x*32, y*32, SlopeDown
 If tile = SlopeUp Then DrawImage tileset, x*32, y*32, SlopeUp
 If tile = BPlatform Then DrawImage tileset, x*32, y*32, BPlatform
 If tile = Sky Then DrawImage tileset, x*32, y*32, Sky
 If tile = FPlatform Then DrawImage tileset, x*32, y*32, FPlatform
 If tile = Lava Then DrawImage tileset, x*32, y*32, Lava
 Next
 Next
End
```

We use constants to declare what tile corresponds to what tile, for example the above code 'BrownTile' is given the value of 0; this means that when a zero appears in my text file for my level it will draw the brown tile at that position.

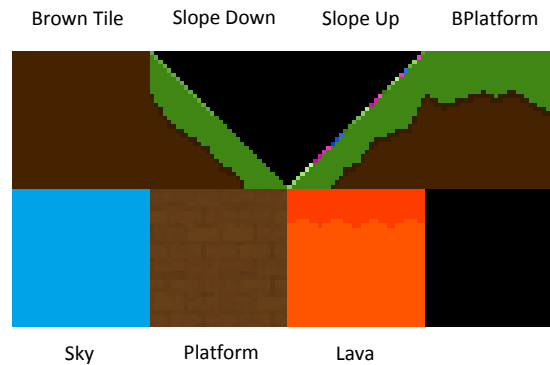
```
Class Level
 Const BrownTile = 0
 Const SlopeDown = 1
 Const SlopeUp = 2
 Const BPlatform = 3
 Const Sky = 4
 Const FPlatform = 5
 Const Lava = 6
```



Candidate Name: [REDACTED]

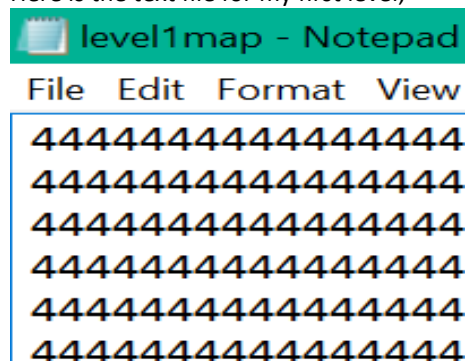
Candidate Number: [REDACTED]

As you can see from my tileset I have called the constants by their description which makes It easier to understand. I can use these tiles to create a map as shown below.



I have chosen to use a resolution of 1600x480 therefore meaning my map will be 50 tiles wide by 15. However, at the moment this all I have chosen to program however as I continue to develop my game I am hoping to include a far more sophisticated map system. This will involve the camera moving with the character, this means not all the map will be drawn to the screen at once, but instead only a bit will be viewable and as the character moves to the right more of the map will display. This not only allows me to have a smaller window but is also representative of the real New Super Mario Bros which my game is based upon. However, this feature is complicated and hard to implement which is why I am choosing to program it when I have finished the general basis for my game.

Here is the text file for my first level;



This code results in these tiles being drawn to the screen.



## 26/01/2017 INTERVIEW WITH ROBBIE

---

I have decided to hold an interview with Robbie in order to get his views on my current situation with the game, at the moment my menus are really simplistic and I only have 1 level but I'm hoping to get his opinions on certain features I could implement.

*Dan*

Robbie

### Using the level program...

*Looking at the level program so far are you happy with tiles and graphics?*

I like the similarities it shares with NSMB and like the tiles that are used.

*Are there any concerns you have?*

I'm not convinced that the current level would be hard enough and feel the best way of the levels working is by having a camera that moves with the character, this way the maps can be much bigger and there would be an element of surprise which the game is currently lacking.

### Using the main program...

*In terms of menu layout do you like how it currently is?*

Yes, the menus work the way I would expect.

*How would you like to be able to switch between the pages and use the menus; would you prefer to use the mouse and click the adequate choice or would you prefer to use the arrows keys and enter to press the selection?*

I would be happy with either, at the moment it is far too simple.

*What sort of graphics would you like to be included in the menus?*

I want colours and designs that run with the look of the game, I like the loading screen so a similar theme would be great.

*Is there anything that you don't like or concerns you at the moment?*

I would have preferred the level to be smaller and have the camera move along the level.

## 27/01/2017 IMPLEMENTING LEVEL CODE INTO MAIN GAME

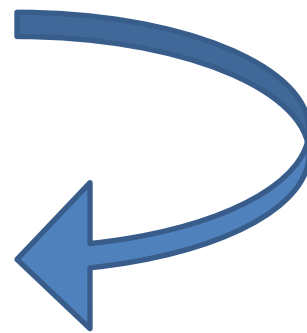
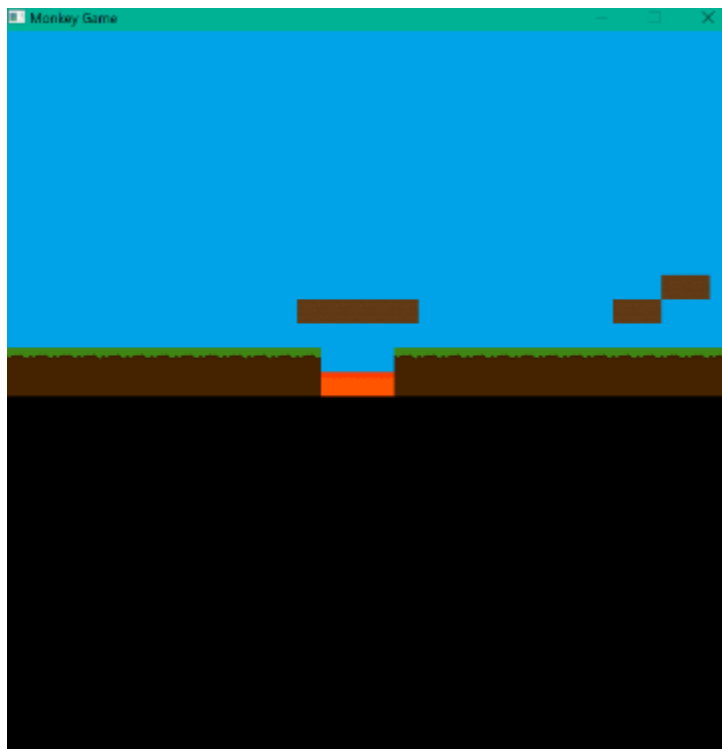
My aim is to add my fully working level class into my game so that it is accessible within the main game, the only thing that I'm aware will be a problem is the resolution as my actual game and my level program have different resolutions.

Here is how my game currently runs:



As you can see the loading screen is currently the wrong size, I have not got round to changing the image size so that it will run as the correct size on the game, I have not done this because I will change the loading screen near the end of development so it wouldn't be worth it. I need to change it to 960x960.





As stated earlier, the level class which I have now implemented into my game is the wrong size. I will not be changing this because I am hoping to have a specialized camera system which moves along the map.

## 28/01/2017 PLAYER CLASS

---

To start with I'm going to keep my player class very simplistic, I will not have movement but will start by just getting it to draw to the screen.

I have started by creating a class, called player, because it is a class I thought about what characteristics the player might have such as the speed, the position the image.

```
Class player
 Field knight:Image
 Field x:Int
 Field y:Int
 Field speed:Int
```

At the moment my class has 4 variables or 4 attributes, these are the image which for the time being is a knight however this is just to test the player class. The next is x which is an integer and is the x position of the player, y is also an integer and is the player's y position. Finally speed is how fast the player moves at, however this will not be off use until we implement the movement of the character.

For our player class we only require one method which is the method that actually draws the character to the screen as shown below.

```
Method draw()
 DrawImage knight, x, y
End
```

It has one line of code which simply draws the image 'knight' at the positions currently declared as x and y which I can set in on the update method of my own game.

However, before the player will be drawn to the screen I need to add code to my main game class such as the code to create a new instance of the player class and actually declaring the image that will be drawn. I firstly needed two variables at the beginning of the main class.

```
Field knight_sprite:Image
Field player1:player
```

The first declares that the png with the actual sprite image is an image, it is called knight\_sprite as that is what the image is. The other variable 'player1:player' declares the player1 as an instance of the class player which I have coded.

```
knight_sprite = LoadImage("testsprite.png")
player1 = New player
player1.knight = knight_sprite.GrabImage(0, 0, 32, 32)
```

This sets the previously declared variable 'knight\_sprite' as the image which is in the data file along with my game shown below, it is a knight that is 32x32.



'player1 = New player' creates a new instance of the player class and calls it player 1, this is so I can create more characters later on in the development of my game. The final line of code gets the actual image from the picture, and declares it as the knight image of player1. This means when the draw function is used this image is drawn to the screen in the position that is declared as x and y coordinates.

The player is drawn to the screen as you can see at the position 0, 0 as declared further up in the player class, it currently has no movement features or anything so that is next to be added.



## 01/02/2017 CHANGING MY PLAYER CLASS

I came to a realisation that I wasn't happy with my player class, my knight was too small and also wasn't very relevant to my game. Because the character was so small removing the background on photoshop didn't work I decided to use a different sprite and have decided to use a 45x45 sprite as well because it looks better on the map.



This was the sprite that I decided to use, this has also had the background removed which means the character will not have a black background around it on the game. Adding this sprite was easy to do because I already had all the code so changing it for the new character was simple.

```
Field p1_img:Image
Field player1:player

Method OnCreate()
 SetUpdateRate 60
 loadingscreen = LoadImage ("loadingscreen.png")
 level1 = New Level
 level1.load()

 p1_img = LoadImage("sprite.png")
 player1 = New player
 player1.p1Sprite = p1_img.GrabImage(0, 0, 45, 45)

End
```

As you can see in the main game class, I have decided to change the name of the sprites image variable and feel it is now more aptly named. In the oncreate the 'sprite.png' image is declared as the p1\_img, p1Sprite is the actual individual image of the sprite. No longer is the sprite 32x32 but I have changed it 45x45 so the sprite is better and looks better in game.

```

Class player
 Field p1Sprite:Image
 Field px:Int
 Field py:Int
 Field speed:Int

 Method draw()
 DrawImage p1Sprite, px, py
 End
End

```

Similarly, I have changed the player class, the variables have changed except for speed. The image variable has changed name as shown further up in the program. My player x and y position is now more appropriately name px and py for when I add more players later in the development of my game.

Setting the position by setting the px and py in the onupdate method off my main game class, I have set it 10 and 374 using trial and error for where it appears in regards to the platforms of my game.

```

Method OnUpdate()
 Select GameState
 Case "LOADING_SCREEN"
 If KeyHit (KEY_SPACE) Then GameState = "MENU"

 Case "MENU"
 If KeyDown (KEY_L) Then GameState = "LEVEL_SELECTOR"
 If KeyDown (KEY_V) Then GameState = "LEVEL_VIEWER"
 If KeyDown (KEY_C) Then GameState = "CHARACTER_VIEWER"
 If KeyDown (KEY_E) Then GameState = "CHARACTER_EDITOR"
 If KeyDown (KEY_I) Then GameState = "INSTRUCTIONS"

 Case "LEVEL_SELECTOR"
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"
 If KeyHit (KEY_1) Then GameState = "LEVEL_1"

 Case "LEVEL_VIEWER"
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"

 Case "CHARACTER_VIEWER"
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"

 Case "CHARACTER_EDITOR"
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"

 Case "INSTRUCTIONS"
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"

 Case "LEVEL_1"
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"
 player1.px = 10
 player1.py = 374
 End
End

```



My new sprite now draws to the screen, and sits on top of the platform, there is no black square around the sprite and the tiles are shown so it looks like the character is actually part of the screen. My next step is to get the movement of the player moving which will mean adding gravity and controls for moving the character.

## 05/02/2017 ADDING MOVEMENT TO PLAYER CLASS

I have started this by changing the variable in my player class called speed, I have changed its name to 'pspeed' to make it easier to understand for later development.

The easiest way to add player movement is to make it so when the arrow key is pressed the speed becomes whatever the speed is set to so that the player moves.

```
If KeyDown (KEY_RIGHT) Then
 player1.px = player1.px + 2
End
```

I have added this if statement which is fairly simple, it checks if the right arrow is pressed down and if it is then it adds two to the player x position. However, this method hasn't worked and I have encountered a problem, pressing the right arrow does move the player two to the right but then the player just returns to the original position straight away.

However, I have realised my error, my player x and y position was set in the level 1 gamestate of my onupdate method and they were set to certain values. Despite when the right arrow being pressed the player x position changes to the player x position + 2 the onupdate method is constantly being run so it just resets to the original position again and again. I have solved this and solved possible further problems by creating a new gamestate called initialise which will contain all of my variable setting such as x and y position and speed as shown below.

```
Case "INITIALISE"
 player1.px = 10
 player1.py = 374
 player1.pspeed = 2.0
 GameState = "LOADING_SCREEN"
```

This initialise gamestate will now be used to set all of the game values before the game is run, you cannot see this gamestate actually doing anything because it does it instantly then changes to the loading screen.

```
Case "LEVEL_1"
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"

 If KeyDown (KEY_RIGHT) Then
 player1.px += (player1.pspeed)
 End

 If KeyDown (KEY_LEFT) Then
 player1.px -= (player1.pspeed)
 End
```

In the level1 gamestate of the onupdate method I have improved upon the if statement and have also added a new one. It basically sees if the right or left arrow is pressed and then adds or minuses the value of player speed which is currently set to 2. If the right arrow is pressed the x position value needs to increase so it adds to the value, whereas if the left arrow is pressed the x position value needs to decrease so it minuses from the value. This results in the player moving to left or right depending on the key pressed.





Candidate Name: [REDACTED]

Candidate Number: [REDACTED]

**Testing the player movement**

| Test Data     | Type    |
|---------------|---------|
| 'Left Arrow'  | Valid   |
| 'Right Arrow' | Valid   |
| 'Spacebar'    | Valid   |
| 'Enter'       | Valid   |
| 'P'           | Valid   |
| 'W'           | Invalid |
| 'A'           | Invalid |
| 'S'           | Invalid |
| 'D'           | Invalid |

Here was my original test table, I am going to try testing some of these game on my game.

| Input         | Output                    |
|---------------|---------------------------|
| 'Left Arrow'  | Player moves to the left  |
| 'Right Arrow' | Player moves to the right |
| 'Spacebar'    | Null                      |
| 'Enter'       | Null                      |
| 'P'           | Null                      |
| 'W'           | Null                      |
| 'A'           | Null                      |
| 'S'           | Null                      |
| 'D'           | Null                      |

After testing some of the controls outlined in my test data, the 3 buttons spacebar, enter and p don't do anything as of yet but this is expected because I haven't started programming their functions yet. As expected the left and right arrow do the appropriate thing.

## 06/02/2017 ADDING PLAYER JUMPING

I have decided to go about implementing this in a simple manner and then building upon it when I improve the player movement system. I will code the player jumping in a very similar fashion to the player movement left and right. With Monkey X there isn't a fixed jumping procedure or button, instead I should create an array that changes the players position increases gradually and then decreases it again so that the player falls to where they were.

I have started by thinking about what code needs to go in my player class that I have created so far, after conducting some research on player jumping I came across some code which makes a cube jump into the air and then fall. I considered this to be the same as what I was aiming to do on my game, instead my sprite would jump into the air then fall to where they were.

### Monkey - 2D Player jumping code example

Press space bar to jump the rectangle player



The code below is an example where a rectangle can jump with gravity. The space

code below :

Tap to code

```
Global posYfloat = 320
Global posYfloat = 240
Global playerJumpBool = False
Global pincyFloat = 0

Class MyGame Extends App
 Method Initialize()
 SetUpdateRate(60)
 End
 Method OnUpdate()
 ' If the player is on the ground and the space bar is pressed
 If playerJump = False And KeyDown(KEY_SPACE) = True
 pincy = 0
 playerJump = True
 End
 ' If the player is in the jump
 If playerJump = True
 pincy += 0.1
 ' If the player is going up
 If pincy <= 0
 For Local i:Int = 0 Until Abs(pincy)
 zy += 1
 End
 End
 ' If the player is going down
 If pincy > 0
 For Local i:Int = 0 Until pincy
 zy += 1
 ' If the player touches the ground
 If zy > 240 Then
 playerJump = False
 pz = 240
 Exit
 End
 End
 End
 End
 End
 Method OnKeyDown()
 If (KEY_SPACE)
 SetColor(255,255,255)
 DrawRect pz,py,32,32
 DrawText "Press space bar to jump the rectangle player",18,28
 End
 End
End

Function Main()
 New MyGame()
End
```

Created by Rudy van Etten at 3:01 PM

- [Open Example \(like page\)](#)
- [View Tutorial with all parts](#)

#### Links

- [Click page with the examples](#)
- [My Youtube](#)
- [My Twitter](#)
- [Java game programming step by step blog](#)
- [HTML5 game programming step by step blog](#)

#### About me



**Rudy van Etten**

[Follow](#) 73

I am Rudy van Etten, I am 38 years old and live in the Netherlands. When I was young I started programming on the Commodore 64 and I never stopped programming. Today I program in Java, JavaScript and PHP. I also make videos on my Youtube channel and I also make HTML5 games. I have also done some programming since I was young.

[View my complete profile](#)

#### Blog articles

- in 2017 (39)
- in 2016 (52)
- in 2015 (140)
- in 2014 (24)
- in December (26)
- Monkey 3 - CreateImage, AddHandle, DrawImage scale...
- Monkey 3 - Creating and drawing an image from area...
- Monkey 3 - Whiffles! and function and LocalVar...
- Monkey 3 - DrawRect and DrawRect using array code...
- Monkey 3 - Draw Rectangles Code Example
- Monkey 3 - AddRect, Width, Height and Data code...
- Monkey 3 - Start, Pop, and Push and Length and Index...
- Monkey 3 - 2D Scrolling map code Example
- Monkey 3 - String Slicing (E..F) Java code Example...
- Monkey 3 - 2D Physics using Java code example
- Monkey 3 - Space Invaders using Java code example...
- Monkey 3 - DrawRect using st.A, st.B, st.C function...
- Monkey 3 - 2D Player jumping code example
- Monkey 3 - Asteroid Field with Class and 100 code...
- Monkey 3 - Player class and creating player class...
- Monkey 3 - Drawing a rectangle below the main code...
- Monkey 3 - 2D topdown moving the player with the code...
- Monkey getting started - Setting up and creating a...
- Monkey getting started - Drawing and updating a 2D...
- Monkey getting started - Drawing rectangles using f...
- Monkey getting started - Drawing a red rectangle...

This was the program from Rudy Van Etten, it makes a square jump in the air when spacebar is pressed and then falls which gives the impression of jumping. I have decided that this would be the perfect code for making my player jump and have therefore implemented it into my game.

The section of code can go into the player class; there are 2 global variables – playerjump which is a Boolean expression saying whether the player is currently jumping or not, the other is gravity which is basically the player's y position. The rest of the code is going inside my player initialise method where the player drawing code is as well.

```
Method initialise()
 If playerjump = False And KeyDown(KEY_SPACE) = True
 gravity = -4
 playerjump = True
 End

 If playerjump = True
 gravity += 0.1

 If gravity <= 0
 For Local i:Int = 0 Until Abs(gravity)
 py -= 1
 End
 End

 If gravity > 0
 For Local i:Int = 0 Until gravity
 py += 1
 End

 If py > 374 Then
 playerjump = False
 py = 374
 Exit
 End
 End
 End

 End
 DrawImage p1Sprite, px, py
End
```

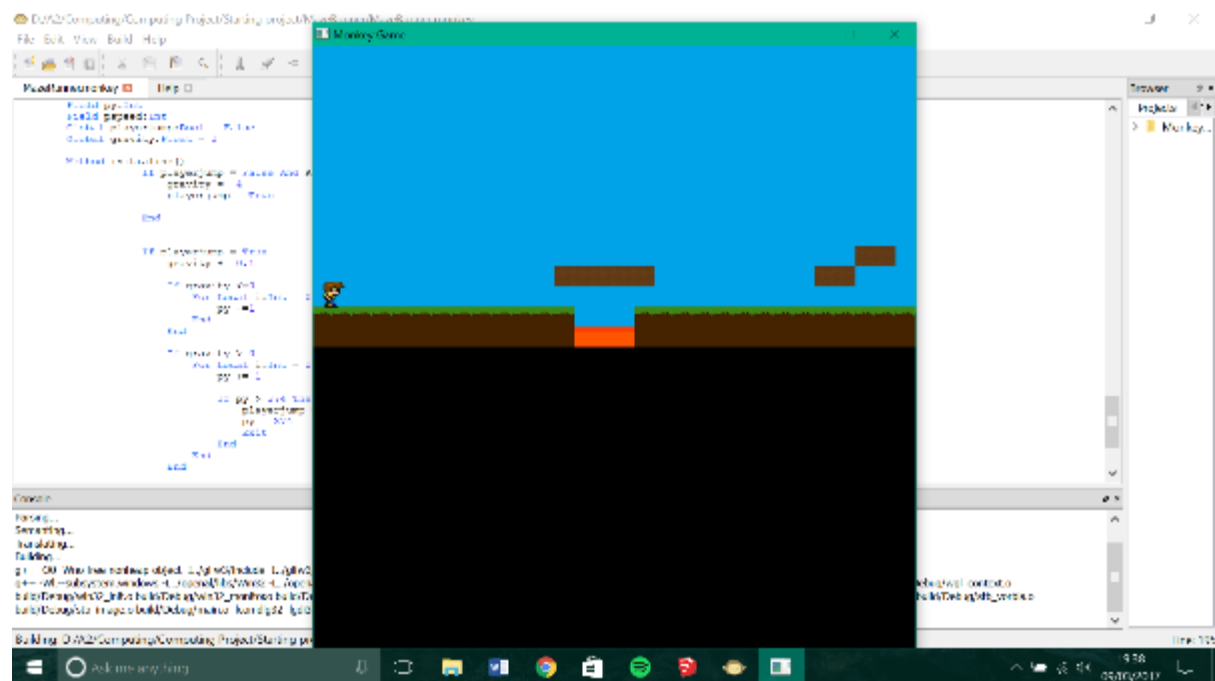
The main body of code consists of a number of IF statements, the main one at the beginning is responsible for actually triggering the player jumping. 'If playerjump = False And KeyDown(KEY\_SPACE) = True' this basically checks that if the player is not currently jumping and player has pressed the jump button (spacebar) then set gravity to -4 and set that the player is jumping.

There is then the main if statement for when the 'playerjump' is set to true, gravity is also set to 0.1 + the current gravity level, this is basically reducing the character's y position so that the player falls. If the gravity is less than or equal to 0; it loops from 0 until the absolute value of gravity and minuses 1 from the players y position. If the gravity is more than 0 it loops from 0 until the value of gravity and adds 1 to the players y-position. The final if statement stops the player from being able to jump mid-air, I will have to alter this when my players y-position changes.

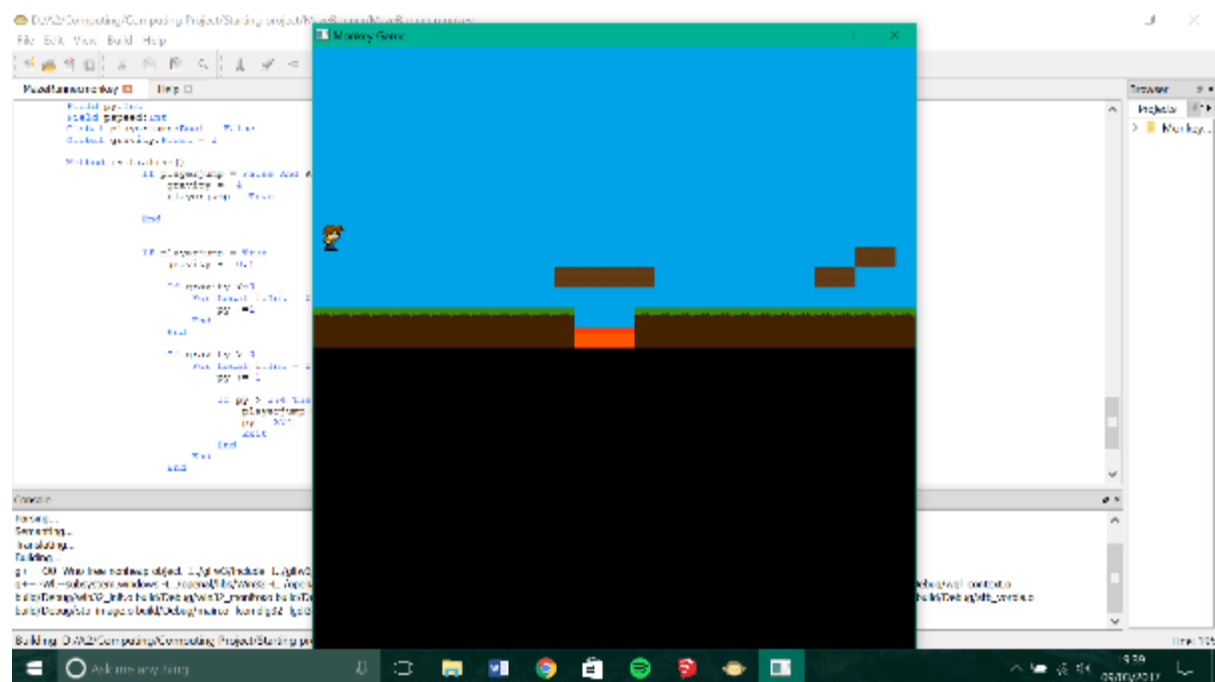
Candidate Name: [REDACTED]

Candidate Number: [REDACTED]

## 09/02/2017 TESTING FOR PLAYER MOVEMENT



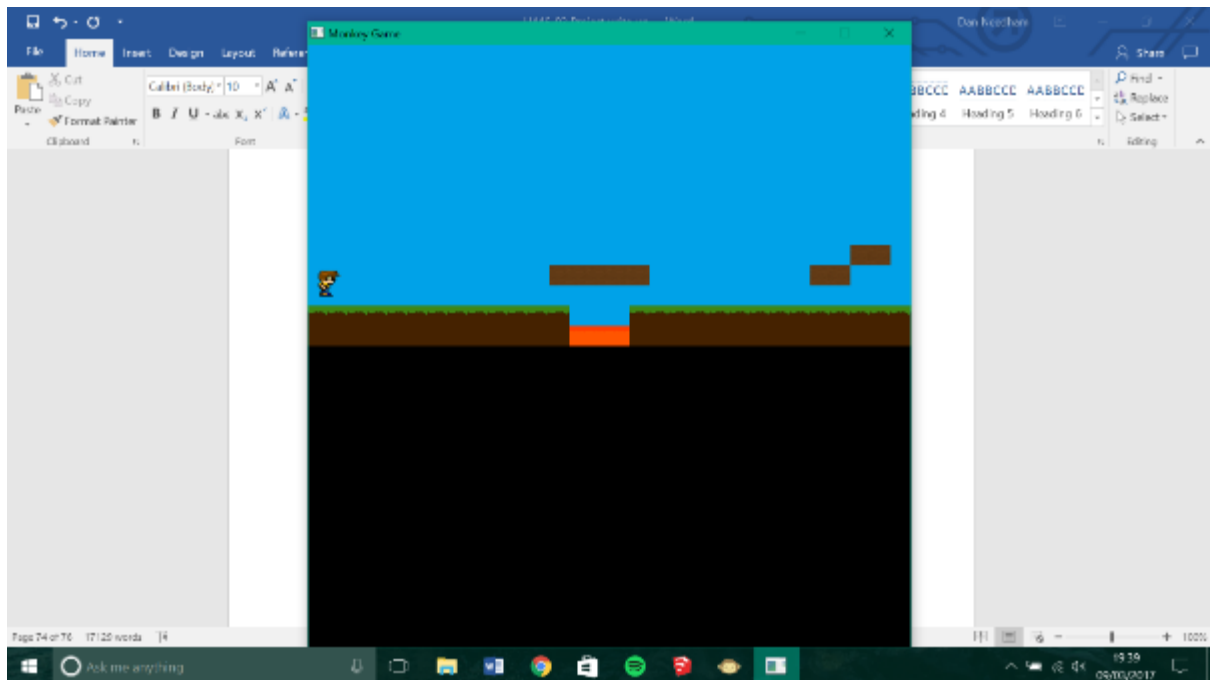
### Pressing Spacebar



Candidate Name: [REDACTED]

Candidate Number: [REDACTED]

Player falls



As you can see from the test table for movement, the spacebar button should be a valid input because this is the button for jumping, after testing this button works and jumping works.

| Test Data     | Type    |
|---------------|---------|
| 'Left Arrow'  | Valid   |
| 'Right Arrow' | Valid   |
| 'Spacebar'    | Valid   |
| 'Enter'       | Valid   |
| 'P'           | Valid   |
| 'W'           | Invalid |
| 'A'           | Invalid |
| 'S'           | Invalid |
| 'D'           | Invalid |

## 12/02/2017 INTERVIEW WITH ROBBIE

Dan

Robbie

*After playing with the character and using the jumping function, are you happy with it currently?*

Yes, in terms of how the player reacts and moves it is perfect and being able to move whilst jumping is something like in New Super Mario Bros.

*Do you think the movement replicates that off New Super Mario Bros?*

I am very happy with the feel and how similar it is to the real game.

*Is there anything you would like to be added or improved with the movement?*

I would like the players image to change, for exam flip and face the direction that the player is currently moving.

*What would you like to see done by the next time you view the game?*

I would like collisions with tiles to be done so that the player can stand on top of the tile and not just jump through it.

**What can I conclude from this interview?**

The next thing in my game that my client wants me to add is collisions with platforms, at the moment when the player moves they just go through the platform because at the moments the platforms are basically invisible

## 15/02/2017 COLLISIONS

---

The primary aim for collisions is that it compares the position of the player with the position of the tiles and sees if they are the same in terms of coordinates in which case the tile and player would be colliding. This means that I will need to find out the coordinates of all the edges of the tiles and the outline of the character's coordinates and then compare them to see if they are colliding.

The code for collisions will go into the level class in a method called tile\_collision, after looking at the maze runner game by 'Craig n Dave' and seeing how collisions work and after struggling with collisions and how to implement them I decided upon building upon the tile collision method.

```
Method tile_collision:Int (x:Int,y:Int)
 Local left_tile:Int = x / 32
 Local right_tile:Int = (x+31) / 32
 Local top_tile:Int = y / 32
 Local bottom_tile:Int = (y+31) / 32

 If left_tile < 0 Then left_tile = 0
 If right_tile > 49 Then right_tile = 49
 If top_tile < 0 Then top_tile = 0
 If bottom_tile > 14 Then bottom_tile = 14

 Local collision_result:Int = 0
 For Local i:Int = left_tile To right_tile
 For Local j:Int = top_tile To bottom_tile
 If tiles[i][j] = FPlatform Then collision_result = FPlatform
 Next
 Next
 Return collision_result
End
```

It creates 4 local variables which are basically find the edges of the tiles that could be colliding with, the next part ensures that the tiles are within the 50x15 array of my level. The next part is responsible for checking the four edges and sees which tile is colliding.

After sorting the first part of collisions out I decided that the best thing to do was to create a small text state which said whether the player is colliding with one of the tiles. I considered this to be a good test before working out the code for not allowing the player to move through the colliding tile.

```
If level1.tile_collision(player1.px, player1.py) = level1.FPlatform Then
 collision = True
Else collision = False
End
```

It checks if there is a collision between FPlatform which in my game is one of the floating platforms and the player1's coordinates and then sets collision to true. 'collision' is a Boolean expression I have used which switches between whether the player is colliding with the tiles or not.

```
If collision = True Then
 DrawText("Collision", 10, 10)
Else collision = False
 DrawText("No Collision", 10, 10)
End
```

If collision is true then it draws collisions but if there is no collision then it outputs no collision, we can look at this in action.

Candidate Name: [REDACTED]

Candidate Number: [REDACTED]



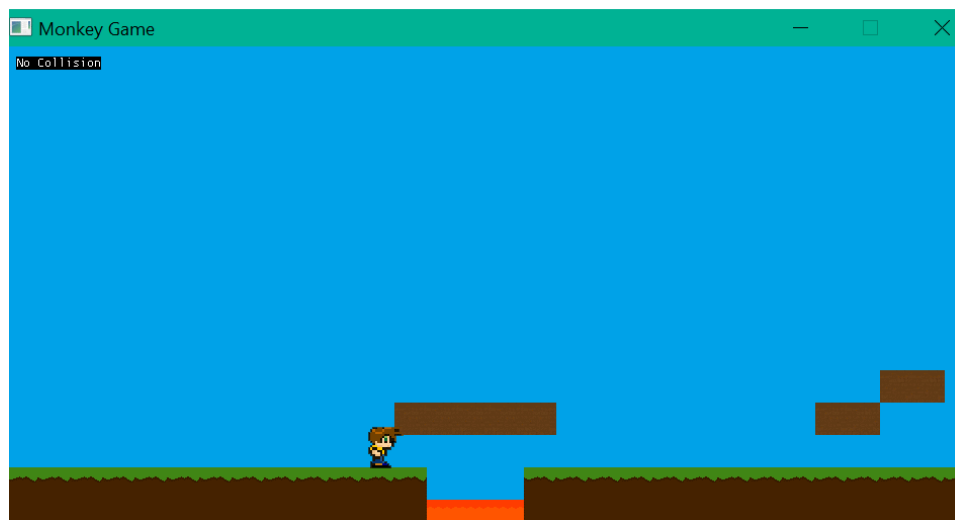
## 23/02/2017 MAKING COLLISION OBJECTS SOLID

---

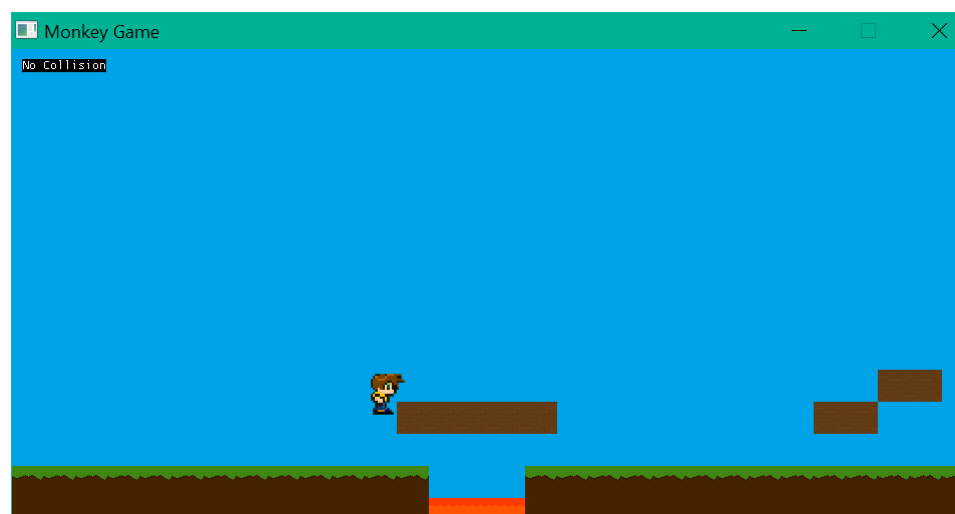
So far my program can compare the coordinates of the player and tiles and then determine whether they are colliding, for the moment my program just outputs this result but now I need to code it so that there is an actual response to them colliding such as stopping the player from moving.

```
If collision = True Then
 player1.px -= player1.pspeed
 player.playerjump = False And player1.py + 0
End
```

When collision is true, the players position is set to be the same as it minus the players speed, this stops the player from moving when it gets to the tile.



As you can see from the screen shot when the player gets to the platform the player stops, however I encountered problems when jumping onto the top of the platform where the player would slide to the left until the edge of the tile and then stay in midair as shown below.

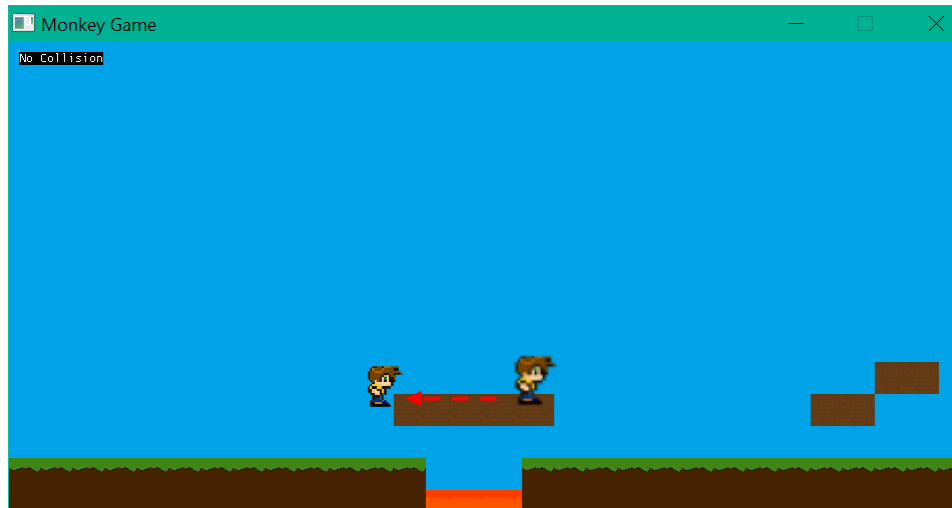




Candidate Name: [REDACTED]

Candidate Number: [REDACTED]

This is because as part of my collision code it disables player jumping so that the character doesn't fall when they collide with the top of the tile but when the player slide past the platform the player jumping method is still disabled so the player doesn't fall. If you press the jump button the player jumps and then falls to the base platform however. The player also shouldn't slide when they land on the top of the platform which as represented below is what happens.



This is down to the line of code – `'player1.px -= player1.pspeed'` , this is all well and good for the vertical collisions with tiles where this stops the players position when they collide with the tile and allows the player to move away from the tile. However, when the player lands on the top of the tile the players position moves to the left because of the players speed, when it moves to the left it is still colliding so it continues to move to the left. When it gets to the left of the tile it doesn't fall because player jumping method is disabled.

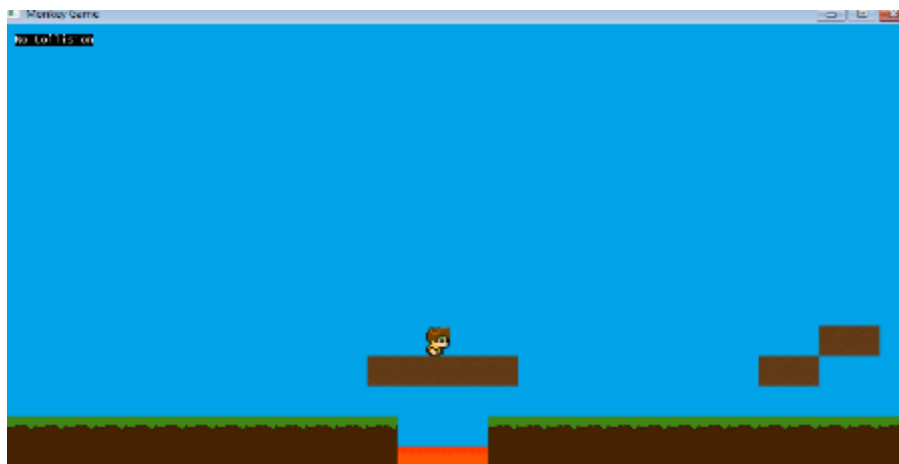
## 26/02/2017 IMPROVING COLLISIONS

---

The problem with the code I currently have is that the x and y collisions are dealt with in exactly the same way as each other, so when it lands on top of a tile it might think it is colliding with any of the edges of the tile. The code that I use to stop the player moving when they hit a tile side on does not work effectively for when the player collides from the top. I have therefore added another if statement when collision is true using the code 'player1.py -= player.gravity'. When a player falls it's because of the gravity value so subtracting this value from the player's y position means that the player stops falling.

```
If collision = True Then
 player1.px -= player1.pspeed
End

If collision = True Then
 player1.py -= player.gravity
 player.playerjump = False And player1.py + 0
End
```



The player is now 'held' by the platform and also it doesn't move from side to side. Note that I have changed the player's size to 32x32 instead of 45x45 temporarily as it's much easier with collision detection when both the tiles and characters are the same size.

## 29/02/2017 USER FEEDBACK – INTERVIEW WITH ROBBIE

---

The aim with the interview with my client is to see if they are happy with the additions to the game, especially whether they are happy with the collisions which make up a big part of the game.

*Dan*

Robbie

*Are you happy with how the collisions work?*

The collisions work okay but I am not completely convinced by them, collisions from the side work acceptably and top collisions as well however collisions from the bottom are buggy and don't work properly. As a whole, the collision system is not exactly smooth and collisions between the character and tiles is buggy.

*Is there anything you would like to be added to the game next?*

Ideally for the game to scroll or change to the appearance because at the moment the displayed level size is not acceptable and not much can go on as there are only a few platforms.

*If this is sorted would you like enemies and power-ups?*

Once the level is more expanded I think it would be appropriate to have enemies and powerups and some kind of progression such as a finishing area which ends the levels.

## 31/02/2017 TESTING COLLISIONS

---

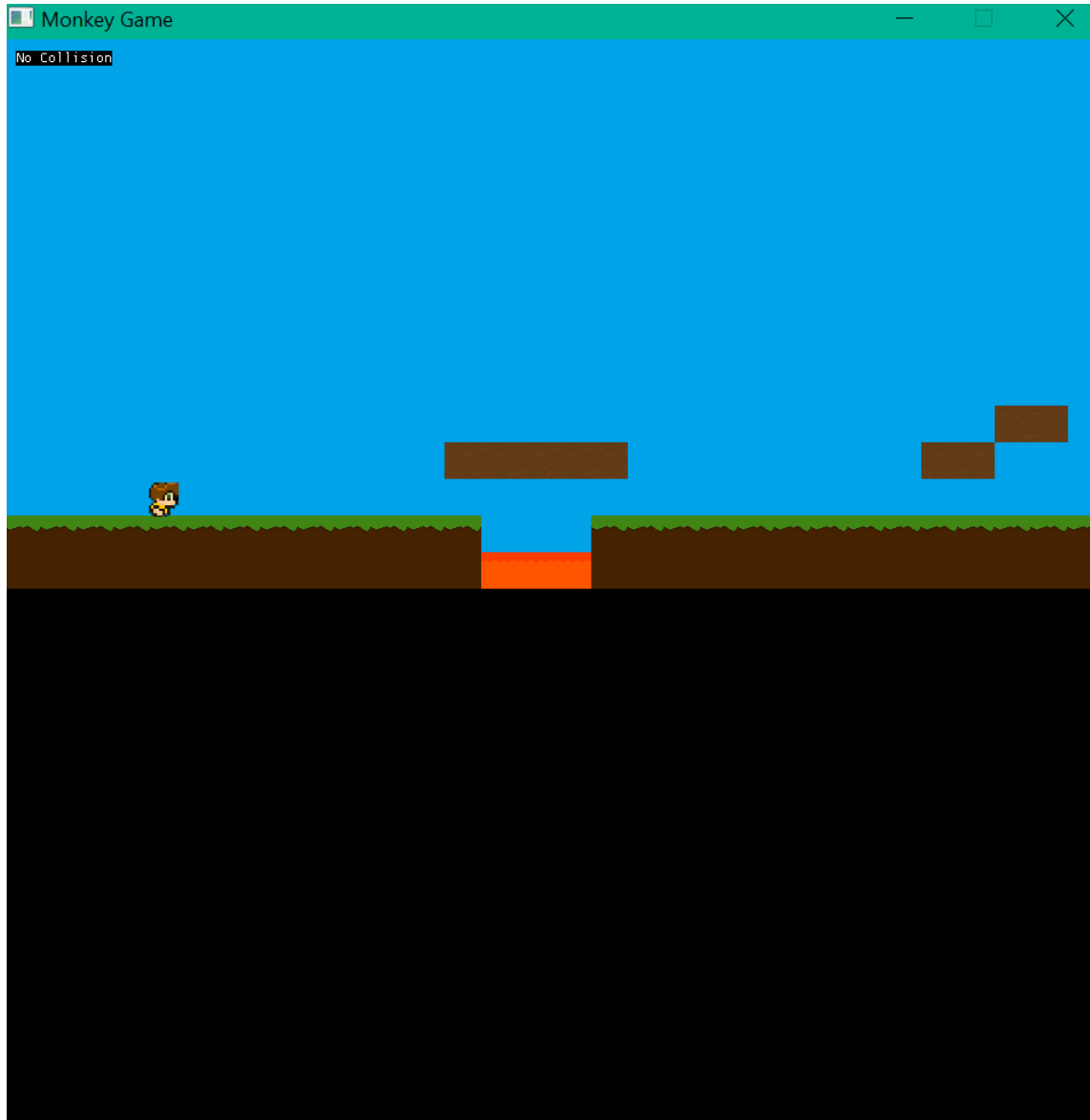
| Type of collision | Expected Outcome                                                                                                                                                                              | Actual Outcome                                                                                                                                                                                             |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Left Side Tile    | Player should stop moving to the right and appear to hit the tile holding it in position.                                                                                                     | Player stops moving but glitches slightly player can move left and the tile stops it moving to the right.                                                                                                  |
| Right Side Tile   | Player should stop moving to the left and appear to hit the tile holding it in position.                                                                                                      | Player can move through the tile if the left arrow key is held, this is also glitchy.                                                                                                                      |
| Bottom of Tile    | Player should hit the tile and not be able to go through it. It is likely the player should be jumping, therefore the player should collide with the tile stop but fall back to the platform. | Player is stopped by the tile moves to the left slightly. However, if you jump holding down space the player can move through the tile.                                                                    |
| Top of Tile       | Player should be held by the tile and not fall through.                                                                                                                                       | Player is held by the tile this is because player jumping is disabled which is a problem because when the player moves away from the tile player jumping is still disabled so player just stays in midair. |

After testing these collisions, I have decided for the moment to move onto other parts of my game. The collisions would work far better and be more efficient if the collision system could detect which side of the tile the player is colliding with, this way I could code different responses to these collisions which make them work better and be less glitchy.

## 02/03/2017 CHANGING THE LEVEL SIZE

---

At the moment because of the size of the menu's the level is a different size from the window. I have decided to have the player start from the bottom and then the aim is to get to the top of the level avoiding, currently the level screen looks like:



This is easy to do now, the majority of the tile system has been coded so this is just a case of changing the size of my array size so that it draws to the screen where it is currently black. At the moment the screen size is 960x960 which is the equivalent of 30 tiles by 30 tiles.

```

Method New()
 tileset = LoadImage ("level1tilemap.png",32,32,7)
 For Local i:Int = 0 To 50
 tiles[i] = New String[16]
 Next
End

Method load()
 Local map:FileStream
 Local map_data:String
 Local data_item:String[]

 map = FileStream.Open("monkey://data/level1map.txt","r")
 map_data = map.ReadString()
 map.Close

 data_item = map_data.Split("~n")
 For Local y:Int = 0 To 14
 For Local x:Int = 0 To 49
 tiles[x][y]=Int(data_item[y][x..x+1])
 Next
 Next
End

Method draw()
 Local tile:String
 For Local y:Int = 0 To 14
 For Local x:Int = 0 To 49

```

This is the code that I had at the moment which creates the array for the tiles, in order to change the size of the screen I need to change the size of the array.

```

Method New()
 tileset = LoadImage ("level1tilemap.png",32,32,7)
 For Local i:Int = 0 To 50
 tiles[i] = New String[31]
 Next
End

Method load()
 Local map:FileStream
 Local map_data:String
 Local data_item:String[]

 map = FileStream.Open("monkey://data/level1map.txt","r")
 map_data = map.ReadString()
 map.Close

 data_item = map_data.Split("~n")
 For Local y:Int = 0 To 29
 For Local x:Int = 0 To 49
 tiles[x][y]=Int(data_item[y][x..x+1])
 Next
 Next
End

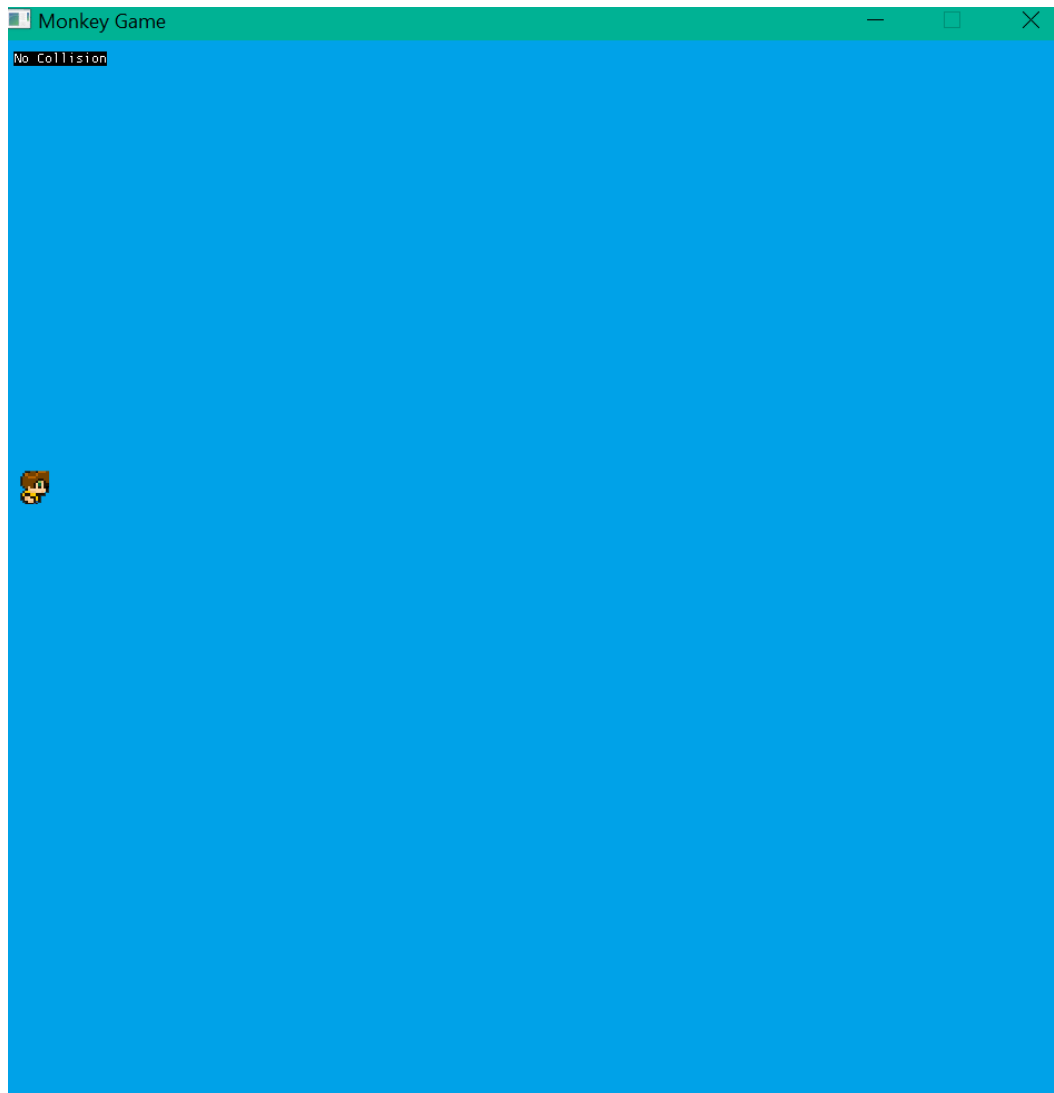
Method draw()
 Local tile:String
 For Local y:Int = 0 To 29
 For Local x:Int = 0 To 49

```

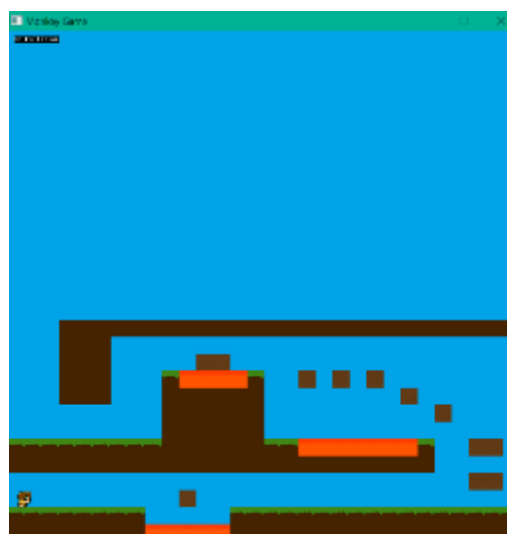
Before the Image screen was 15 tiles but I have now changed that to 30, this is so the game can be a bottom to the top game, I have also changed all the tiles to be a blue sky for the moment as you can see below. No I can start to sort out the appearance of the level introducing a finish line as well.

Candidate Name: [REDACTED]

Candidate Number: [REDACTED]



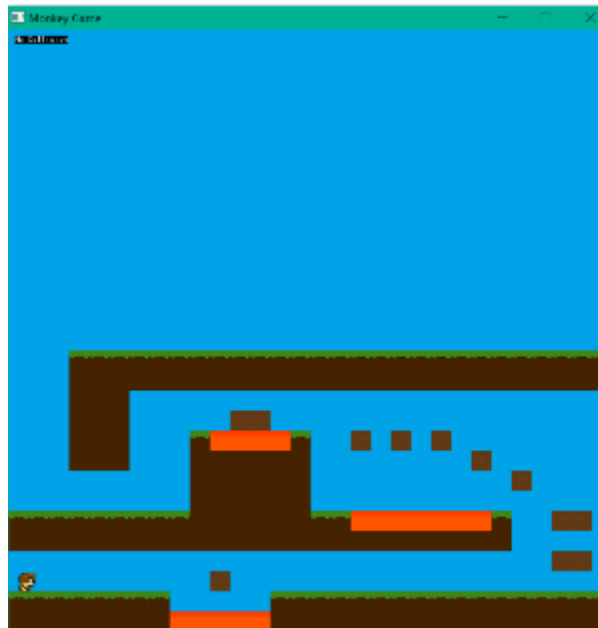
Now to finish this level I will just need to make changes to the level1 text file, drawing platforms this is relatively easy. This takes a lot of time changing about the text file and reviewing what the level then looks like.



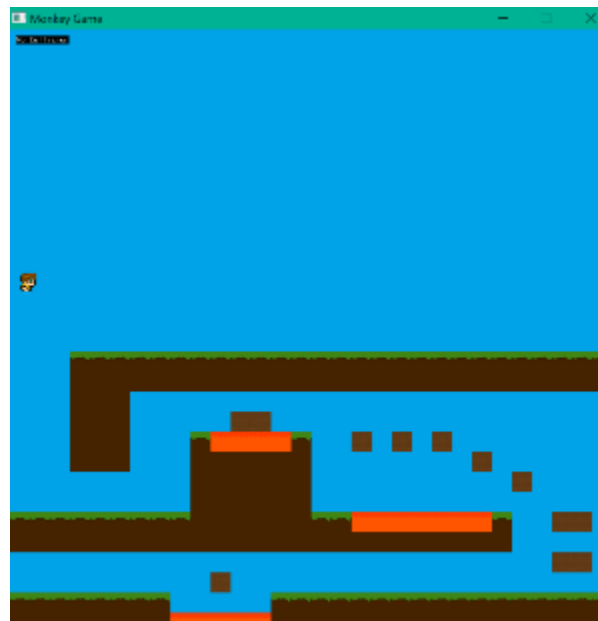
Candidate Name: [REDACTED]

Candidate Number: [REDACTED]

This involves lots of messing around with the txt file and adding numbers and deleting numbers to ensure that the tiles are in the right place. After testing around a bit with the character I encountered a problem where the player would teleport to a position in the middle of the level after jumping as shown below.

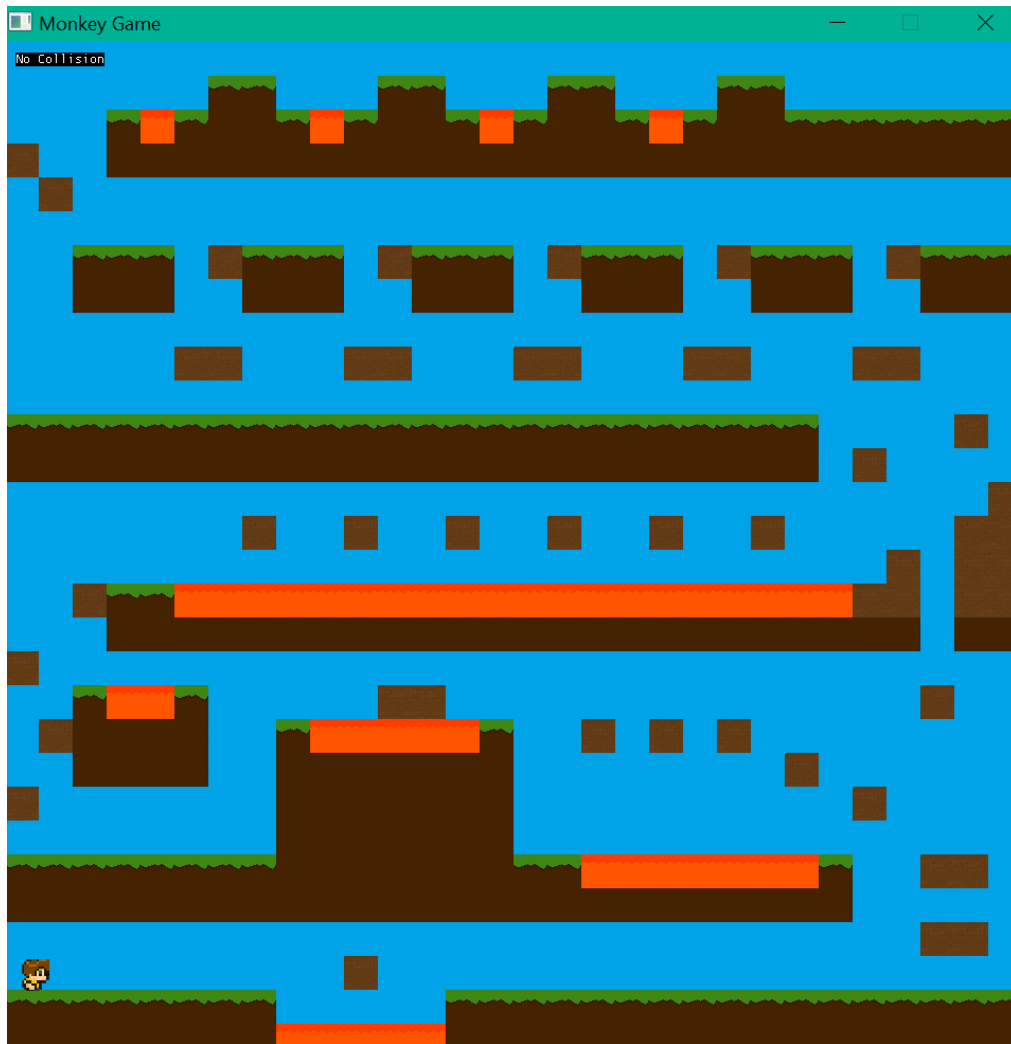


After pressing space bar (jump)



### 03/03/2017 FINISHING THE LEVEL OFF

After spending a while messing around with the appearance of the level I have settled on the first level but this is open to change after messing around with other things.



This results from the text file below.

```
level1
File Edit
4444444
4444444
444363
544000
4544444
4444444
4433334
4400004
4444444
4444445
4444444
3333333
0000000
4444444
4444444
4444444
4453336
4440000
5444444
4436663
4500000
4400000
5444444
4444444
```



## 06/03/2017 ADDING A FINISHING LINE

I am going to add a finish line at the top of my level so that the player can actually finish the level, when the player goes through the finish line it will say something along the lines of 'you have completed the level' and it will display all the score information once that has been added to the game.

Adding a finishing line is very easy as it is just like adding any other image.

```
Class Game_app Extends App
 Global GameState:String = "INITIALISE"
 Field level1:Level
 Field loadingscreen:Image
 Field p1_img:Image
 Field player1:player
 Field finishline:Image
 Global collision:Bool = False

 Method OnCreate()
 SetUpdateRate 60
 loadingscreen = LoadImage ("loadingscreen.png")
 level1 = New Level
 level1.load()

 p1_img = LoadImage("sprite.png")
 finishline = LoadImage("finishline.png")
 player1 = New player
 player1.p1Sprite = p1_img.GrabImage(0, 0, 32, 32)
 End
```

I first have started by declaring the image as a variable of the image type, the image is called finish line. In the main game class in the on create method is the code for loading the image from a file location. Because it knows to fetch everything from the data file with the game you just specify the name of the image which is 'finishline.png'.

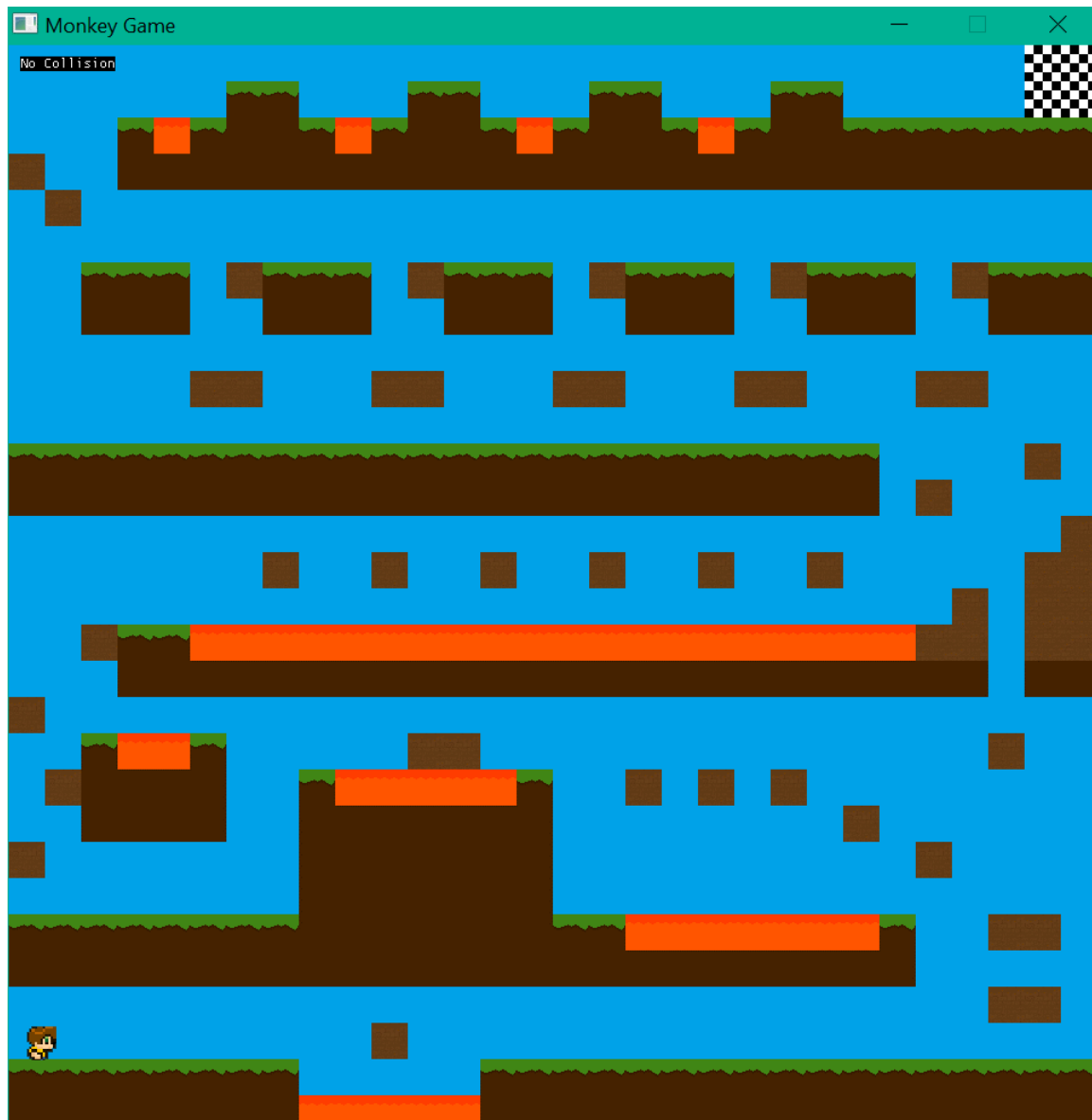
```
Case "LEVEL 1"
 Cls(0, 0, 0)
 DrawText("Level 1", 450, 10, 0.5)
 level1.draw
 player1.initialise
 DrawImage finishline, 896, 0

 If collision = True Then
 DrawText("Collision", 10, 10)
 Else collision = False
 DrawText("No Collision", 10, 10)
 End
```

All that is left is to actually draw the image in the on render method in the case level so basically on the level screen, it draws the finish line at the position, 896,0 which is the top right corner of the screen.

Candidate Name: [REDACTED]

Candidate Number: [REDACTED]



As you can see the finish line is drawn to the top right of the screen, this can now be used to deal with the end part of the level such as end of the timer when that is implemented and also calculating the players score.

## 08/03/2017 BUILDING UPON THE FINISH

---

I thought the easiest way of detecting when the player is at the finish is through comparisons between the image and the players position which is fairly simple. I have started by creating a global variable which is Boolean that basically states whether the level is finished or not, this will be used to draw things to the screen when the player is in the finishing zone.

```
Class player
 Field p1Sprite:Image
 Field px:Int
 Field py:Int
 Field pspeed:Int
 Field pxspeed:Int
 Field pyspeed:Int
 Global playerjump:Bool = False
 Global gravity:Float = 0
 Global levelfinish:Bool = False
```

The next bit of code checks the value of the players y position and x position and sees if the players position is within a certain area, in this case it's whether the player is in a 64 x 64 square at the top right off the screen or basically at the coordinates of where the finish line is.

```
If py < 64 And px > 864 Then
 levelfinish = True
Else levelfinish = False
End
```

The rest of the code is the response to what happens when the levelfinish = true, at the moment I have just created a text state which appears in the top corner saying Finish. If I test this and it works then I know that I can add to this such as displaying finish messages and scoring information when I get to that stage. This is easier its just a simple if statement which checks if 'levelfinish' is true then it draws the text.

```
Case "LEVEL_1"
 Cls(0, 0, 0)
 DrawText("Level 1", 450, 10, 0.5)
 levell.draw
 DrawImage finishline, 896, 0
 player1.initialise

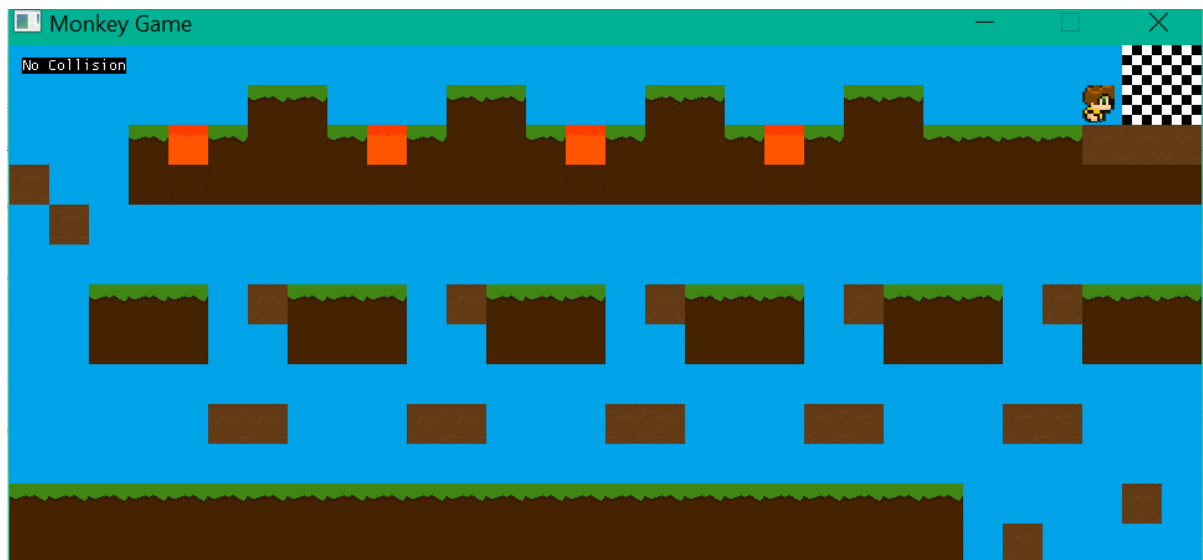
 If collision = True Then
 DrawText("Collision", 10, 10)

 Else collision = False
 DrawText("No Collision", 10, 10)
 End

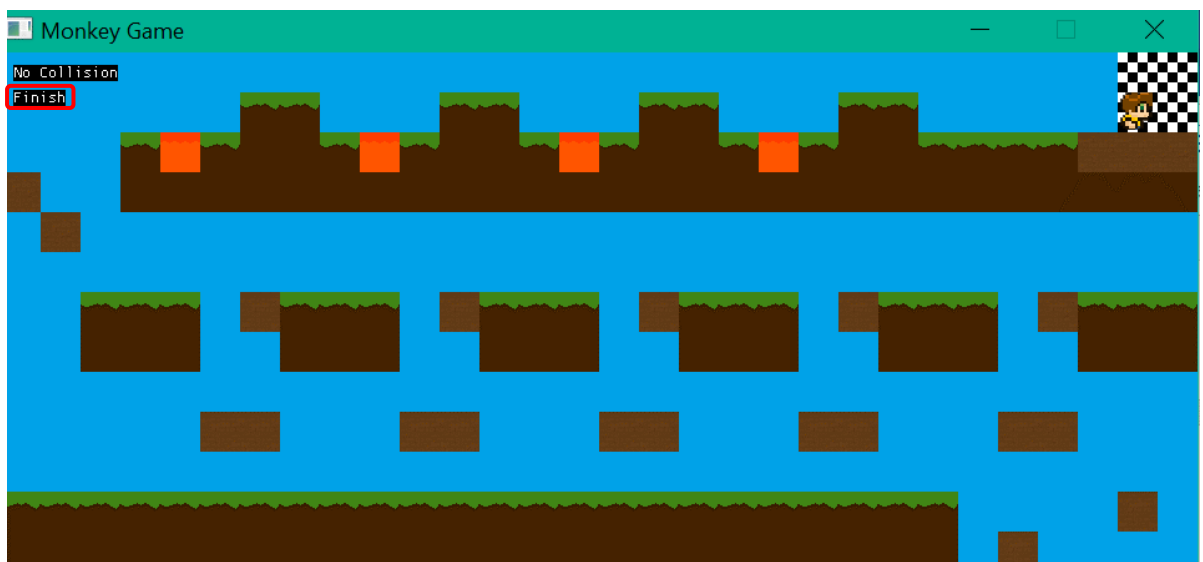
 If player.levelfinish = True Then
 DrawText("Finish", 10, 30)
 End
```

Candidate Name: [REDACTED]

Candidate Number: [REDACTED]



Moving the player into the finish line...



As expected a finish text appears when the player moves into the finish line, it also disappears when the player leaves the finish line as expected.

## 12/03/2017 LEVEL ENDING SCREEN

Now that I have implemented a finishing line I will now be coding what happens when the player actually gets to the finishing line. I am going to do this through a coordinate comparison where it checks if the player has the same coordinates as the finish line.

```

If py < 64 And px > 880 Then
 levelfinish = True
Else levelfinish = False
End

```

This basically checks that the players y position is less than 64 but also checks if the players x position is more than 880, or basically if it is within the finish line image and then sets level finish to true. This means that it can detect when then the player has finished and I can now code the response to this.

```

Case "LEVEL_1"
 Cls(0, 0, 0)
 DrawText("Level 1", 450, 10, 0.5)
 level1.draw
 DrawImage finishline, 896, 0
 player1.initialise

 If collision = True Then
 DrawText("Collision", 10, 10)
 Else collision = False
 DrawText("No Collision", 10, 10)
 End

 If player.levelfinish = True Then
 SetColor(0,0,0)
 DrawRect(230, 230, 500, 500)
 SetColor(255, 0, 0)
 DrawText("Level 1 Complete!", 480,480,0.5)
 DrawText("Press Enter to go to level selector!", 480, 500, 0.5)
 End

```

This is the main game app class in the on-render method, it completes a number of visual processes such as drawing a black rectangle, this black rectangle then has red text saying that the level is complete. I have also implemented the navigation for when the level finishes, when the level finish sign appears after they cross the finish line they should be able to go back to level selector, this is done easily.

```

Case "LEVEL_1"
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"
 If KeyHit (KEY_E) Then Error("")

 If KeyDown (KEY_RIGHT) Then
 player1.px +=(player1.pspeed)
 End

 If KeyDown (KEY_LEFT) Then
 player1.px -= (player1.pspeed)
 End

 If level1.tile_collision(player1.px, player1.py) = level1.FPlatform Then
 collision = True
 Else collision = False
 End

 If collision = True Then
 player1.px -= player1.pspeed
 End

 If collision = True Then
 player1.py += player.gravity
 player.playerjump = False And player1.py > 0
 End

 If player.levelfinish = True And KeyHit (KEY_ENTER) Then GameState = "LEVEL_SELECTOR"

```

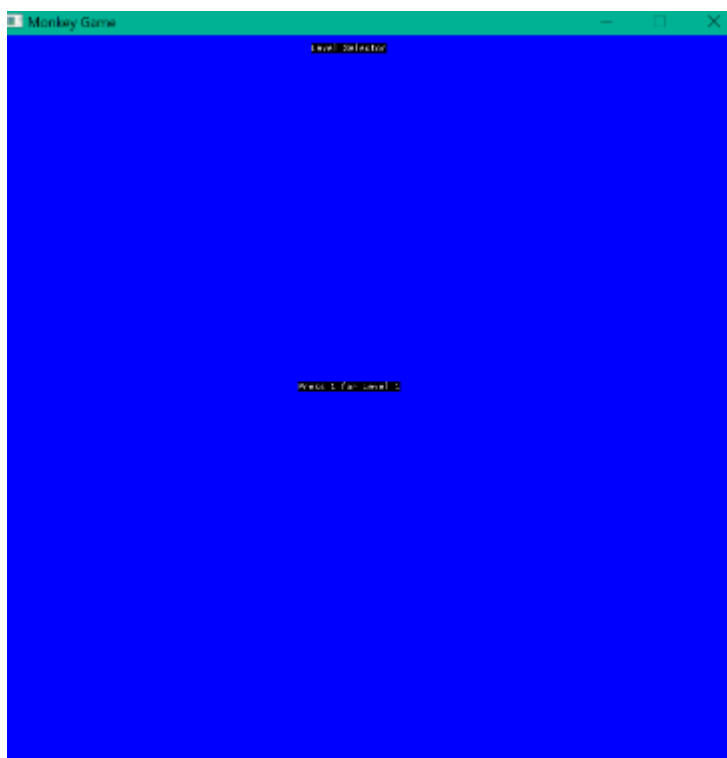
It basically checks that if the level finish is true which means the level finish screen will pop up then the player can press enter and it will take the player back to the level selector screen.

Candidate Name: [REDACTED]

Candidate Number: [REDACTED]



Pressing Enter...



## 15/03/2017 ADDING AN ENEMY

This involves creating an enemy class, I'm going to start by having an enemy which moves towards the character, they will not shoot the character but if they collide then the level will finish and the player will lose a life. I am going to start by just simply adding an enemy class which will then have a variety of functions. I have started by declaring a simple enemy class that is similar to my player class.

```
Class enemy
 Field enemy1Sprite:Image
 Field ex:Int
 Field ey:Int
 Field exspeed:Int

End
```

The class has 4 variables or attributes, this is the actual enemies image, then ex which is the enemies x position and then ey which is the player's y position, finally exspeed is the speed of the player in the y direction. Similarly, to the players code I have to create a new instance of the enemy class further up in the program, such as getting an image and setting a position, I have copied the code from creating a player because it is very similar.

```
Class Game_app Extends App

 Global GameState:String = "INITIALISE"

 Field level1:Level

 Field loadingscreen:Image

 Field p1 img:Image
 Field e1 img:Image
 Field player1:player
 Field enemy1:enemy
 Field finishline:Image
 Global collision:Bool = False

 Method OnCreate()
 SetUpdateRate 60
 loadingscreen = LoadImage ("loadingscreen.png")
 level1 = New Level
 level1.load()

 p1 img = LoadImage("sprite.png")
 e1 img = LoadImage("enemy1.png")
 finishline = LoadImage("finishline.png")
 player1 = New player
 enemy1 = New enemy
 player1.p1Sprite = p1 img.GrabImage(0, 0, 32, 32)
 enemy1.enemy1Sprite = e1 img.GrabImage(0, 0, 45, 40)

 End
```

Now we have declared the enemy class we need to deal with creating an instance of this class, I did this by getting the image from it which we grab from the enemy sprite sheet. We then need to set the ex, ey and exspeed for the instance I do this In initialize game state of my game.

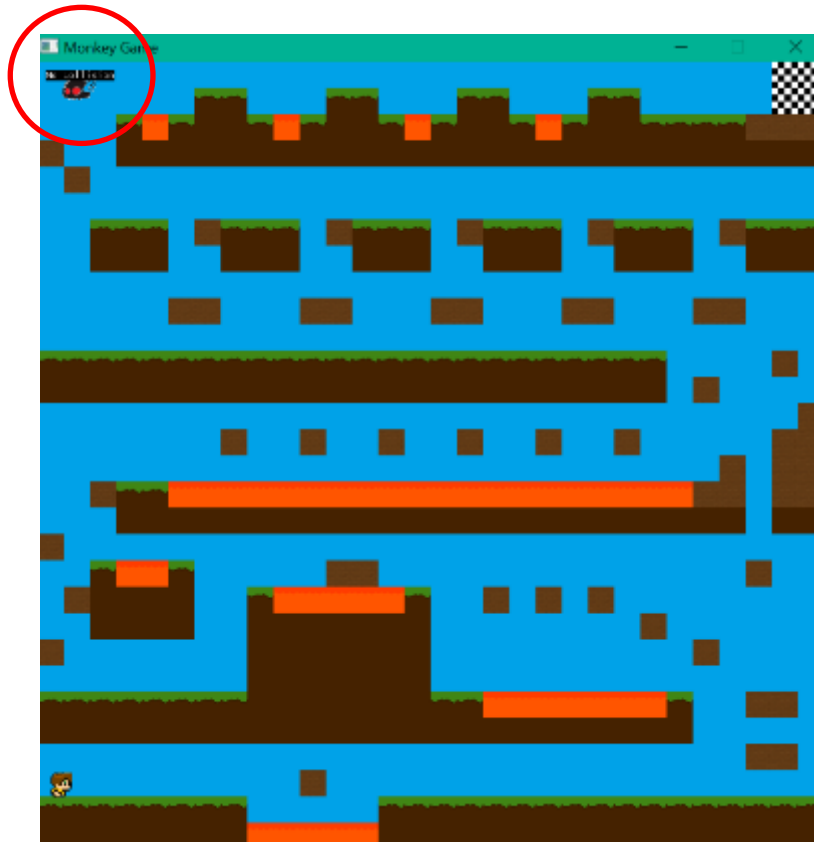
```
enemy1.exspeed = 1.0
enemy1.ex = 10
enemy1.ey = 10
```

This means the player will be at position 10, 10 and the speed is set to 1 for the moment. Then like the player class there is an initialize method in my enemy class.

```
Class enemy
 Field enemy1Sprite:Image
 Field ex:Int
 Field ey:Int
 Field exspeed:Int

 Method initialise()
 DrawImage enemy1Sprite, ex, ey
 End
End
```

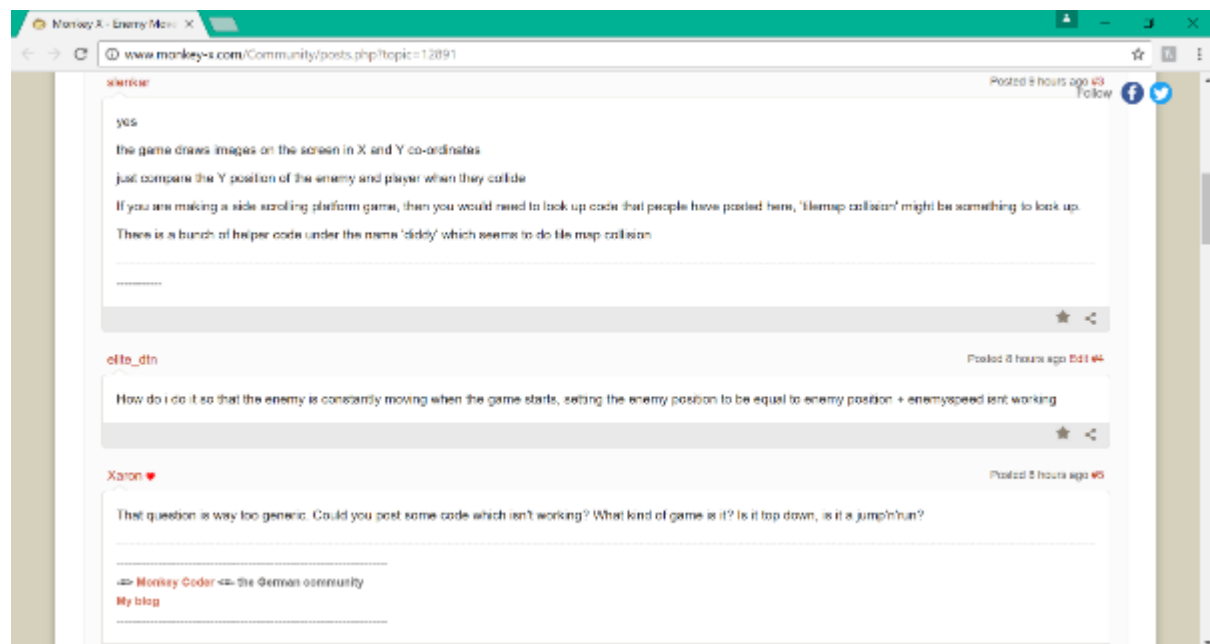
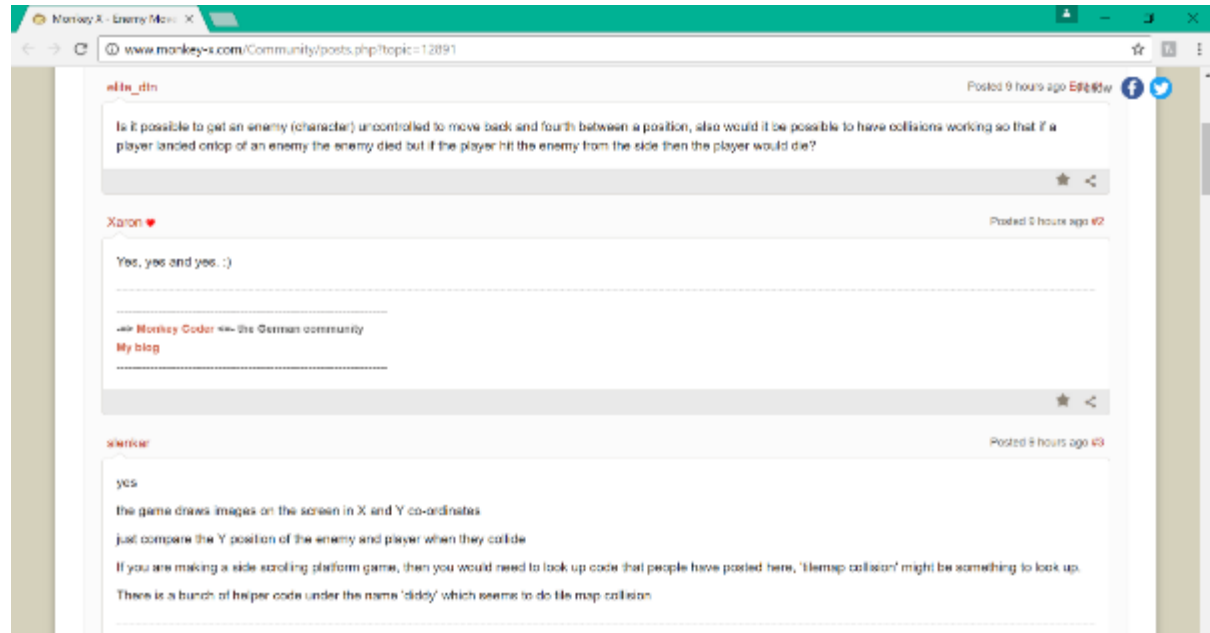
At the moment this just consists of the code for actually drawing the image to the screen, it draws the enemy at ex, ey which are declared earlier in the program as 10, 10. At the moment this won't actually draw the image because we need 'enemy1.initialise' which goes in the on render method of the main game class. The enemy sprite is now drawn to the screen.

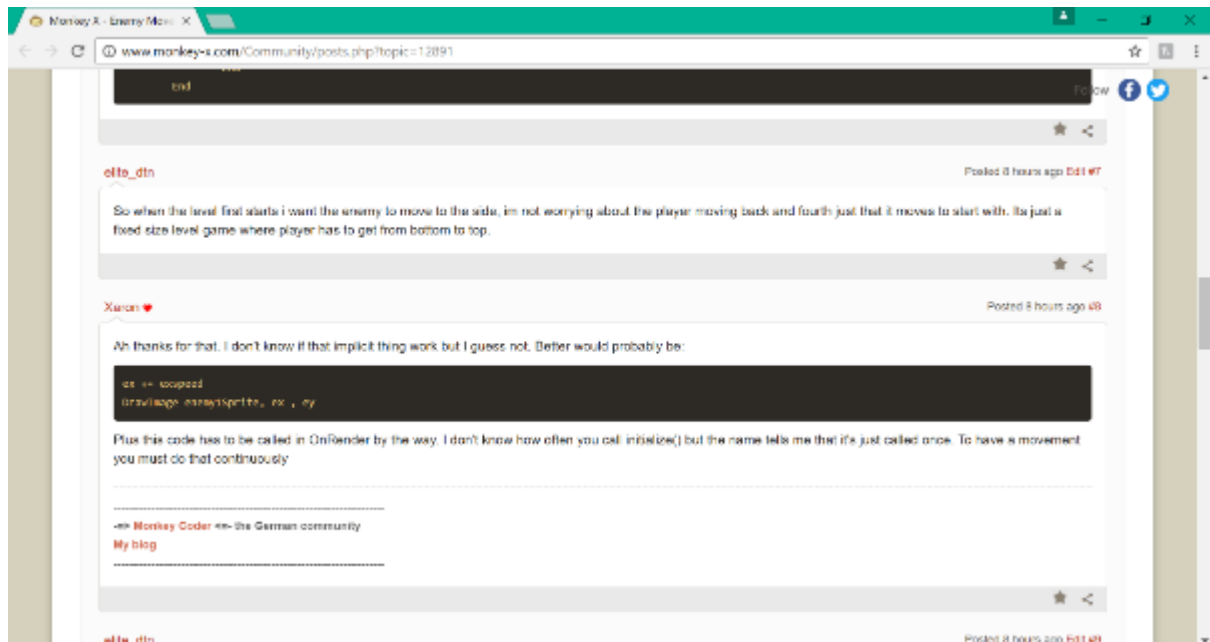




## 20/03/2017 GETTING THE ENEMY TO MOVE BACK AND FORTH

My aim was to have the enemy moving from side to side or along a platform a bit like in new super Mario bros however I wasn't sure on how to do that so I asked the question in an online forum to get some support and help on how to deal with the issue.





The line that helped the most was the 'ex += speed' I knew it was something along these lines and knew what I was trying to code but I wasn't sure on the most efficient and effective way of achieving this. I then implemented this into my program and tested whether it worked.

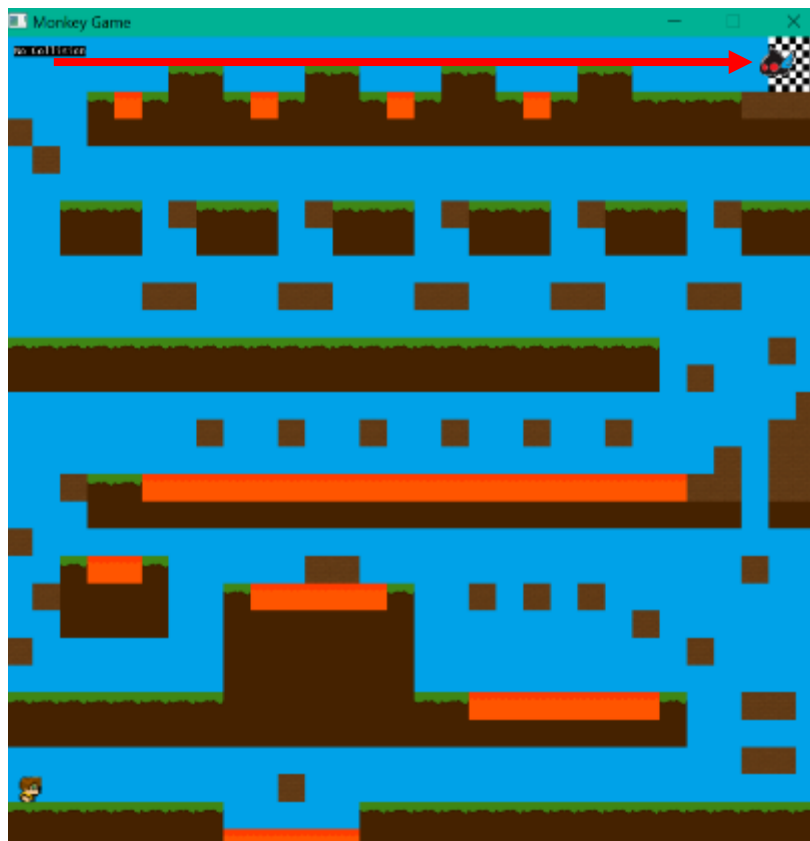
```

Class enemy
 Field enemy1Sprite:Image
 Field ex:Int
 Field ey:Int
 Field exspeed:Int

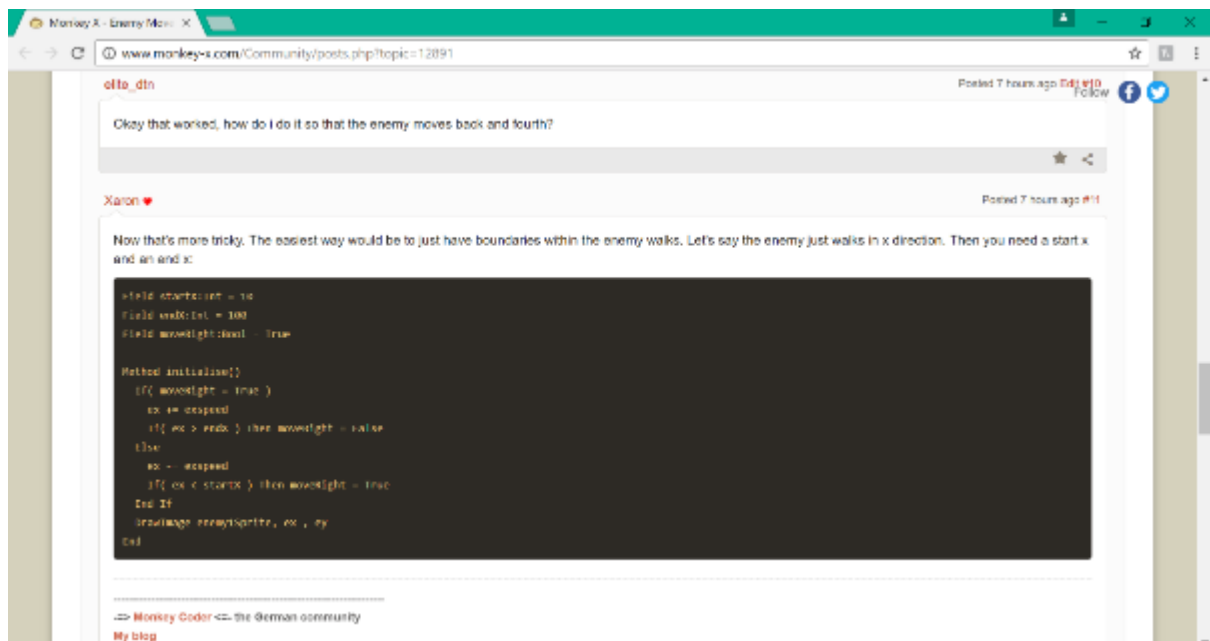
 Method initialise()
 ex += exspeed
 DrawImage enemy1Sprite, ex, ey
 End
End

```

This works so that it sets the enemies position to be the enemies position plus the enemies speed which is 1, so it is basically constantly adding 1 to the position which should result in the player moving across the screen.



Now the player moves to the screen I want to code it so that the player moves back and forth around a point like the shells do on NSMB, this was something that I was also unsure of so asked about in the forum as well.



```

Class enemy
 Field enemy1Sprite:Image
 Field ex:Int
 Field ey:Int
 Field exspeed:Int
 Field moveRight:Bool = True

 Method initialise()

 If(moveRight = True)
 ex += exspeed

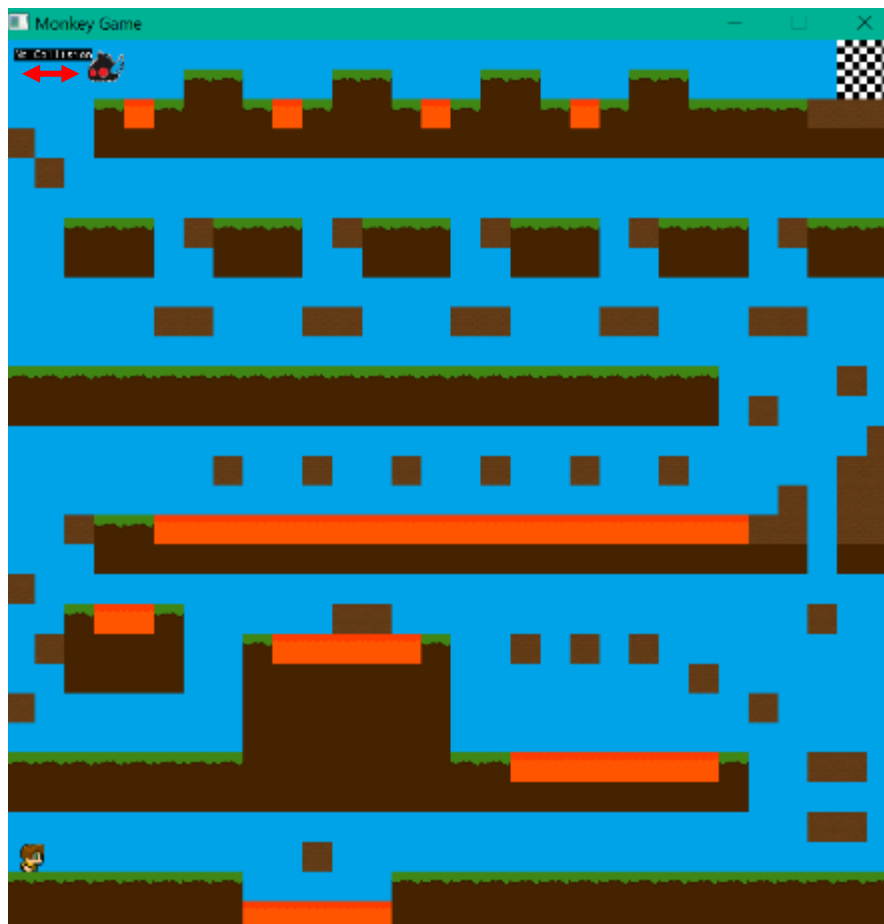
 If(ex > 100) Then moveRight = False
 Else
 ex -= exspeed

 If(ex < 10) Then moveRight = True
 End If

 DrawImage enemy1Sprite, ex, ey
 End
End

```

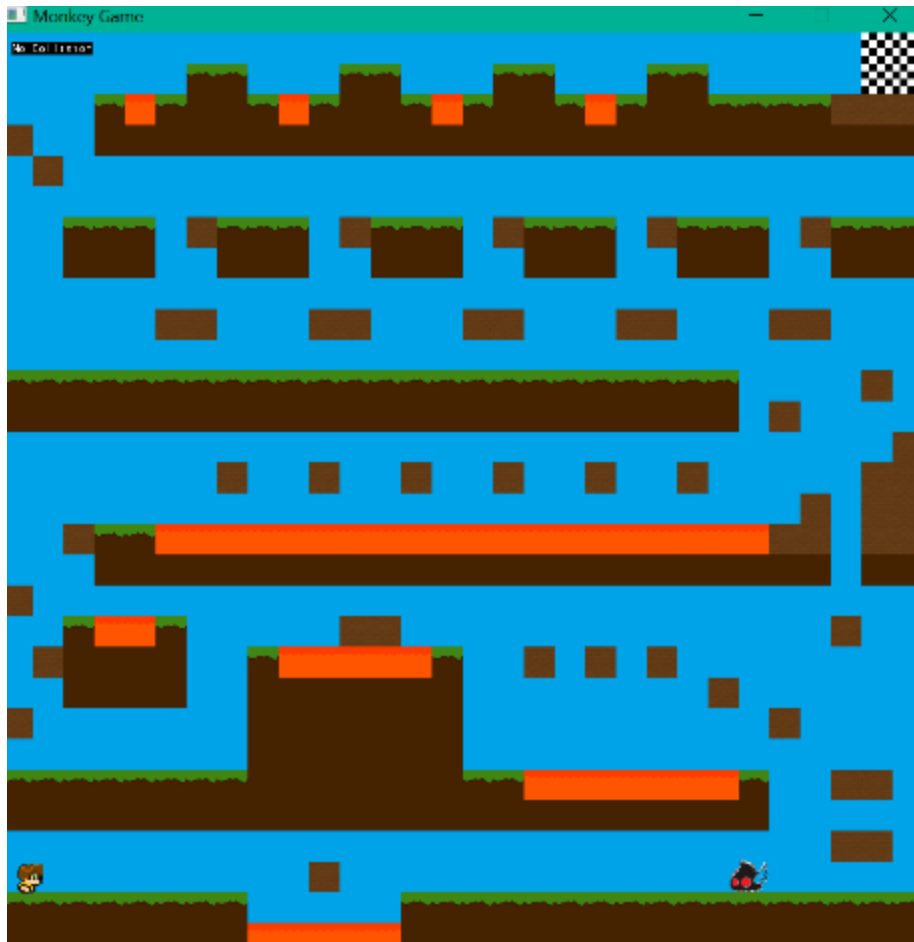
This is done simply with a new variable called moveRight which basically is used to determine which way the player is moving which helps with changing the speed and direction of the player. When the player first starts they move to the right this I had already coded, when the player gets past the coordinate of 100 so basically when they are 100 along then moveRight is set to false and the opposite rule is applied to the x position so it subtracts the enemies speed so it starts moving to the left. And then finally when the enemies x position is less than 10 moveRight is set to true and it does the same thing again, it should now do this in a constant loop so the player goes back and forth.



As expected the player moves back and forth.

## 22/03/2017 ENEMY COLLISIONS

Now that my player moves back and forth around a point on my level, I want to make it so that if the controlled player runs into the enemy then the level ends and then eventually the player will lose one of their lives when that is sorted. I'm going to start by moving the player down to the bottom area of the level, I have also realized that my level hasn't been designed completely effectively to allow the player room to jump over the enemy so this will have to be worked upon.



I have changed the enemies position to be 700, 860 which is now on the bottom platform and will be the players first obstacle. At the moment the player can go through the character however I want this to change so that it stops the level and the player loses a life, this can be done through comparing coordinates. There is an efficient way of doing it from the maze runner game which I have looked at earlier, it requires creating a function which I have called enemycollision.

```
Function enemycollision:Bool (x1:Int, y1:Int, w1:Int, h1:Int, x2:Int, y2:Int, w2:Int, h2:Int)
 If x1 >= (x2 + w2) Or (x1 + w1) <= x2 Then Return False
 If y1 >= (y2 + h2) Or (y1 + h1) <= y2 Then Return False
 Return True
End
```

This is then combined with the enemy collision if statement which goes in the level 1 game state in the on-update method:

```
Case "LEVEL_1"
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"
 If KeyHit (KEY_E) Then Error("")

 If KeyDown (KEY_RIGHT) Then
 player1.px +=(player1.pspeed)
 End

 If KeyDown (KEY_LEFT) Then
 player1.px -=(player1.pspeed)
 End

 If level1.tile_collision(player1.px, player1.py) = level1.FPlatform Then
 collision = True
 Else collision = False
 End

 If collision = True Then
 player1.px -= player1.pspeed
 End

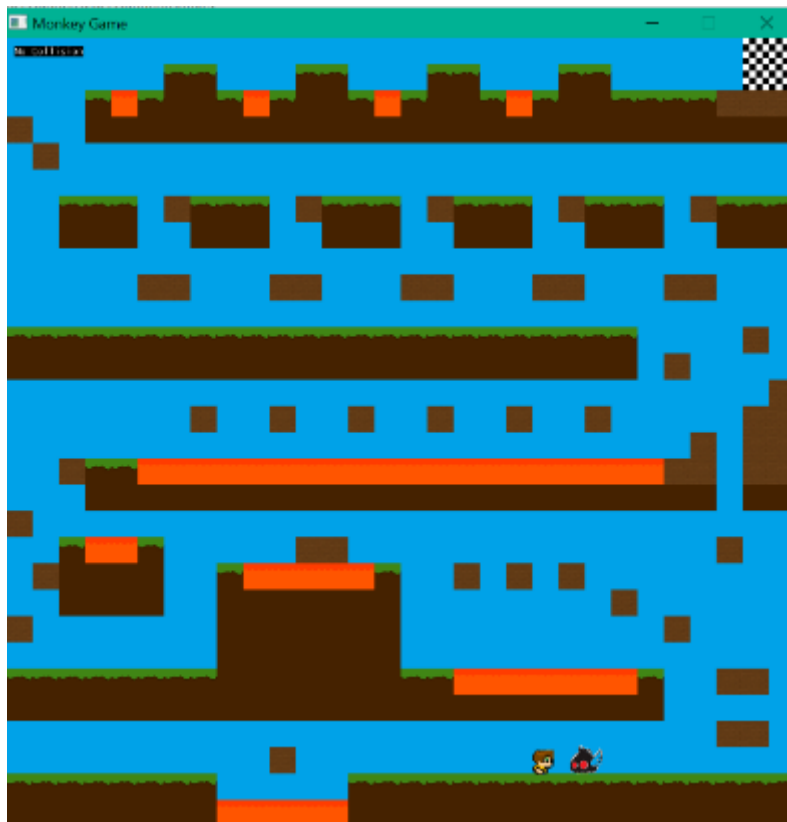
 If collision = True Then
 player1.py -= player.gravity
 player.playerjump = False And player1.py + 0
 End

 If player.levelfinish = True And KeyHit (KEY_ENTER) Then GameState = "LEVEL_SELECTOR"

 If enemycollision(player1.px, player1.py, 32, 32, enemy1.ex, enemy1.ey, 40, 45) Then GameState = "LEVEL_SELECTOR"
```

This gives the 'enemycollision' function the parameters of the px, py which are the players coordinates and size of the player (32x32 pixels) and then gives ex, ey which are the enemies coordinates and then enemies size (40x45 pixels).

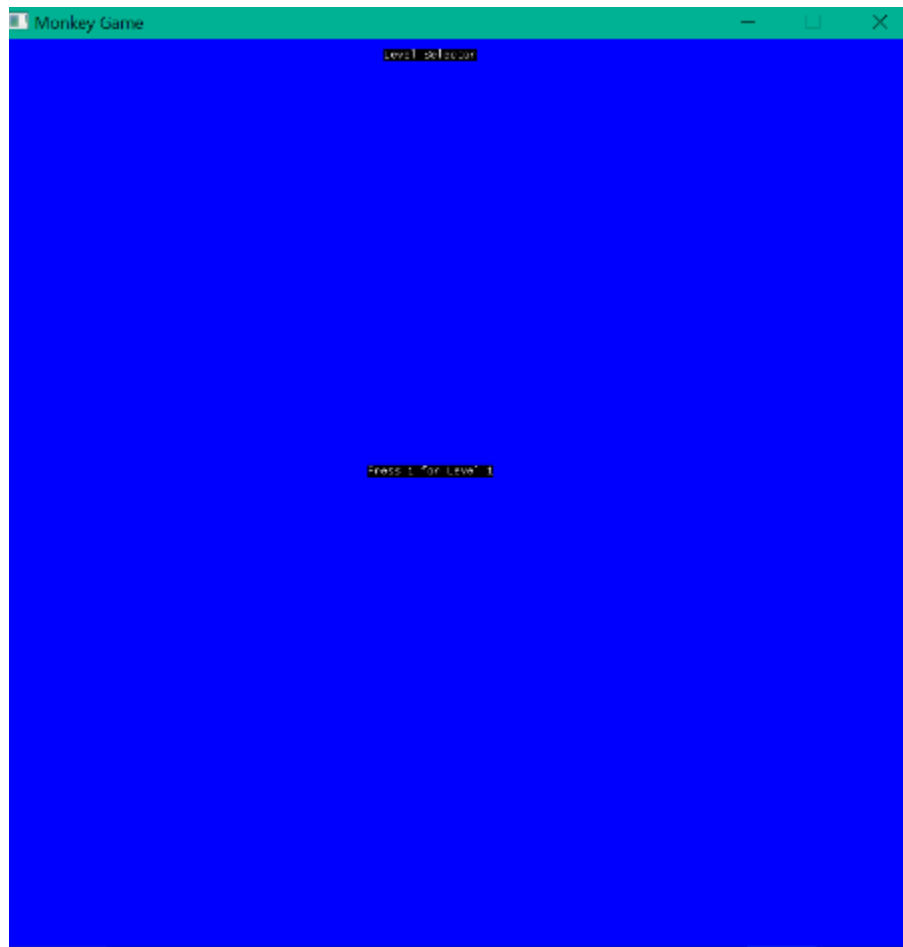
The actual function compares the two sprites coordinates and then when they are touching inside it changes the game state to the level selector.



After they collide...

Candidate Name: [REDACTED]

Candidate Number: [REDACTED]



However, I want to make it so that a message saying the player has died comes up this will be easier when it comes to lives. I have decided to make the level failing special by having a moving banner which joins together to read level failed, this is inspired from my enemy moving by itself.

Adjustment to test table

| What is the test for?                        | Explanation                                                                                                | Expected Outcome                               |
|----------------------------------------------|------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| Player should die after collision with enemy | Enemies will appear where declared in level code, the player should die after the collide with the player. | Player collides with enemy and level finishes. |

## 23/03/2017 ADDING A LEVEL FAILED BANNER

My idea was to have a moving banner across the screen from when the player dies, this was then expanded on to be two pictures that joined together to make the banner. I started by making the image and braking it into two 500x150 images as shown below.



I decided the easiest way was to load the images into the game using the code I have used before with other images such as my loading screen, however to make things easier I have made a class for the whole banner. It has 5 attributes, which are the x and y coordinates for both banners and then the banner speed which will make the banner move across the screen.

```
Class levelfailedbanner
 Field ban1_x:Int = 230
 Field ban2_x:Int = 230
 Field ban1_y:Int = -150
 Field ban2_y:Int = 960
 Field banspeed:Int = 5.0

End
```

Then in the main game class is where the code for loading the image from the data file of my game, this like the other images requires only a few lines of code.

```
Global GameState:String = "INITIALISE"

Field level1:Level
Field loadingScreen:Image
Field pl_img:Image
Field el_img:Image
Global ban1_img:Image
Global ban2_img:Image
Field player1:player
Field enemy1:enemy
Field levelfailedbanner:levelfailedbanner
Field finishline:Image
Global collision:Boolean = False

Method OnCreate()
 SetUpdateRate 60
 loadingScreen = LoadImage ("loadingScreen.png")
 ban1_img = LoadImage ("levelfailed1.png")
 ban2_img = LoadImage ("levelfailed2.png")
 level1 = New Level
 level1.load()

 pl_img = LoadImage("sprite.png")
 el_img = LoadImage("enemy1.png")
 finishline = LoadImage("finishline.png")
 player1 = New player
 enemy1 = New enemy
 player1.plSprite = pl_img.GrabImage(0, 0, 32, 32)
 enemy1.enemySprite = el_img.GrabImage(0, 0, 48, 48)

End
```



This creates two image variables for my two images, they are loaded from the data file in the same way as all the other images have been. Next I need to create an initialize method inside the banner class so that the actual images can be displayed, this is similar to what is done with the player and enemy classes, I can then just call the method at the appropriate time for it to be used.

```
Method initialise()
 DrawImage Game_app.ban1_img, ban1_x, ban1_y
 DrawImage Game_app.ban2_img, ban2_x, ban2_y
End
```

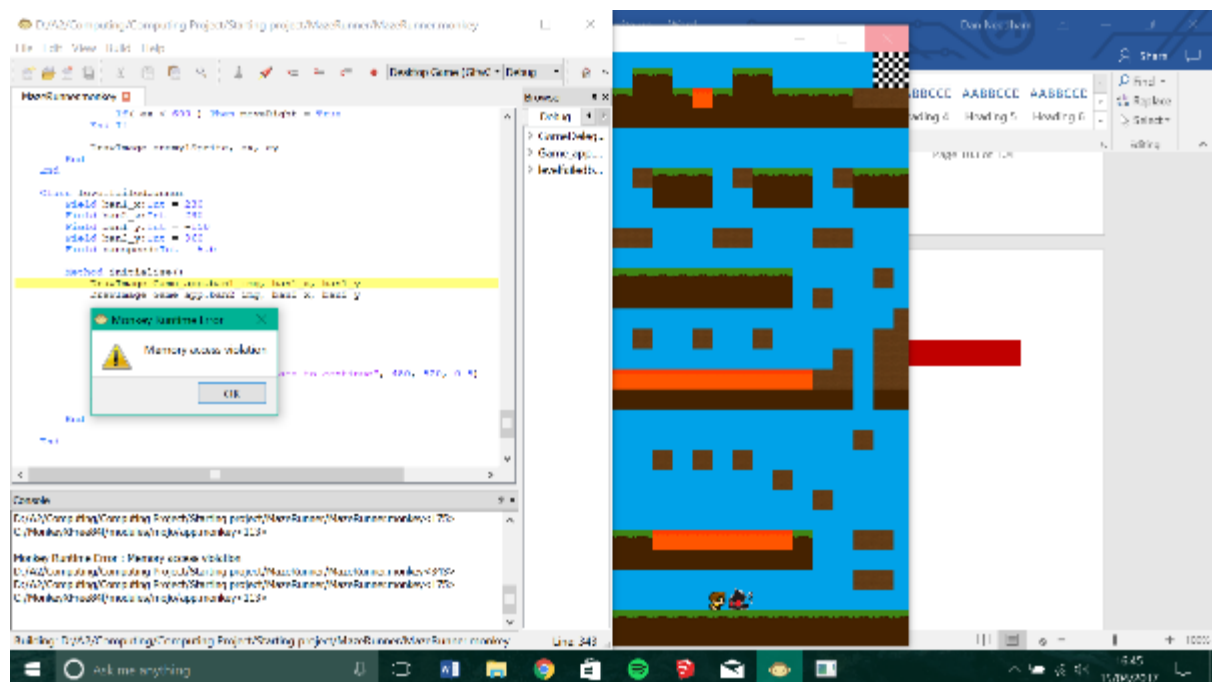
This draws the image at the coordinates, ban1\_x and ban1\_y for the first image, these are stated when I declared the variables, similarly the second image is drawn at ban2\_x, ban2\_y also stated earlier. All that is left to call the method initialize, just to test it I have called it as soon as the level starts.

```
Case "LEVEL 1"
 Cls(0, 0, 0)
 DrawText("Level 1", 450, 10, 0.5)
 levell.draw
 DrawImage finishline, 896, 0
 player1.initialise
 enemy1.initialise
 levelfailedbanner.initialise

 If collision = True Then
 DrawText("Collision", 10, 10)

 Else collision = False
 DrawText("No Collision", 10, 10)
 End
```

However after compiling the game, this result in a 'memory access violation' error.



However because I realized that most of the variables were being accessed from elsewhere in the program I made every variable in relation to the banner; ban1\_x, ban2\_x, ban1\_y, ban2\_y, banspeed, ban1\_img, ban2\_img a global variable which solved the problem meaning the images were drawn to the screen.

However, I wanted to get them to move across the screen and then create the final image when the player died, this required creating a new Boolean expression variable in the player called 'playerDead'.

```
Class player
 Field piSprite:Image
 Field px:Int
 Field py:Int
 Field pspeed:Int
 Field pxspeed:Int
 Field pyspeed:Int
 Global playerDead:Bool = False
 Global playerjump:Bool = False
 Global gravity:Float = 0
 Global levelfinish:Bool = False
```

This will basically be used to determine whether or not the player is dead and therefore whether or not the banner needs to be drawn to the screen, it is set to false by default because the player won't be dead when the game starts. This is going to be changed to true when the player collides with the enemy and therefore needs to be triggered in response to the collision.

```
Case "LEVEL_1"
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"
 If KeyHit (KEY_E) Then Error("")

 If KeyDown (KEY_RIGHT) Then
 player1.px +=(player1.pspeed)
 End

 If KeyDown (KEY_LEFT) Then
 player1.px -=(player1.pspeed)
 End

 If level1.tile_collision(player1.px, player1.py) = level1.FPlatform Then
 collision = True
 Else collision = False
 End

 If collision = True Then
 player1.px -= player1.pspeed
 End

 If collision = True Then
 player1.py -= player.gravity
 player.playerjump = False And player1.py + 0
 End

 If player.levelfinish = True And KeyHit (KEY_ENTER) Then GameState = "LEVEL_SELECTOR"

 If enemycollision(player1.px, player1.py, 32, 32, enemy1.ex, enemy1.ey, 40, 45) Then player.playerDead = True
```

So when there is an enemy collision then player.playerDead is set to true, I can now code what happens in response to this in my banner class. This is done with a simple if statement.

```
Class levelfailedbanner
 Global ban1_x:Int = 230
 Global ban2_x:Int = 230
 Global ban1_y:Int = -150
 Global ban2_y:Int = 960
 Global banspeed:Int = 5.0

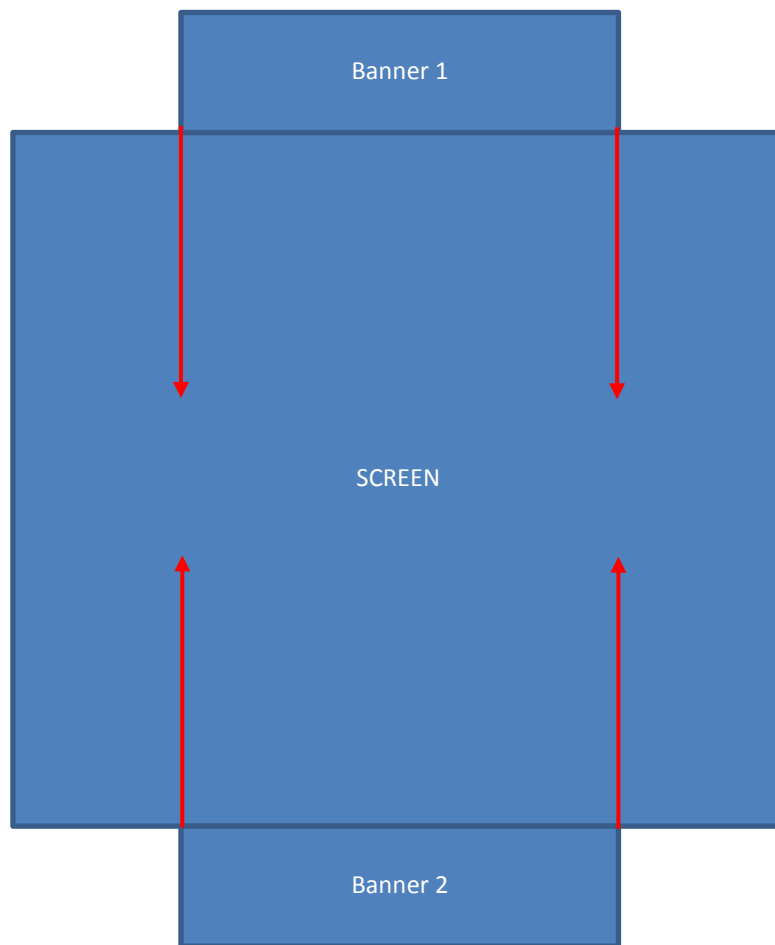
 Method initialise()
 DrawImage Game_app.ban1_img, ban1_x, ban1_y
 DrawImage Game_app.ban2_img, ban2_x, ban2_y

 If player.playerDead = True Then
 ban1_y += banspeed
 ban2_y -= banspeed
 If (ban1_y = 330) Then
 banspeed = 0
 DrawText("Press backspace to continue", 480, 520, 0.5)
 Endif
 End

 End

End
```

When the player is dead then the first banner which is above the screen starts to move downwards onto the screen and similarly the second banner which is below the screen starts to move upwards.



```
If (ban1_y = 330) Then
 banspeed = 0
 DrawText("Press backspace to continue", 480, 520, 0.5)
Endif
```

Finally this stops the picture when it gets so that they are both joined and in the middle of the screen, it only needs to be once because they are symmetrical this just stops both of them at the same time. Finally is calling the initialize method of the banner class in the right place of the game app, this goes in the on render method in the level 1 game state but in response to an if statement of if the player is dead.

```
If player.playerDead = True Then
 levelfailedbanner.initialise
 player1.pspeed = 0
 enemy1.exspeed = 0
End
```

This also sets the player speed and enemy speed to 0 so that they stop moving.

## 25/03/2017 TESTING LEVEL FAILED BANNER

---

Now that I have coded the level failed banner I need to see that it works correctly, after colliding the player with the enemy:



The two banners move up and down the screen respectively, and when they get to the middle of the screen they stop correctly to join and make the final image.



I have also added some navigation from this level failed state where the player can return to the level selector screen and restart the level.

## 28/03/2017 IMPROVING MENUS

---

Because I already have a lot of the barebones of the menus working this should be easy, I had already designed the screens so it was a matter of loading the image and then just drawing it to the screen like I have been doing with all the other images.

Firstly, I have started by changing the size of the loading screen and adding the level image to it to improve the appearance of the game. Originally the image wasn't big enough for the screen caused by me frequently changing the size of the game window, however now I have confirmed it to be 960 x 960 I have changed the image size in paint and added an image of the level to the screen. This requires no change in the game code as you just overwrite the old image in the game data file and it will now draw this new image.



Next I have improved the menu using the screen I had designed in the development stage, I have just copied that image and then using the same code as loading screen added it to my game.

```
Field loadingscreen:Image
Field levelselector:Image
Field mainmenu:Image

Field p1_img:Image
Field e1_img:Image
Global ban1_img:Image
Global ban2_img:Image
Field player1:player
Field enemy1:enemy
Field levelfailedbanner:levelfailedbanner
Field finishline:Image
Global collision:Bool = False

Method OnCreate()
 SetUpdateRate 60
 loadingscreen = LoadImage ("loadingscreen.png")
 levelselector = LoadImage ("levelselector.png")
 mainmenu = LoadImage ("mainmenu.png")
 ban1_img = LoadImage ("levelfailed1.png")
 ban2_img = LoadImage ("levelfailed2.png")
 level1 = New Level
 level1.load()
```

It creates an image variable called main menu, the image is then loaded using this variable, then all I have to do is draw this image to the screen at the appropriate time.

```
Method OnRender()
 Select GameState
 Case "LOADING_SCREEN"
 DrawImage loadingscreen, 0, 0
 Case "MENU"
 DrawImage mainmenu, 0, 0
 DrawText("Press L", 235, 175, 0.5)
 DrawText("Press V", 235, 280, 0.5)
 DrawText("Press C", 235, 385, 0.5)
 DrawText("Press E", 235, 490, 0.5)
 DrawText("Press I", 235, 595, 0.5)
 DrawText("Press ESC", 770, 875, 0.5)
```

This results in the image being drawn to the screen. Because I have not done proper navigation such as being able to click or use arrow keys I have letters for moving to the next screens from the main menu as shown above. It would be too complex to have the original red box system I wanted for selected options and then pressing enter would go to the selected screen, this therefore isn't possible at the moment.



I have also added some more navigation where the player can now exit the game by pressing escape, this is similar to the code for level 1 so I have just been able to recycle it. So now when the player is on the menu they can click escape and the game will close.

```
Method OnUpdate()
 Select GameState
 Case "INITIALISE"
 player1.px = 10
 player1.py = 565
 player1.pspeed = 2.0
 player1.pxspeed = player1.pspeed
 player1.py speed = 0

 enemy1.exspeed = 1.5
 enemy1.ex = 700
 enemy1.ey = 880

 GameState = "LOADING_SCREEN"

 Case "LOADING_SCREEN"
 If KeyHit (KEY_SPACE) Then GameState = "MENU"

 Case "MENU"
 If KeyDown (KEY_L) Then GameState = "LEVEL_SELECTOR"
 If KeyDown (KEY_V) Then GameState = "LEVEL_VIEWER"
 If KeyDown (KEY_C) Then GameState = "CHARACTER_VIEWER"
 If KeyDown (KEY_E) Then GameState = "CHARACTER_EDITOR"
 If KeyDown (KEY_I) Then GameState = "INSTRUCTIONS"
 If KeyDown (KEY_ESCAPE) Then Error("")
```



Next I want to add a level selector screen which I had designed in the design stage of my game development, this again is as easy as the menu and loading screen have been and uses exactly the same code.

```

Class Game_app Extends App

 Global GameState:String = "INITIALISE"

 Field level1:Level

 Field loadingscreen:Image
 Field levelselector:Image
 Field mainmenu:Image

 Field pl_img:Image
 Field e1_img:Image
 Global ban1_img:Image
 Global ban2_img:Image
 Field player1:player
 Field enemy1:enemy
 Field levelfailedbanner:levelfailedbanner
 Field finishline:Image
 Global collision:Bool = False

 Method OnCreate()
 SetUpdateRate 60
 loadingscreen = LoadImage ("loadingscreen.png")
 levelselector = LoadImage ("levelselector.png")
 mainmenu = LoadImage ("mainmenu.png")

```

In the main on render method...

```

Case "LEVEL_SELECTOR"
 DrawImage levelselector, 0, 0
 DrawText("Press 1 for Level 1", 220,400,0.5)

```

This draws the image to the screen...





### 30/03/2017 EXPANDING UPON COLLISIONS

---

I have already got the collisions working for the floating platforms but now I want them to work for some of the other platforms, I would like the base platforms to have collisions so the player can legally progress up the level without it being completely glitchy. After looking in to improving upon the collisions I am not sure without some considerable code that it would be possible to do this.

### 17/04/2017 RESETTING OBJECTS FOR RESTARTING A LEVEL

---

Now that I have coded a failed level system and also a winning level system, I want to make it so that the player can restart the level so that it is completely 'new' again. The problem is when it goes back to the level selector all the players and objects are in the position they were in when the level ended, so this might be with the level finished banner up and player's positions disabled or with the level completed banner. What I need to do is have it so that all the positions and object positions. However due to how my game is laid out and coded this is going to be tricky, a lot of the positions and images for both the enemy and player have been code outside of their classes which makes it hard to create a new instance of them when I want to create a new level which would be the easiest way of resetting everything.

**D. EVALUATION****TEST TABLE****ENEMIES AND CHARACTERS**

| Test Video No. | What is the test for?                                               | Explanation                                                                                                                                   | Expected Outcome                                                               | Requirement Met?                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------|---------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.1            | Enemies should appear.                                              | Enemies will appear where declared in level code, harder levels will have more enemies.                                                       | Enemy should appear.                                                           | Yes; I have managed to get the enemy to draw to the screen, there is only 1 on level 1 however now it should be far easier to add more and more. Because I didn't get around to creating more levels I therefore have not shown or done the increasing difficulty of the enemies. I have however created a level failed banner which appears when the enemy hits the player thus ending the level. |
| 1.1            | Character does not fall through the floor.                          | Character should not fall through floor when they move.                                                                                       | Character will stay fully above the platform.                                  | Yes; When the level starts, the player is held by the floor and doesn't fall through the floor.                                                                                                                                                                                                                                                                                                    |
| 1.2            | Character shouldn't fall through if they land on elevated platform. | If character jumps and lands on one of the elevated boxes / platforms they shouldn't fall through, should only fall if nothing is below them. | Character should stay in position on platform.                                 | Yes; The player does not fall through the base platform and doesn't fall through the floating platform.                                                                                                                                                                                                                                                                                            |
| X              | Enemy shooting bullets                                              | Enemy should shoot bullets horizontally from current position at a fire rate of a bullet every 2 seconds.                                     | Enemy shoots bullet                                                            | No; Because I didn't get around to implementing power-ups it would've made the enemy shooting bullets unfair. The enemy can still kill the player but only by running into the player. I therefore don't have the enemy shooting bullets yet.                                                                                                                                                      |
| X              | Character collisions with enemy bullet.                             | If character in collision with bullet, then level ends. Unless giant and invincible power up detected.                                        | If collision detected between bullet and character level ends and lose a life. | No; Because the enemy can't shoot a bullet yet, then there can be no collision between character and enemy bullet. However, collisions between enemy and player have been implemented.                                                                                                                                                                                                             |

|     |                                               |                                                                                                            |                                                                         |                                                                                                                                                                                             |
|-----|-----------------------------------------------|------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.3 | Player should die after collision with enemy  | Enemies will appear where declared in level code, the player should die after the collide with the player. | Player collides with enemy and level finishes.                          | Yes; When the player collides with the enemy the level finishes as expected, there is a level finished banner that appears. However, I haven't got a lives system working.                  |
| X   | Character falling in obstacle (water or fire) | If character falls into obstacle, then level ends. Unless giant or invincible power up detected.           | Level ends and player loses a life if character falls into an obstacle. | No; Collisions have been added but I didn't get around to adding collisions with the lava due to time constraints however with the collision system in place this would be very easy to do. |

## MENUS

| Test Video No. | What is the test for?                   | Explanation                                                                                                        | Expected Outcome                                       | Requirement Met?                                                                                                                                                                          |
|----------------|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2.1            | Game starts and loading screen appears. | The loading screen for the game should appear when the game is compiled.                                           | Loading screen appears.                                | Yes; When the game first compiles and loads up the loading screen appears.                                                                                                                |
| 2.2            | Main Menu appears                       | Main menu should appear when the space button on loading screen is pressed.                                        | Main Menu with options appears.                        | Yes; After pressing spacebar on the loading screen, it then goes to the main menu as expected.                                                                                            |
| 2.2            | User can navigate menu                  | User should be able to use arrow keys to navigate menus.                                                           | Arrow keys allow different options should be selected. | Yes; I have some of the screens working with simplified navigation, however navigation such as using the arrow keys is not possible, I didn't get to implementing it.                     |
| X              | Red box around current selected option. | When the arrow keys are used to select different options then a red box should appear around current selected one. | Red box around currently selected option.              | No; Because I have not had the time to improve the navigation I haven't got to advanced parts such as a red box around the selected choice. This would be relatively hard to add as well. |

|     |                                   |                                                                                       |                                                         |                                                                                                                                                                                                                                                                                                                                                           |
|-----|-----------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2.3 | Exit button should close the game | The exit button on the main menu should close the game.                               | The exit button on the main menu should close the game. | Yes; I have the exit button working for level 1 so when the player clicks 'E' the game closes. This is very easy to implement and I could do it for more of the screens. I haven't got an exit button as expected, buttons don't exist in Monkey X making navigation hard to implement.                                                                   |
| 2.4 | Display level selector screen.    | Clicking enter on level selector button should display the level selector screen.     | Level selector screen appears.                          | Yes; Pressing L on the main menu displays the level selector screen, navigation is not as I had planned but still to a high quality. Because I haven't come up with the red box selection system for menus I haven't implemented it into my game. This means that some of the expected navigation I had planned for in the design have not been possible. |
| 2.4 | Display level viewer screen       | Clicking enter on level viewer button should display the level viewer screen.         | Level viewer screen appears.                            | Yes; Despite the fact you have to click V, the level viewer screen appears. This is not as I had expected originally when designing the game.                                                                                                                                                                                                             |
| 2.4 | Display character selector        | Clicking enter on character selector button should display character selector screen. | Character selector screen appears.                      | Yes; Despite the fact you have to click C, the character selector screen appears. This is not as I had expected originally when designing the game.                                                                                                                                                                                                       |
| 2.4 | Display character editor          | Clicking enter on character editor button should display character editor screen.     | Character editor screen appears.                        | Yes; Despite the fact you have to click E, the character editor screen appears. This is not as I had expected originally when designing the game.                                                                                                                                                                                                         |
| 2.4 | Display instructions              | Clicking enter on instructions button should display the instructions screen.         | Instructions screen appears.                            | Yes; Despite the fact you have to click I the instruction screen appears. This is not as I had expected originally when designing the game.                                                                                                                                                                                                               |
| 2.4 | Back Button                       | The back button on all of these screens should go back to the main menu.              | Back button goes back to main menu.                     | Yes; I haven't added a back button to each of these screens, however pressing backspace on each of these screens takes it back to the main menu where they can access the other screens.                                                                                                                                                                  |

## LEVELS

| Test Video No. | What is the test for?                            | Explanation                                                                              | Expected Outcome                                                    | Requirement Met?                                                                                                                                                                                                                                 |
|----------------|--------------------------------------------------|------------------------------------------------------------------------------------------|---------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3.1            | Level selector shows 5 levels, 2 – 5 are locked. | When the game starts the first level should be unlocked and the others should be locked. | Level 1 unlocked<br>Level 2 to 5 locked                             | No; I have only programmed level 1 so far, despite adding the other levels as options for the level viewer there is only 1. This means that I haven't had to implement the method for filtering and blocking the other levels from being opened. |
|                | Selected locked levels results in error.         | It should not be possible to play any of the locked levels.                              | Clicking locked level should do nothing, level shouldn't be loaded. | No; Because I don't actually have the system for locked levels or the others levels so there is no need to have it.                                                                                                                              |
| 3.2            | Clicking level 1 should load level.              | Level 1 which is unlocked when game starts should load.                                  | Level 1 loaded into array.                                          | Yes; Player can click 1 on the level selector screen in order to load the first level.                                                                                                                                                           |

## POWER-UPS, TIMER, COINS

| What is the test for?                                     | Explanation                                                                                                                                       | Expected Outcome                                                                    | Requirement Met?                                        |
|-----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------|
| Coins are loaded and appear randomly.                     | Coins should appear in random places on base platforms and floating platforms. (Around 100 will appear).                                          | Coins appear when level starts                                                      | No; I have not coded a system for coins in my game.     |
| Coins should disappear when they are collected.           | Coins should disappear when character goes through them to stop them being collected again.                                                       | Coin should disappear.                                                              | No; I have not coded a system for coins in my game.     |
| Power ups should be loaded and appear randomly.           | Power ups should appear in random places on the level, should be a level of difficulty collecting them.                                           | Power ups appear.                                                                   | No; I have not coded a system for power-ups in my game. |
| Power up should disappear when collected                  | The power ups should disappear when the character goes through one and collects it.                                                               | Power ups should disappear.                                                         | No; I have not coded a system for power-ups in my game. |
| Timer set to 400                                          | Timer should count down from 400 (1 = 1 second).                                                                                                  | Timer set to 400 and counts down.                                                   | No; A timer has not be added to my game.                |
| Giant power up increases size of the character.           | When giant power up collected and used the character's size will increase.                                                                        | Character becomes giant with giant power up.                                        | No; I have not coded a system for power-ups in my game. |
| Invincible power up makes character invincible.           | When invincible power up collected and used the characters colour changes and player will be immune to all ways of dying.                         | Character becomes invincible.                                                       | No; I have not coded a system for power-ups in my game. |
| Shooter power up allows character to shoot bullets.       | When shooter power up collected and used the character can shoot bullets by pressing enter button. At increased fire rate in comparison to enemy. | Character is able to shoot bullets by pressing enter.                               | No; I have not coded a system for power-ups in my game. |
| Cannot collect more power ups if one is stored or in use. | If the character has a power up or is using one then they shouldn't be able to collect, character should go straight through it.                  | Character cannot collect another power up if they have one to use or are using one. | No; I have not coded a system for power-ups in my game. |

|                                                                  |                                                                                                                                                                |                                                                         |                                                                                                     |
|------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| Power up lasts for 20 seconds when collected.                    | When the power up is collected it will last for 20 seconds giving the character the extra power.                                                               | Power up lasts for 20 seconds before ending.                            | No; I have not coded a system for power-ups in my game.                                             |
| Coin collected                                                   | When a coin is collected validly then coin score incremented by 1.                                                                                             | Coin collected = Coin Score + 1                                         | No; I have not coded a system for coins in my game.                                                 |
| Timer runs out then level should end.                            | When the timer runs out and the person has not completed the level then level will be kept as uncompleted, it will go back to level selector and live is lost. | Timer runs out level ends and life is lost.                             | No; A timer has not be added to my game.                                                            |
| Timer and coins should be displayed changing in the scoring bar. | The scoring bar should update to show the coin score and the time left (counting down).                                                                        | The coin score and timer should update in the score bar.                | No; A timer has not be added to my game. I have not coded a system for coins in my game.            |
| Finishing level, timer stops and score calculated.               | When the player completes the level then timer stops and overall score is calculated with calculation -> score = timer x coin score.                           | Timer stops when level finishes and overall score correctly calculated. | No; A timer has not be added to my game. I have not coded a system for coins or scoring in my game. |
| Scoreboard updated                                               | The scoreboard should be updated once a level is correctly completed. Scores will need to be compared to see which is biggest.                                 | Scoreboard correctly updated when level completed.                      | No; A score system has not be added to my game.                                                     |
| Player gets 2 lives if level completed.                          | If the level is completed within time, then the player gets two lives.                                                                                         | Player gets two extra lives and it should be added to live count.       | No; I have not coded a lives system.                                                                |

## SOUNDS

| What is the test for?                                          | Explanation                                                                   | Expected Outcome                                             | Requirement Met?                                                                            |
|----------------------------------------------------------------|-------------------------------------------------------------------------------|--------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| When player jumps there should be a jumping sound.             | As requested by Robbie there will be a sound for jumping.                     | Sound plays when character jumps (presses spacebar).         | No; I have not got round to adding any sound to my game and thus this hasn't been possible. |
| When character shoots bullet there should be a shooting sound. | As requested by Robbie shooting with the bullet power up should have a sound. | Sound plays when enter button pressed with shooter power up. | No; I have not got round to adding any sound to my game and thus this hasn't been possible. |

## MOVEMENT / CONTROLS

| What is the test for?                            | Explanation                                                                                         | Expected Outcome                                       | Requirement Met?                                                                  |
|--------------------------------------------------|-----------------------------------------------------------------------------------------------------|--------------------------------------------------------|-----------------------------------------------------------------------------------|
| Left arrow moves character to the left.          | Left arrow button moves character a tile to the left as long as there is no obstacle like a wall.   | Left arrow moves character to the left.                | Yes; Movement works so pressing the left arrow moves the character to the left.   |
| Right arrow moves character to the right.        | Right arrow button moves character a tile to the right as long as there is no obstacle like a wall. | Right arrow moves character to the left.               | Yes; Movement works so pressing the right arrow moves the character to the right. |
| Jump button (spacebar) causes character to jump. | The jump button is pressed characters' y coordinates should increase.                               | The character will jump and fall if no platform below. | Yes; Pressing spacebar makes the character jump and then fall.                    |



Candidate Name: [REDACTED]

Candidate Number: [REDACTED]

---

## USER SIGN OFF AGREEMENT

### INTERVIEW WITH CLIENT

---

Dan

Robbie

Q. Are you happy with what has been achieved in the initial development?

A. I am happy with what has been achieved, however I am also concerned with some of the features that haven't been included in comparison to our agreement.

Q. Would you want the program to be further developed?

A. Yes, there are features such as a timer, coins, scoring, lives and power-ups which haven't even been attempted which reduce the quality of the game drastically and makes the game relatively unplayable.

### SIGNATURE OF APPROVAL

---

Sign below if happy with what has been achieved and that the program is to the appropriate level and standard agreed at prior meeting during design.

[REDACTED]

Robbie Needham

---

## EVALUATION OF SOLUTION

### MENU'S

---

#### USABILITY FEATURES

- When game is started the loading screen will appear and be loaded, the player will then have to click space to go to the main menu.
- On the main menu there are a number of options, consisting of 5 buttons: Level selector, Level Viewer, Character selector, Character editor, instructions.
- Clicking on the level selector is the more important button because this is the only thing that leads to a further part of the game. Level selector lists all the levels that you can play within the game, in this case I have made it 5 to start with, with the hope to grow this to 10. When a level is selected that is unlocked you shouldn't be able to play it.
- Clicking on level viewer will display all the levels, clicking on them will display a further in depth image of the level, it will also show statistics for the level such as high scores and whether it is unlocked.
- The character viewer is a way of viewing all the characters available within the game and selecting which the player wishes to use.
- Character editor is a way of editing the character such as colours for instance, in the final game this is likely to be combined with the character viewer selection.
- Instructions will include controls and what the aim of the game is.
- There is also an exit button at the bottom right which will close the game.

### LEVEL SELECTOR

#### USABILITY FEATURES

- When this button is clicked on the main menu it will display a screen consisting of 5 squares, these will be labelled as level 1 and so on to level 5.
- It will also be displayed which levels are playable and which aren't which will be obvious because of the lock symbol in the image of the level, the player is unable to do these levels until they have enough score to unlock the levels. Player also knows which level is currently selected because there will be a red outline around the level.
- They can also go back to the main menu by clicking the back button in the corner.
- When the player clicks on a level they can play, a red button will appear beside the level they can click on this and it will lead to the level.
- If a player clicks on one of the levels that is currently locked it should not display a play button as the player shouldn't be able to play it.

## LEVEL VIEWER

### USABILITY FEATURES

- When this button is clicked on the main menu it will display a screen consisting of 5 squares, these will be labelled as level 1 and so on to level 5.
- When the player clicks on one of these boxes it will go to a different screen, this screen will have a bigger full image of the level that has been clicked on.
- Beside this will be statistics of the current level, it will say whether it is locked with what is required if it is locked, it will also display the highest scores for the level.
- You will not be able to play a level through this you must go to the level selector.

## CHARACTER VIEWER

### USABILITY FEATURES

- When this button is clicked on the main menu it will display a screen consisting of a central character who will be big and the one currently selected, there will then be arrows either side so they can change between the available characters.
- My game will hopefully have a variety of characters which they can play with, these will also be editable in terms of colours however this will be quite complicated to code.
- The screen will have a central character big on the screen, this is the currently selected character either side of this character will be a left and right button so that the player can switch between the characters.
- To play with this character they just have it selected then click the back button in the bottom right and that character will be the character currently used, the player then just has to select the level they want and play and they will be able to use the character.
- To change the character they go in the character viewer scroll to the character they want click the back button and play a level and the character will change.

## CHARACTER EDITOR

### USABILITY FEATURES

- When this button is clicked on the main menu it will display a screen showing a broken apart image of the currently selected character.
- You will be able to change accessories, hair, skin colour as well as other things to personalize the character, to save the changes the player just has to click the 'make changes' button and then the back button to go back to the menu.
- You will be editing the character that you currently have selected.
- This will ideally be built in to the character viewer selection further through development of the game.

## INSTRUCTIONS

### USABILITY FEATURES

- When this button is clicked on the main menu it will display a screen with a large section of text.
- This will be an in depth guide on how to play the game such as;
- Controls
- How the lives system works
- How the score system works
- There will be a back button in the bottom left which will take them back to the main menu.

## DOES SOLUTION FOR MENU MEET REQUIREMENTS?

- As I had planned when the game first loads the loading screen is loaded and appears, user can then press spacebar which results in it going to the main menu.
- I have also coded 5 options for the main menu, this includes Level selector, View Levels, Character selector, Character editor, instructions however these are not buttons as originally planned. You must manually press for example the key L to go the level selector, I have not made them buttons so you cannot physically click the buttons but use keys to navigate instead.
- Using the level selector displays 5 options for the levels as originally intended, only 1 of the options works (level 1) however this is not as intended player presses 1 to go to it rather than physically clicking on the level 1 icon. There is also only 1 icon which is for level 1 the rest are blank and cannot be interacted with, because of this there is no need to have locked levels as intended.
- The level viewer or as name on my game view levels is a screen which doesn't have anything on it as I haven't done any of its functions or interface yet so I have achieved any of what was intended. There is also no scoring system let alone high score system which was expected to be included on this page to show players what the scores they are that they need to beat.
- The character editor and viewer screen is blank, it doesn't even display the default character. There is no sort of customization as I never got round to being able to change things such as the characters colours.
- The instruction screen is completely blank, I was going to do this once the game was actually complete and thus functions would need to be explained but I never got round to this.
- There is an exit button at the bottom of the screen but this cannot be pressed however pressing ESC on the menu screen closes the game.

## HAVE ANY FUNCTIONS OR FEATURES OF THE MENU BEEN CHANGED FROM DESIGN?

- As explained I have not been able to get buttons into my game which means I have had to add a different way of navigating between pages, despite this being efficient it is not what was expected during design. I chose to use the keys such as E, L, I for example to switch between them, this is labelled in my game. Generally pressing backspace will take it to the screen before, game can be closed by going to the menu and pressing the ESC key, also game can be closed by being on level 1 and pressing E.

## WHAT HASN'T BEEN ACHIEVED AND HOW MIGHT IT BE?

- One of my biggest changes was through how the navigation was different to what I had wanted to have during the designing of my game, I originally wanted it to be controlled through buttons controlled by the mouse and arrow keys however this has not been possible. This might be done by using a collision system similar to that found in my level collisions, this would work by comparing the coordinates of the mouse cursor with the coordinates of the buttons (images) that I want to use. An example of a button would be the button for level selector on the main menu, it could compare the coordinates of this button with the coordinates of the mouse cursor and if the mouse is being clicked; if it is then it goes to the level selector screen.
- The only page off my menu screen that works is the level selector which is used to start the levels, this is the only page which has added functionality;
- Ideally the level viewer could be used to see what the level looks like, and also display information on the level such as high scores for the level. Because I haven't got scoring working it doesn't seem worth it developing the screen until this is done. This level viewer could be done relatively easy however by just having a picture of the level and then having a system where scores are saved and then the code could just compare the order of the scores to which is highest and then display that as well as less scores as well.
- The character and viewer could more than likely be rolled into one screen as they are very similar and it seems pointless having them both. I wanted an area where the player could change the character around customize them and then use them in the game, this might be changing colours etc. This would be relatively to do, instead of having 1 sprite I could have numerous sprites on a sprite sheet drawn in different colours. On the main screen it would have the currently selected character and then the character could switch using some sort of arrow keys to change which character is actually being used, this selected character should then be the in game character.

- Now that the game is relatively near the end of development it would be easy to do an instruction screen. It could just be a simple text screen outlining the buttons to use and how to actually use the game.
- The exit button would have been easy to do and would have worked like my other buttons. Having a comparison between the cursor coordinates and the coordinates of the exit button and then also if the mouse is clicked. I also wanted back buttons on most the screens so it was possible to navigate between the pages, this would work exactly like the exit button.

## LEVELS

---

### USABILITY FEATURES

- Level needs to be loaded, level graphics need to be loaded into array as well as characters and enemies.
- Characters need to be loaded and set in the right position.
- Count down from 3 before starting the level.
- Coins need to spawn.
- Coins score needs to be set to 0.
- Timer needs to be set to 400.
- When count down finishes and level starts timer counts down from 400.
- Coin score is incremented by 1 when a coin is collected.
- When player reaches finishing area level completing screen appears with the option to go back to main menu. Will also display players score for that level.

### DOES SOLUTION MEET REQUIREMENTS

- The level as designed is drawn to the screen and fits the screen, the level consists of different platforms and obstacles which will get harder as the levels progress. However I have only been able to implement 1 level into my game, this game is also lacking a wide variety of functions that I expected to have in my game. Ideally during design I wanted around 10 levels, however realistically I have only been able to do 1 level, because I don't have a system for resetting levels other levels haven't been possible however once this is I can do multiple levels.
- As expected for level 1 the characters (player and enemy) are loaded into position correctly, there is only 1 enemy at the moment but this could be easily added to.
- There is no countdown when the game starts, it just goes straight into the level, and this doesn't matter at the moment because there is no time anyway.
- There are no coins score or timer.
- I have added a system where when the player reaches the finishing line a completed level screen appears, it also stops the characters from moving. Pressing enter allows the player to go to the level selector screen, however no level data is saved and the level also doesn't reset which means that when the player tries to start the next level everything is as it was when the level was completed.

### HAVE ANY FUNCTIONS OR FEATURES BEEN CHANGED FROM THE DESIGN?

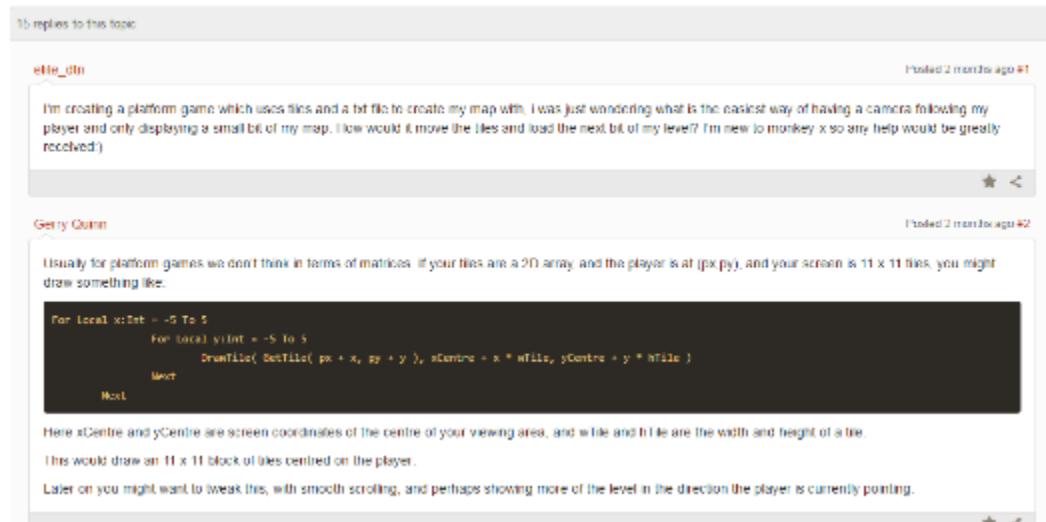
- Originally I wanted my game to be a side scrolling platform game, this is representative of me basing my game upon New Super Mario Bros. This would work so that only a section of the level is displayed at the start and then as the play moves to the side more and more of the level appears giving the impression that the camera is following the player and it also allows the level to be bigger than the size of the screen. However this became too complex and I was unable to do it in the specified time frame, this meant I had to change my level so that I could still have a big level but fit it onto the screen. I made it so that the player starts from the bottom of a square level and then using the platforms they would have to make it to the top of the level where the finishing area is.

## WHAT HASN'T BEEN ACHIEVED AND HOW MIGHT IT BE?

- Currently my game only has 1 level, this is considerably less than the 10 I had aimed to achieve. I was unable to program a method which could reset the levels so that they could be restarted over and over again, because of this it made having new levels difficult because I couldn't clear the other level from the screen and draw the new level along with the players and enemies. This could be done with some sort of clear screen method and reset the objects from the level so that the level can be done again or objects such as the player can be used in other levels. Once this method is done then it should be relatively easy to have other levels, I had also planned to have different themed levels this again is very easy and could just be done with a different tile sheet with the different tiles.
- I was unable to add more enemies into my game, I only have 1 which appears on level 1 this can kill the enemy however so I have player enemy collisions working. Ideally level 1 would have 3 enemies, this could be possible by just creating another instance of the enemy class and drawing at different position, collisions for this would still be the same. Having this means that as the level progress I can add more enemies with different powers to make the level harder and more fun to play.
- I have been unable to get side scrolling working for my game, ideally following on from New Super Mario Bros the game would follow the player as they move to the side with more and more of the level appearing. I did try to implement this feature and address the issue on a Monkey X online forum and actually was given a method of doing this as shown below.

**IMPLEMENTING A MATRIX INTO MY GAME**

MONKEY PROGRAMMING FORUMS / MONKEY BEGINNERS / IMPLEMENTING A MATRIX INTO MY GAME



- This would be of sorting my problem, it would be different to this but it is a method of how it could be done to add side scrolling to my game.

**SOUND**

## USABILITY FEATURES

- Robbie doesn't want any backing sound so there won't be any.
- However he requested that there be a sound for jumping, this will work so that when the spacebar key is pressed the jumping sound will play.
- He has also requested there be a sound for shooting a weapon, there could possibly be a different sound for character bullet and enemy bullet to make it distinguishable. Having a trigger for this sound will be harder because the power up button won't necessarily just be when the character is able to shoot so it can't make a sound when the power up button is pressed. However there could be some kind of if statement which checks if the power-up button is pressed as well as having the shooter power-up. Alternatively there could be a distinct shoot button which they can use to shoot when they have the shooter power up.

**DOES SOLUTION MEET REQUIREMENTS?**

- There is no sound at all in my game, this means that I haven't got sound working at all for my game so there is no sound for shooting and jumping.

**WHAT HASN'T BEEN ACHIEVED AND HOW MIGHT IT BE?**

- Sound is actually really easy to have in Monkey X, it works very similar to how the images work creating a variable and loading the sound file and then calling it when it is required. Because it was an original requirement to have sound for jumping I would just have the code so that when spacebar is pressed the sound for jumping plays and then stops.

**DIFFICULTY**

---

**USABILITY FEATURES**

- The first level will be the easiest level to complete, it will be important to have variable difficulties for my level, first level will be easiest and last level will be the hardest. They can be made harder by making jumps harder having more obstacles and having more enemies shooting at the player.
- Level 1 will have minimal jumps to avoid water and fire obstacles, the platforms over the obstacles will be easier to use and jump between so that there is less chance of the character falling in and losing a life.
- Throughout the levels there will be gradually more and more obstacles consisting of water and fire, if the character falls in they will have to restart the level and they will also lose a life. The enemy will also shoot their weapon more and more often, in the first level they might shoot every 5 seconds in higher levels this will reduce to 1 second – this means the character will constantly have to avoid these weapons.
- The enemy will also become more agile where they are placed so they move faster towards the character, the easiest way for the character to get past the enemies at this stage will be via using power-ups.
- It is important in the earlier levels for the character to do the level without losing lives and gaining the two when they pass the level so that they have loads of lives in the latter stages of the game.

**DOES SOLUTION MEET REQUIREMENTS**

- This difficulty area has not really been addressed because I only have 1 level and therefore the ways of making the others harder is not necessary, this means that the methods for improving difficulty doesn't meet the requirements. The first level as expected is relatively easy to complete, at the moment it only has 1 enemy and a few lava obstacles, however lava obstacles haven't been added yet so the player can't die from the lava.
- Level 1 has lots of floating platforms over the obstacles and the enemy speed is reduced so that the level is very easy to complete, it is difficult for the player to die – especially as lava collisions haven't been added to the game yet.

**WHAT HASN'T BEEN ACHIEVED AND HOW MIGHT IT BE?**

- Because I have only got 1 level I haven't really had to worry about making the levels more difficult, the methods that I had planned to use were not necessary.
- But after adding more levels I can simply make the levels more difficult by adding more obstacles such as fire and enemies and shooting enemies to try and kill the player, also reducing the number of platforms that could be used to avoid the obstacles. This is very easy to do and can be done by just changing the tile map around changing which tiles are drawn in the text file.

## COINS

---

### USABILITY FEATURES

- The coins will need to spawn at the start of the level, this will be on platforms and other hard places so that it isn't too easy to collect all the coins, the more coins the player collects the higher there score will be however they have to complete the level in the shortest time too.
- Coins will be collected by having an if statement which compares the coordinates of a coin with the coordinates of the character, if they match the coin will disappear and the coin score will increase by 1.

### DOES SOLUTION MEET REQUIREMENTS?

- There is currently no coin system or scoring system at all in my game, the game is just simply getting from the bottom to the top in this final prototype version.

### WHAT HASN'T BEEN ACHIEVED AND HOW MIGHT IT BE?

- I never looked into adding a coin system as it was one of the less important parts of the game, I considered getting the general structure of the game to work as more important. However I think adding coins could be done fairly easy, it would require having a coin image with coordinates and then someway of generating random positions for the coin to be drawn at, this would also need to be for as many coins as required for the level which would change depending on the level.

## TIMER

---

### USABILITY FEATURES

- When the level initializes the timer will need to be set to 400 seconds.
- It will then need to count down from 400 seconds, this gives the player 400 seconds to complete the level.
- When the player runs out of time it will need to say level / game over and go back to the main menu.
- It will need to stop once the player completes the level and then be used in calculating the players score.

### DOES SOLUTION MEET REQUIREMENTS?

- There is currently no timer on my game at all so none of the requirements have been met for the timer.

### WHAT HASN'T BEEN ACHIEVED AND HOW MIGHT IT BE?

- What contributed to me not including a timer in the game is that there is no timer function or module to fulfil this, this makes implementing a timer incredibly complex. I am therefore not sure how this could be code easily into my game, however I have researched how this might be done;





## SCORING SYSTEM

---

### USABILITY FEATURES

- To start with when the player is completing the level it will not give the score, instead it will give a breakdown of the score so it will show you the number of coins collected and how long is left to complete the level.
- The coin score and timer will need to be reset at the beginning of each level.
- Coin score goes up one every time a coin is collected.
- Timer counts down from 400 until the player completes the level or runs out of time.
- At the end once the level is completed there will be a popup saying the score.
- The score is calculated by multiplying the number of coins collected (coin score) by the remaining time (timer value), so the higher the two values the higher the score will be.
- There will be a different popup message if the person achieves a high score, saying 'congratulations on getting the highest score'.
- This score will also need to be added on level viewer where it lists scores for that level, must give the scores in the right order. It would probably be appropriate to only store the 3 highest scores and no more.

### DOES SOLUTION MEET REQUIREMENTS?

- I focused mostly on getting the main aspects of the game working such as movement and graphics and therefore did not have enough time to complete the extra parts such as a score system when the player can collect coins and have a timer. This therefore meant that I didn't get around to achieving the requirements for a scoring system.

### WHAT HASN'T BEEN ACHIEVED AND HOW MIGHT IT BE?

- There is no score bar on the game, this can be done simply with it just displaying text, I could make it more complex by designing one and then adding it as an image.
- There are no coins so this has not been done but if this was done then it would be simple to just have a score count and whenever the player collects a coin then the coin score is incremented by 1. Then at the end of the level it just gives the total of the coin score.
- Similarly there is no timer as stated earlier and it is explained how this could be done, the timer would be used as part of the score system however. The game is going to count down from 400 (seconds) If the player doesn't finish the level by then the level finishes but if they do then their timer score is taken, the higher being better. This timer should also be displayed on the score bar.
- At the end of the game there should be a score popup which displays the calculated score, the score can be calculated using by multiplying the coin count by what's left on the timer. This means the higher the score the better.
- Again, a pop up message which can be done using text should display if the score achieved was a high score, some sort of high score system could be easy to achieve with the totals being saved in a text file and then fetched when required.
- The level viewer was supposed to be very interactive highlighting scores achieved, but because there isn't a score system, this wasn't possible.

## LIVES SYSTEM

---

### USABILITY FEATURES

- My lives system is very similar to that of the original NSMB and is something that I have reused.
- When the player starts the game they will be given 5 lives.
- The level will end if player falls from base platform and they will lose 1 life.
- The level will end if player is shot by one of the enemy characters and will lose 1 life.
- When the player completes a level will receive 2 lives to their overall live count.
- In the event that a player loses all their lives the game will be completely over and they will lose all the progress they have made and start with 5 lives on level 1 and all the other levels locked.
- If the character has the invincible power up and are shot they will not die, and if they fall in an obstacle such as fire and water they will also not die and will be able to jump out.
- If the character has the giant power up and are shot they will return to their original size, if they fall into an obstacle they will not die and will stay giant.
- This means that it might be a good idea for the character with minimal lives to do easier levels again in order to get lives to allow them more chance at the level.
- However with further development of my game it might be possible to have a section where you can purchase more lives with the game currency.

### DOES SOLUTION MEET REQUIREMENTS?

- No lives system at all as per requirements, thus no 5 lives when the game starts or gaining or losing of lives.
- I have working collisions between player and enemy, if player collides with the enemy the level finishes. This would usually result in a lost life but this hasn't been done.

### HAVE ANY FUNCTIONS OR FEATURES BEEN CHANGED FROM THE DESIGN

- In my games current state there is no lives, however the player can still die and would be able to restart the level they just don't have lives so they can try and complete it as many times as they want.

### WHAT HASN'T BEEN ACHIEVED AND HOW MIGHT IT?

- Having a lives system which could just be a simple counter, every time a player dies then 1 is taken away from the counter and every time a level is completed 2 is added to the timer. This counter can then just be displayed on the level and menu to show the player how many lives they have.

## CONTROLS

---

### USABILITY FEATURES

- Left arrow button used for moving the character to the left this will be done in terms of coordinates when I code it, for instance clicking left should move the character x coordinates to the left or change the x coordinate by a certain amount.
- Right arrow button used for moving the character to the right this will be done in terms of coordinates when I code it, for instance clicking right should move the character x coordinates to the right or change the x coordinate by a certain amount.
- The spacebar button will be used for jumping this will be done in terms of coordinates when I code it, for instance clicking spacebar should move the character up x coordinates or change the y coordinate by a certain amount.
- The enter button will be used for using the powerups this will require code to distinguish between the powerups because they do different things. Simply when the enter button is pressed the power up that was collected will be used.
- The p button will be used for pausing the level, a pause screen will appear, pressing p again will un pause it with a 3 second countdown.

## DOES SOLUTION MEET REQUIREMENTS?

- Pressing the left arrow moves the player to the left.
- Pressing the right arrow moves the player to the right.
- Pressing spacebar makes the player jump and fall.
- There are no power-ups so no need for a power-ups button.
- There is no pause function for my game but this doesn't matter as there is no timer anyway.

## WHAT HASN'T BEEN ACHIEVED AND HOW MIGHT IT BE?

- Having the actual key inputs is easy to implement but I haven't had to do this because things such as power-ups haven't been done, there is therefore no need for a power-ups button. This would work the same way as all the other buttons do.
- Pausing the game might be slightly more complex, however it is done by simply stopping all of the moving objects. For example, freezing the players position and enemy's positions and any moving objects, then also stopping the timer. Un pausing could just be done by reversing these actions.

## COLLISIONS

---

## USABILITY FEATURES

- There will need to be a collision for when the enemy shoots a bullet at the character, this will be relatively simple to implement. For instance, when bullet coordinates match the coordinates of the player then the player will die.
- There will also need to be a collision detection for when the character falls into one of the obstacles, such as when the character falls into the water or fire.

## DOES SOLUTION MEET REQUIREMENTS?

- I have no shooting bullet enemy's so the collision for this has not been necessary or achieved.
- Similarly, I didn't get around to adding collisions for if the player falls into lava, however because I have tile collision working then this would be very easy.

## HAVE ANY FUNCTIONS OR FEATURES BEEN CHANGED FROM THE DESIGN

- The enemy does not kill the player through shooting bullets but is programmed to just kill the player if the two collide, so if the player runs into the moving enemies then the level finishes as the player dies.

## WHAT HASN'T BEEN ACHIEVED AND HOW MIGHT IT?

- The enemy shooting a bullet requirement is missing but this could be done simply with a bullet that has increasing or decreasing x coordinates to move across the screen. Collisions would just work by comparing the bullet coordinates with the coordinates of the player, if the two match then they are colliding and so the level should end.
- As said earlier the collisions for the obstacle such as lava could be added easily, it would require a few more lines. I already have tile collisions working so it would be the same as this but instead of player not being able to go through the tile for instance the player would fall in the lava and thus the level would finish as the player dies.

## POWER-UP'S

---

### USABILITY FEATURES

- There will be a power up for becoming giant, this will entail an If statement for if the giant power up actually has been collected.
- When the power up has been collected and the enter button has been pressed to trigger the use of the power up the player size will change to 3x what it was.
- It will last for 20 seconds and it gives the player the ability to charge through platforms destroying them in the process.
- During this time the character will also be immune to enemy weapons.
- They will also be immune from dying to static obstacles such as the water and fire traps, in this event the player will just be able to jump out of the obstacle.
- When power up finishes player will return to original size.
- There will be a power up for becoming invincible, this will entail an if statement for if the invincibility power up has actually been collected.
- When this power up has been collected and the enter button has been pressed to trigger the use of the power up a flash text screen saying 'You are Invincible' on will appear.
- The character will also become a flashing color.
- This power up will last for 20 seconds and gives the character complete invincibility to every way of dying, the person should try get as far as they can in the 20 seconds to advance quickly through the level.
- When the power up finishes, player will return to normal.
- The final power up will enable the character to be able to shoot bullets, this will entail an if statement from if the shooter power-up actually has been collected.
- When the power up has been collected and the enter button has been pressed the character will be able to shoot every time they click the enter button.
- This will work the same way as the enemies will however shot rate may be increased to provide more of an advantage.
- Shooting a bullet / missile at the enemy will result in the enemy dying, however this power up doesn't make the character immune from being killed in any form.

### DOES SOLUTION MEET REQUIREMENTS?

- There is no power up system so it doesn't meet the requirements, there is no special features at all for the player as I have had to concentrate on the player movement and level parts.

### WHAT HASN'T BEEN ACHIEVED AND HOW MIGHT IT?

- A power-up system is required as per the requirements; this would involve 3 mains concepts – becoming giant, becoming invisible and being able to shoot bullets
- Becoming giant would be simple; when the player uses the power-up the size of the sprite is changed which can be done very easily as the size is declared when the sprite is drawn so you can just code it to draw it a different size.
- Becoming invisible would just be a case of disabling collisions between enemies and the obstacles and then the player can't be killed by them.
- Shooting a bullet could just be replicated with the enemy code and then do a collision function for if the bullet collides with the enemy then the enemy should die.

## PAUSING

---

### USABILITY FEATURES

- Robbie has requested that he should be able to pause the game whilst playing the level, all he will have to do is click the p button and a pause screen will appear and everything will stop.
- Unpausing will result in a 3 second countdown giving the player time to get ready.

### DOES SOLUTION MEET REQUIREMENTS?

- There is no pausing available with the game, there is no timer so this isn't really necessary although I guess the enemy can still move.

### WHAT HASN'T BEEN ACHIEVED AND HOW MIGHT IT BE?

- Pausing the game could be done by simply stopping all of the moving objects. For example, freezing the players position and enemy's positions and any moving objects, then also stopping the timer. Unpausing could just be done by reversing these actions.
- The countdown could then just be a smaller timer that counts down from 3 and draws the text to screen.

## GAME ENDING

---

### USABILITY FEATURES

- The game will not end if the player dies in the level however it will start the level again and they will lose a life.
- If the player loses all their lives, then it will go back to main menu and everything will reset as if they have just started the game.
- The player can close the game by clicking the exit button on the main menu, it will take it back to the computers desktop.

### DOES SOLUTION MEET REQUIREMENTS?

- There is no lives system so there is nothing stopping the player from starting the level again. However, when the level restarts there is no reset thing so it thinks when the level starts that the level is instantly finished or failed.
- There is no game reset if the player loses all their lives.

### HAVE ANY FUNCTIONS OR FEATURES BEEN CHANGED FROM THE DESIGN

- There is an exit button on the main menu however it doesn't work like a button so the player can't press it, however the player can press ESC on the main menu and the game will close. Similarly, the player can click E whilst on the level and game will close as well.

### WHAT HASN'T BEEN ACHIEVED AND HOW MIGHT IT BE?

- There should be some way of resetting the level when the player dies and when the player completes the level. This could be done by resetting the positions back to their starting positions of all the objects in the level but this isn't easy and will become more complex when there are more levels.

## MOVEMENT

---

### USABILITY FEATURES

- If the left button on the arrows is clicked, then the character will move to (current coordinate - 32) along to the left
- If the right button on the arrows is clicked, then the character will move to (current coordinate + 32) along to the right.
- If the spacebar button is clicked, then the character will move to (current coordinate + 32) above.

### DOES SOLUTION MEET THE REQUIREMENTS?

- Pressing the left arrow moves the player to the left.
- Pressing the right arrow moves the player to the right.
- Pressing spacebar makes the player jump and fall.
- It must be noted however, the movement is still not completely perfect or as the client would like, however this might be down to the collisions not working 100% as well.

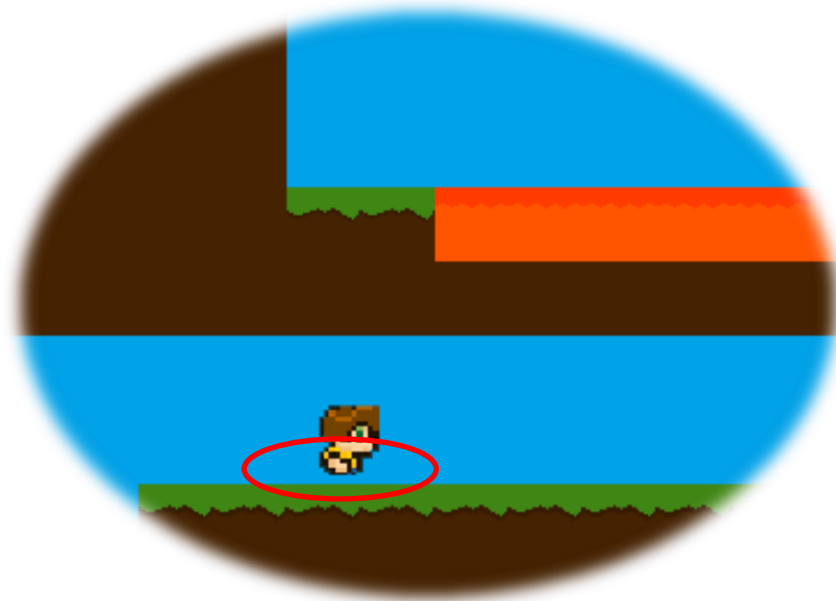
---

## MAINTAINENCE

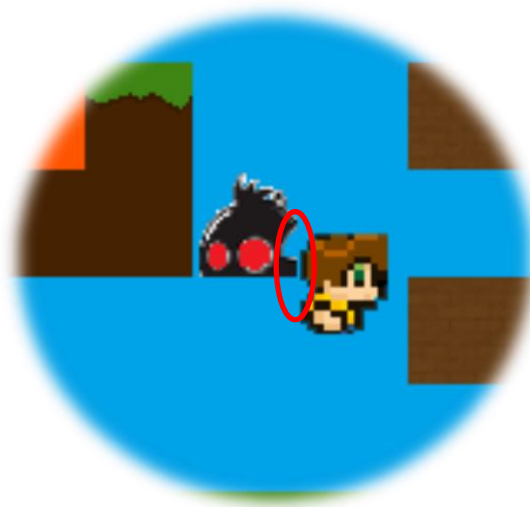
### BUGS THAT CURRENTLY EXIST WITH MY GAME THAT WOULD HAVE TO CHANGED BEFORE RELEASE

My game currently has a number of bugs and problems that would need to be solved before the game can be released, this makes the game unstable and it doesn't function as expected without this.

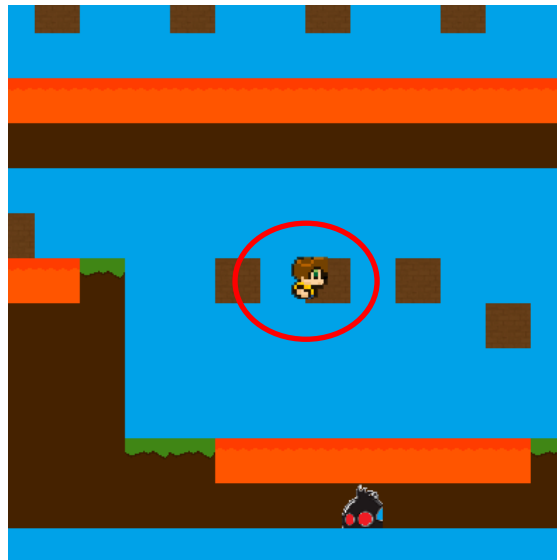
- The player doesn't fit properly on the platform and floats just above the platforms by a few pixels which looks glitch.



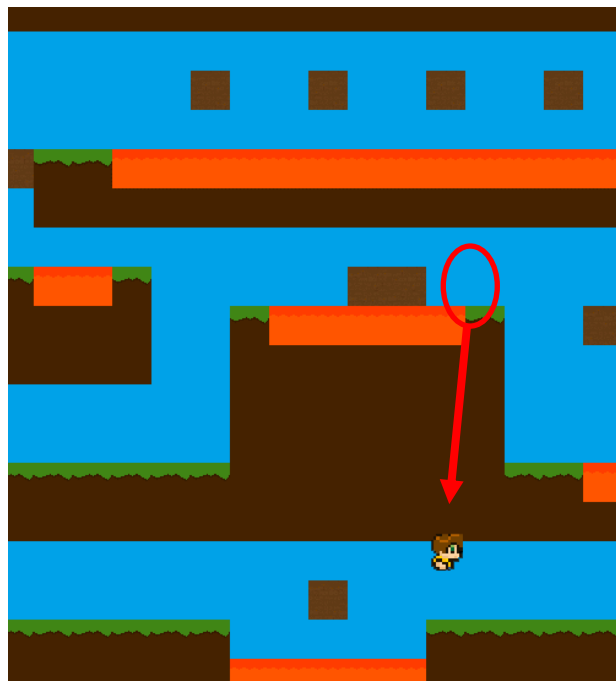
- Collisions are not pixel perfect, in my game they are currently box to box collisions however this is not clean and due to the shape of my player and enemy this doesn't work as they can be colliding when they are physically not. This means pixel perfect collisions need to be implemented to make the game less glitchy in this area.



- Collisions are glitchy and the collision between tiles and player are not working that great. The collisions where the player hits the tiles from the side work properly however top and bottom collisions do not work. Player glitches when they hit a floating platform and can move inside it.



- When the player hits the lava they fall through the floor somehow for unexplained reasons.





## UPDATES TO GAME

After my game has been released in order to maintain it then I will need to add updates to the game which would be the requirements that were not achieved in the original development of the game.

- One of my main parts to the game that I wanted to have was power-ups, this can be implemented into my game as part of an update to the game. This would provide the game with a lot more competitiveness, it would make the game more enjoyable. Power-ups consists of 3 mains areas; becoming giant, becoming invincible and finally the ability to shoot bullets.
- Also simply I would need more levels so that there is more game progression for the player, this could be done easily by creating a new instance of the level class and then just changing the text file to change how the level looks. As per the requirements the levels need to get harder so this can be done by just reducing the number of platforms and increasing the number and frequency of obstacles and enemies. Originally the client also wanted themed levels such as the desert and snow which resembles that of New Super Mario Bros, this could be done by having a different tile map with the different themes.
- There was also supposed to be a scoring system with a coin count and timer displayed at the top of the screen.
- Coins would need to be drawn to the screen and displayed as a coin count, this will just consist of a simple coin collection method which when a coin is collected the coin score is incremented by 1 and displayed at the top of the screen.
- The timer would work by having a standard timer which counts down from 400, it does also however need to pause when the pause button is pressed which is another function that should be added.
- The score bar at the top should display the current coin count and should therefore start on 0 and should increase by 1 every time a coin is collected. The timer will also then be displayed showing the player how long they have left at the end of the game.
- The final game scoring and scoring for the levels works so that the coin count and timer are multiplied together, this means the higher the score the better.
- The score information needs to be held for the levels so that on the level viewer the player can review the highest scores for each level and therefore know what they require to beat the original high score.
- The client requested there be a sound for the functions jumping and shooting a bullet but said there didn't need to be game sound. However this could be easily done and there could be an options section on menu where game sound could be muted.
- The menus also need to be worked upon because despite them working they are still not up to the level that was intended before the game was developed, at the moment the menus are mostly controlled using the keys. However the client suggested it would be better if it were controlled using the mouse cursor and having the different options as buttons which the user could click on.
- Finally it might be possible to have some sort of character editor which was missing from the requirements of the game, this would allow the player to edit the character they are using and change the colours of the characters clothing for example. This would then be the in game character that is being controlled so the controlled character can be changed.

# Appendix A

# Code Listing

```

MazeRunner.monkey
Import Mojo
Import brl
Global Game:Game_app

Function Main()
 Game = New Game_app
End

Class Game_app Extends App

 Global GameState:String = "INITIALISE"

 'Variables

 'Declares level 1 as an instance of the level class
 Field level1:Level

 'Game menus image
 Field loadingscreen:Image
 Field levelselector:Image
 Field mainmenu:Image
 Field finishline:Image

 'player image variable
 Field p1_img:Image

 'enemy image variable
 Field e1_img:Image

 'image variables for game finish banner
 Global ban1_img:Image
 Global ban2_img:Image
 Field levelfailedbanner:levelfailedbanner

 'Declares player1 as a new instance of player class
 Field player1:player
 'Declares enemy1 as a new instance of enemy class.
 Field enemy1:enemy

 'collision function for tile collisions
 Global collision:Bool = False

 'Game initialisation code
 Method OnCreate()
 SetUpdateRate 60

 'Loads images for the game
 loadingscreen = LoadImage ("loadingscreen.png")
 levelselector = LoadImage ("levelselector.png")
 mainmenu = LoadImage ("mainmenu.png")
 ban1_img = LoadImage ("levelfailed1.png")
 ban2_img = LoadImage ("levelfailed2.png")
 finishline = LoadImage("finishline.png")

 'Creates level1 as a new instance of Level class
 level1 = New Level
 level1.load()

 'Loads images for the two characters, player and enemy
 p1_img = LoadImage("sprite.png")
 e1_img = LoadImage("enemy1.png")

 'Creates player1 as a new instance of Player
 player1 = New player

 'Creates enemy1 as a new instance of Enemy
 enemy1 = New enemy

 'Grabs image from the sprite sheet
 player1.p1Sprite = p1_img.GrabImage(0, 0, 32, 32)
 enemy1.enemy1Sprite = e1_img.GrabImage(0, 0, 32, 32)

 End

```

```

'Variables that change frequently
Method OnUpdate()

 Select GameState

 Case "INITIALISE"

 'Sets the player position
 player1.px = 10
 player1.py = 865
 player1.pspeed = 2.0
 player1.pxspeed = player1.pspeed
 player1.pyspeed = 0

 'Sets the enemy position
 enemy1.exspeed = 1.5
 enemy1.ex = 700
 enemy1.ey = 800

 GameState = "LOADING_SCREEN"

 Case "LOADING_SCREEN"
 'Button inputs for using loading screen
 If KeyHit (KEY_SPACE) Then GameState = "MENU"

 Case "MENU"
 'Button inputs for using menu screen
 If KeyDown (KEY_L) Then GameState = "LEVEL_SELECTOR"
 If KeyDown (KEY_V) Then GameState = "LEVEL_VIEWER"
 If KeyDown (KEY_C) Then GameState = "CHARACTER_VIEWER"
 If KeyDown (KEY_E) Then GameState = "CHARACTER_EDITOR"
 If KeyDown (KEY_I) Then GameState = "INSTRUCTIONS"
 'Button to close the game
 If KeyDown (KEY_ESCAPE) Then Error("")

 Case "LEVEL_SELECTOR"
 'Button inputs for using level selector screen
 If KeyHit (KEY_BACKSPACE) Then GameState = "MENU"
 If KeyHit (KEY_1) Then GameState = "LEVEL_1"

 Case "LEVEL_VIEWER"
 'Button inputs for using level viewer screen
 If KeyHit (KEY_BACKSPACE) Then GameState = "MENU"

 Case "CHARACTER_VIEWER"
 'Button inputs for using character viewer screen
 If KeyHit (KEY_BACKSPACE) Then GameState = "MENU"

 Case "CHARACTER_EDITOR"
 'Button inputs for using character editor screen
 If KeyHit (KEY_BACKSPACE) Then GameState = "MENU"

 Case "INSTRUCTIONS"
 'Button inputs for using instructions screen
 If KeyHit (KEY_BACKSPACE) Then GameState = "MENU"

 Case "LEVEL_1"
 'Button controls on level1
 If KeyHit (KEY_ESCAPE) Then GameState = "MENU"
 If KeyHit (KEY_E) Then Error("")

 player1.update

 'Controls for the player
 If KeyDown (KEY_RIGHT) Then
 player1.px +=(player1.pspeed)
 End

 If KeyDown (KEY_LEFT) Then
 player1.px -=(player1.pspeed)
 End

 'Collisions with floating platform
 If level1.tile_collision(player1.px, player1.py) = level1.FPlatform Then
 collision = True
 Else collision = False
 End

 'Collisions with base platform
 If level1.tile_collision(player1.px, player1.py) = level1.BPlatform Then
 player1.py -= player.gravity
 End

 'Responses to collisions
 If collision = True Then
 player1.px -= player1.pspeed
 End

```

```

 If collision = True Then
 player1.py -= player.gravity
 player.playerjump = False And player1.py + 0
 End

 'Detects if the level is finished
 If player.levelfinish = True And KeyHit (KEY_ENTER) Then GameState = "LEVEL_SELECTOR"

 'If enemy is colliding with player sets the player to being dead
 If enemycollision(player1.px, player1.py, 32, 32, enemy1.ex, enemy1.ey, 40, 45) Then player.playerDead = True

 'Response to the player being dead
 If player.playerDead = True And KeyHit (KEY_BACKSPACE) Then GameState = "LEVEL_SELECTOR"
 End
End

Method OnRender()
 Select GameState
 Case "LOADING_SCREEN"
 'Draws loadingscreen
 DrawImage loadingscreen, 0, 0
 Case "MENU"
 'Draw main menu
 DrawImage mainmenu, 0, 0
 DrawText("Press L", 235, 175, 0.5)
 DrawText("Press V", 235, 280, 0.5)
 DrawText("Press C", 235, 385, 0.5)
 DrawText("Press E", 235, 490, 0.5)
 DrawText("Press I", 235, 595, 0.5)
 DrawText("Press ESC", 770, 875, 0.5)
 Case "LEVEL_SELECTOR"
 'Draw level selector
 DrawImage levelselector, 0, 0
 DrawText("Press 1 for Level 1", 220,400,0.5)
 Case "LEVEL_VIEWER"
 Cls(0,0,255)
 DrawText("Level Viewer", 450, 10, 0.5)
 Case "CHARACTER_VIEWER"
 Cls(0,0,255)
 DrawText("Character Viewer", 450, 10, 0.5)
 Case "CHARACTER_EDITOR"
 Cls(0,0,255)
 DrawText("Character Editor", 450, 10, 0.5)
 Case "INSTRUCTIONS"
 Cls(0,0,255)
 DrawText("Instructions", 450, 10, 0.5)
 Case "LEVEL_1"
 Cls(0, 0, 0)
 DrawText("Level 1", 450, 10, 0.5)
 levell.draw
 DrawImage finishline, 896, 0
 player1.render
 enemy1.initialise

 If collision = True Then
 DrawText("Collision", 10, 10)
 Else collision = False
 DrawText("No Collision", 10, 10)
 End

 If player.levelfinish = True Then
 SetColor(0,0,0)
 DrawRect(230, 230, 500, 500)
 SetColor(255, 0, 0)
 DrawText("Level 1 Complete!", 480,480,0.5)
 DrawText("Press Enter to go to level selector!", 480, 500, 0.5)
 End
 End

```

```

 If player.playerDead = True Then
 levelfailedbanner.initialise
 player1.pspeed = 0
 enemy1.exspeed = 0
 End

 End

End

End

Class Level

 Field tiles:String[51][]
 Field tileset:Image
 Const BrownTile = 0
 Const SlopeDown = 1
 Const SlopeUp = 2
 Const BPlatform = 3
 Const Sky = 4
 Const FPlatform = 5
 Const Lava = 6

 Method New()
 tileset = LoadImage ("level1tilemap.png",32,32,7)
 For Local i:Int = 0 To 30
 tiles[i] = New String[31]
 Next
 End

 Method load()
 Local map:FileStream
 Local map_data:String
 Local data_item:String[]

 map = FileStream.Open("monkey://data/level1map.txt","r")
 map_data = map.ReadString()
 map.Close

 data_item = map_data.Split("~n")
 For Local y:Int = 0 To 29
 For Local x:Int = 0 To 29
 tiles[x][y]=Int(data_item[y][x..x+1])
 Next
 Next
 End

 Method draw()
 Local tile:String
 For Local y:Int = 0 To 29
 For Local x:Int = 0 To 29
 tile = tiles[x][y]
 If tile = BrownTile Then DrawImage tileset, x*32, y*32, BrownTile
 If tile = SlopeDown Then DrawImage tileset, x*32, y*32, SlopeDown
 If tile = SlopeUp Then DrawImage tileset, x*32, y*32, SlopeUp
 If tile = BPlatform Then DrawImage tileset, x*32, y*32, BPlatform
 If tile = Sky Then DrawImage tileset, x*32, y*32, Sky
 If tile = FPlatform Then DrawImage tileset, x*32, y*32, FPlatform
 If tile = Lava Then DrawImage tileset, x*32, y*32, Lava
 Next
 Next
 End

```

```
Method tile_collision:Int (x:Int,y:Int)
 Local left_tile:Int = x / 32
 Local right_tile:Int = (x+31) / 32
 Local top_tile:Int = y / 32
 Local bottom_tile:Int = (y+31) / 32

 If left_tile < 0 Then left_tile = 0
 If right_tile > 29 Then right_tile = 29
 If top_tile < 0 Then top_tile = 0
 If bottom_tile > 29 Then bottom_tile = 29

 Local collision_result:Int = 0
 For Local i:Int = left_tile To right_tile
 For Local j:Int = top_tile To bottom_tile
 If tiles[i][j] = FPlatform Then collision_result = FPlatform
 If tiles[i][j] = BPlatform Then collision_result = BPlatform
 Next
 Next
 Return collision_result
End

End

Method render()
 DrawImage p1Sprite, px, py
End

End

Class enemy
 Field enemy1Sprite:Image
 Field ex:Int
 Field ey:Int
 Field exspeed:Int
 Field moveRight:Bool = True

 Method initialise()

 If(moveRight = True)
 ex += exspeed

 If(ex > 800) Then moveRight = False
 Else
 ex -= exspeed

 If(ex < 600) Then moveRight = True
 End If

 DrawImage enemy1Sprite, ex, ey
 End
End
```

```

Class player
 Field plSprite:Image
 Field px:Int
 Field py:Int
 Field pspeed:Int
 Field pxspeed:Int
 Field pyspeed:Int
 Global playerDead:Bool = False
 Global playerjump:Bool = False
 Global gravity:Float = 0
 Global levelfinish:Bool = False

 Method update()
 If playerjump = False And KeyDown(KEY_SPACE) = True
 gravity = -4
 playerjump = True

 End

 If playerjump = True
 gravity += 0.2

 If gravity <= 0
 For Local i:Int = 0 Until Abs(gravity)
 py -= 1
 End
 End

 If gravity > 0
 For Local i:Int = 0 Until gravity
 py += 1

 If py > 865 Then
 playerjump = False
 py = 865
 Exit
 End
 End
 End

 If py < 64 And px > 880 Then
 levelfinish = True
 Else levelfinish = False
 End

 End

 End

End

Class levelfailedbanner
 Global ban1_x:Int = 230
 Global ban2_x:Int = 230
 Global ban1_y:Int = -150
 Global ban2_y:Int = 960
 Global banspeed:Int = 5.0

 Method initialise()
 DrawImage Game_app.ban1_img, ban1_x, ban1_y
 DrawImage Game_app.ban2_img, ban2_x, ban2_y

 If player.playerDead = True Then
 ban1_y += banspeed
 ban2_y -= banspeed
 If (ban1_y = 330) Then
 banspeed = 0
 DrawText("Press backspace to continue", 480, 520, 0.5)
 Endif
 End

 End

End

Function enemycollision:Bool (x1:Int, y1:Int, w1:Int, h1:Int, x2:Int, y2:Int, w2:Int, h2:Int)
 If x1 >= (x2 + w2) Or (x1 + w1) <= x2 Then Return False
 If y1 >= (y2 + h2) Or (y1 + h1) <= y2 Then Return False
 Return True
End

```



Candidate Name: [REDACTED]

Candidate Number: [REDACTED]