***Code and screenshots***

***(Previous project and current)***

*Previous project - pg1:*

* *Version 1 - pg1*
* *Version 2 - pg3*
* *Version 3 - pg4*
* *Version 4 - pg6*
* *Version 5 - pg7*
* *Version 6 -pg8*
* *Version 7 - pg9*
* *Version 8 - pg10*
* *Version 9 - pg10*
* *Version 10 - pg11*
* *Version 11 - pg13*
* *Version 12 - pg13*
* *Sprite Masks - pg14*
* *Justifications - pg15*

*Maze project - pg16:*

* *Version 1* ***(Contains first development of 2D arrays)****- pg16*
* *Version 2* ***(Contains first development of recursion)****- pg20*
* *Version 3 - pg21*
* *Version 4 - pg24*
* *Version 5 - pg25*
* *Version 6* ***(Contains full internal commentary)*** *- pg27*
* *Justifications - pg36*
* *End-game screenshots - pg38*

***Prev, project***

* ***Bits of this project’s code are used in the ‘maze’ project. I.e: The maze project’s main menu is almost a direct copy from this project***
* ***Any screenshot on its side is on its side to make it more readable***

**V1**

Changelog:

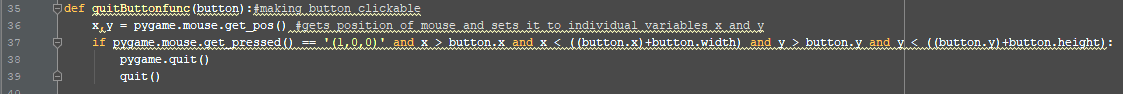
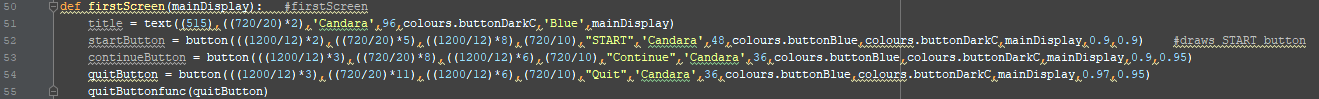
* Imports implemented
* Colours library class implemented
* mainDisplay class implemented
* mainDisplay instance created in main() function
* Button class implemented
* Text class implemented
* firstScreen function in development
* Main function created and (most likely) finished
* Instances of ‘text’ and ‘button’ created in firstScreen function

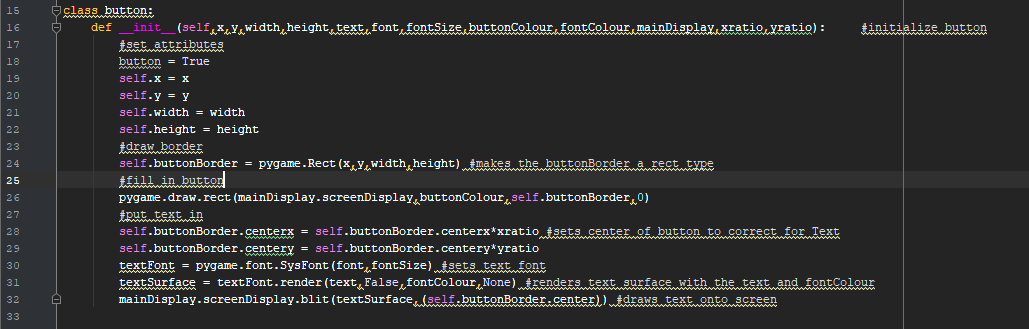
***First screenshot is on its side to make it readable***

****

**V2**

Changelog:

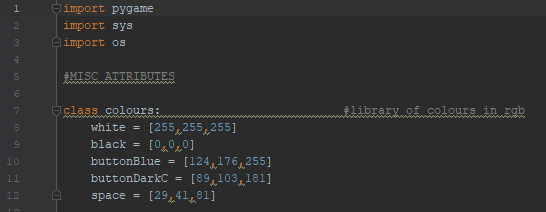
* Button class given more attributes
* quitButtonfunc in development, unsure if in end product
* firstScreen function now has three instances of buttons for the start, continue and quit buttons

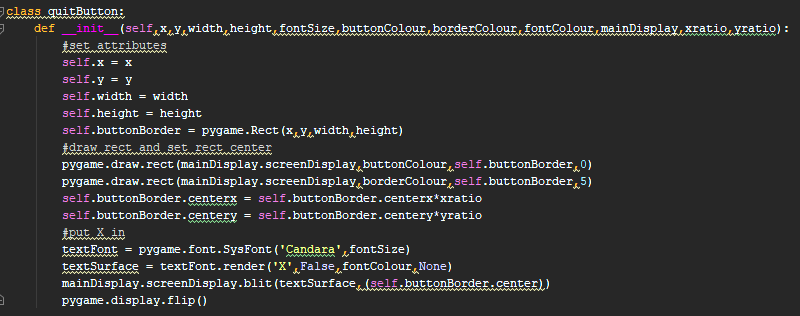


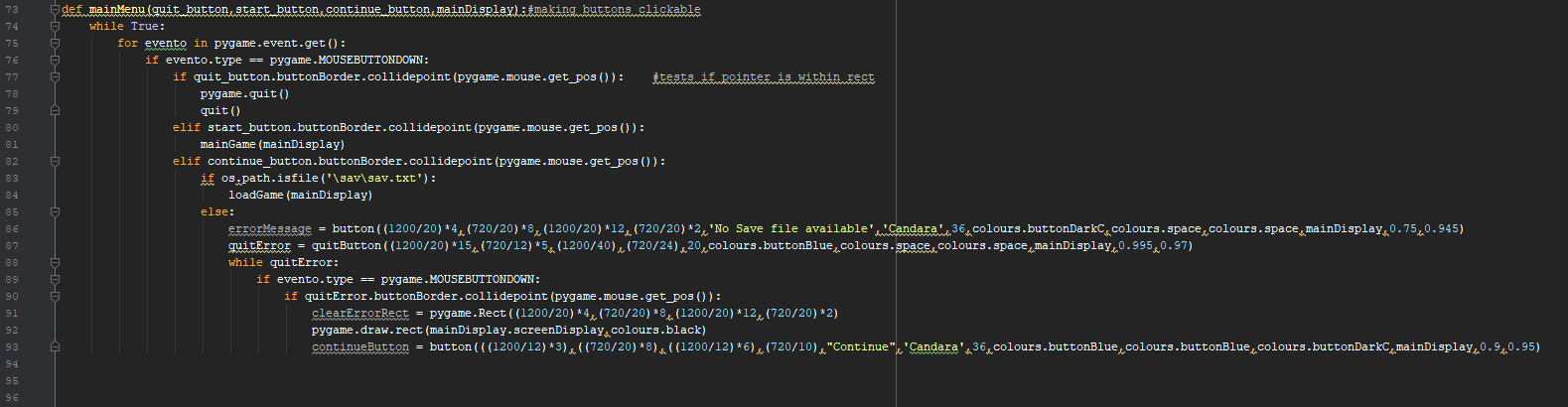
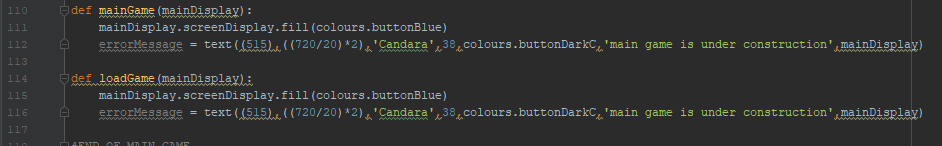
**V3**

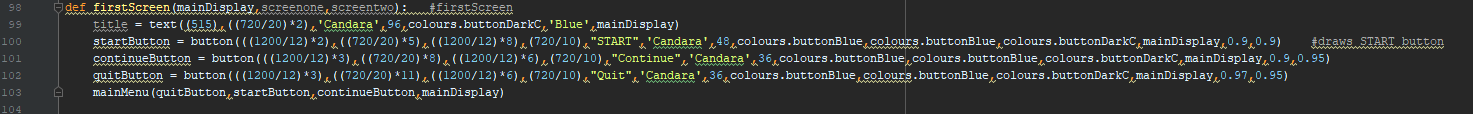
Changelog:

* Imports updated
* Colours class has one more colour attribute
* quitButton class implemented
* mainMenu function in development for button functionality, trying to implement quitButton on error message when there is no sav.txt file and loadGame is attempted
* mainMenu function called in firstScreen function
* mainGame placeholder function created
* loadGame placeholder function created





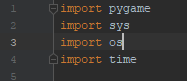


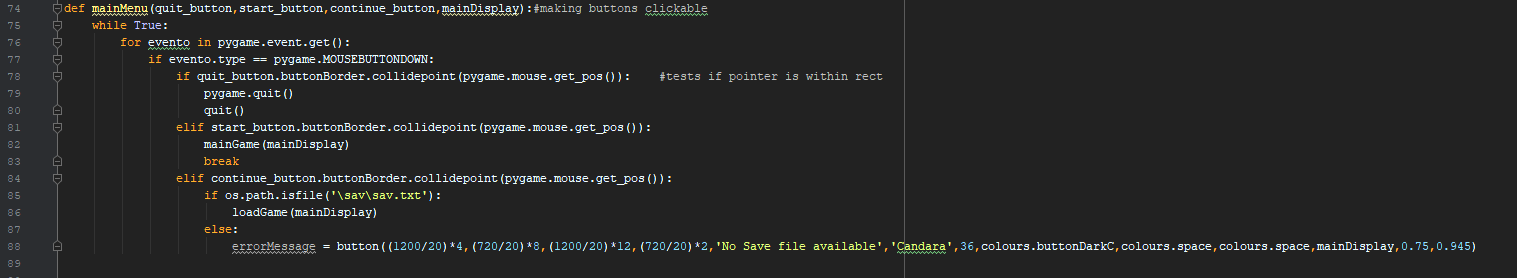


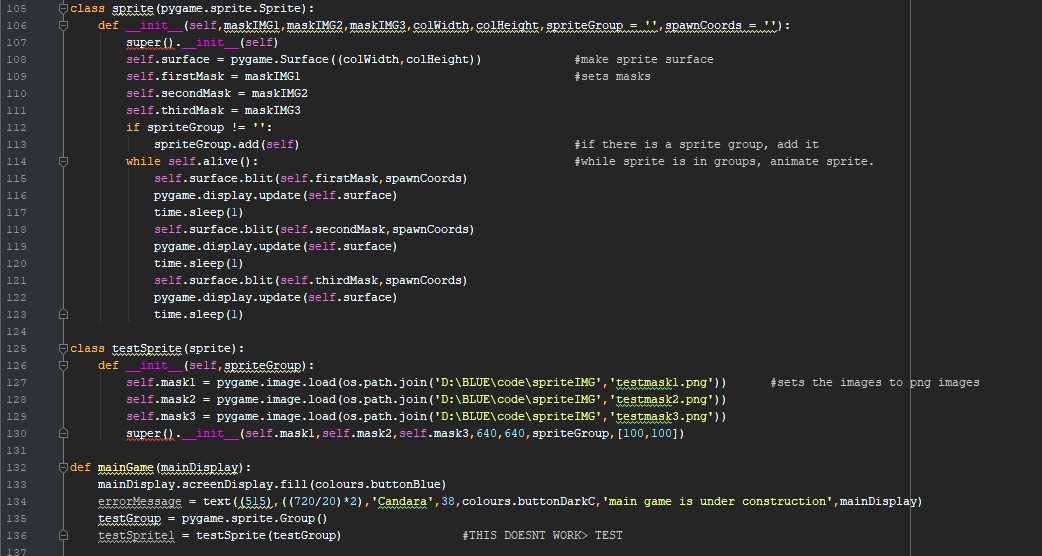
**V4**

Changelog:

* Imports updated
* quitButton class scrapped in mainMenu function
* Sprite class in development, attempting animation of sprites
* testSprite (subclass of sprite) implemented to test out whether animation works
* mainGame now has an instance of a sprite group and an instance of testSprite
* testGroup created for testSprite

****

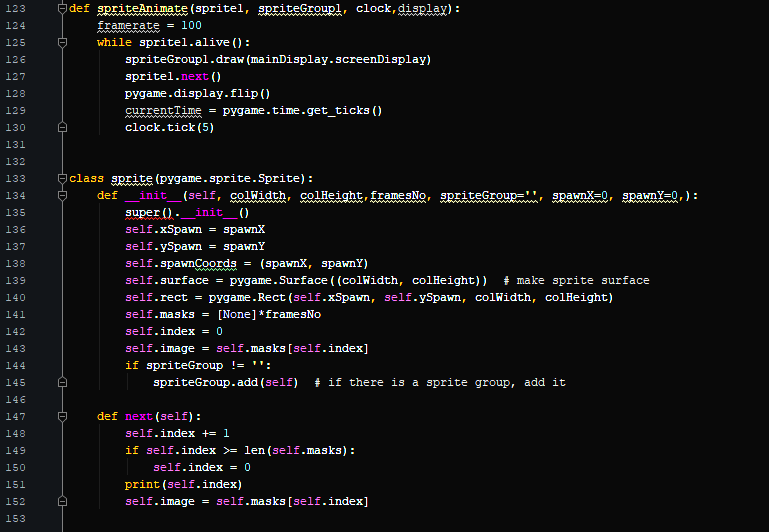
****

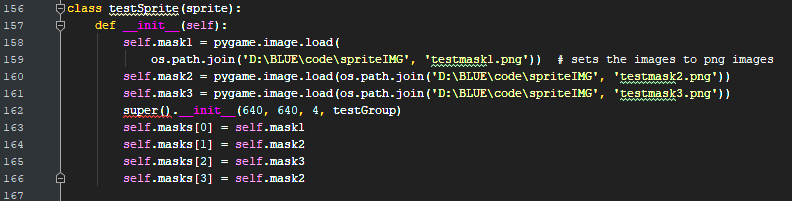
****

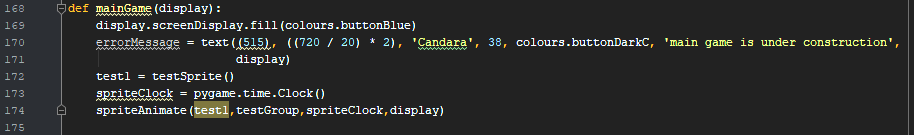
**V5**

Changelog:

* Sprite animation assigned to a separate spriteAnimation function, freshly implemented, which includes time related features
* Sprite masks assigned to array instead of individual variables, next method added to sprite for animation
* testSprite now allows for new array system for animation
* spriteAnimate called in mainGame
* testGroup made a global variable

****

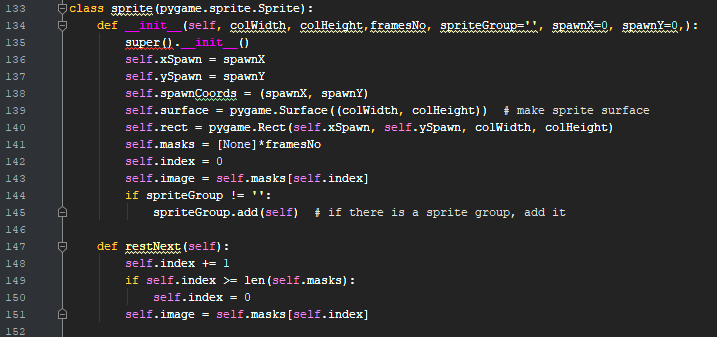
****

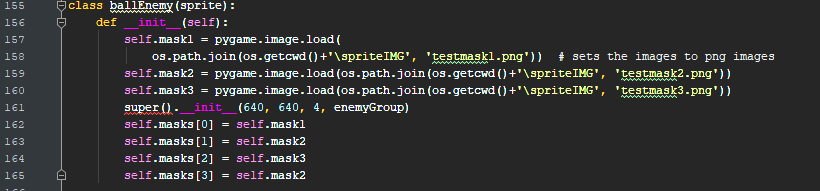
****

**V6**

Changelog:

* ‘Next’ method in sprite changed to ‘restNext’
* testSprite changed to ‘ballEnemy’
* Sprite given optional parameters of spawnX and spawnY

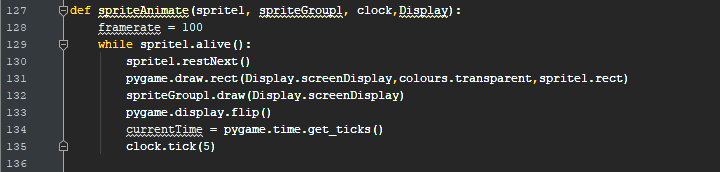


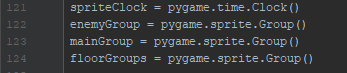


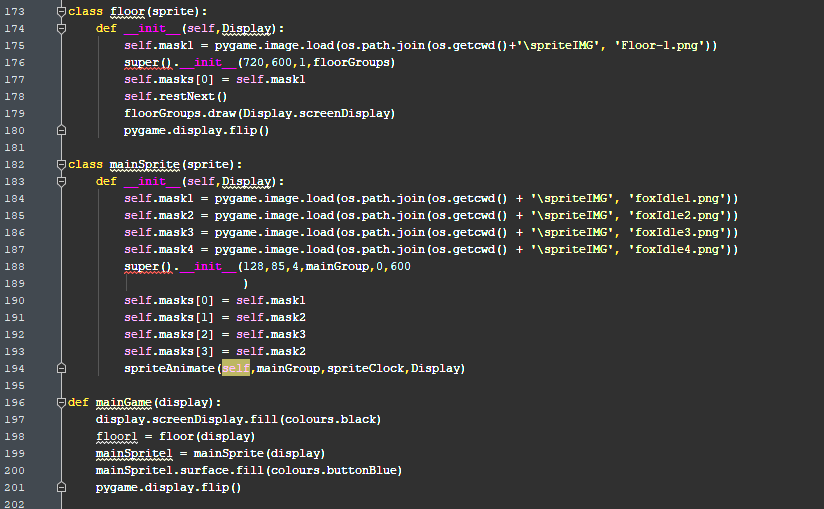
**V7**

Changelog:

* spriteAnimate adjusted in attempt to make animated background transparent
* Three new global spriteGroups for floor, enemy and main
* Floor sprite created
* mainSprite in development
* Instances of floor and mainSprite are created in mainGame

****

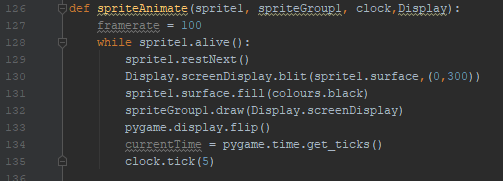
****

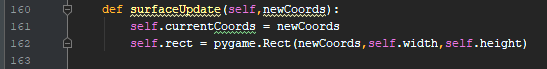
****

**V8**

Changelog:

* Changed Spritegroup1.draw method in spriteAnimate to blit method
* Added surfaceUpdate method to sprite class

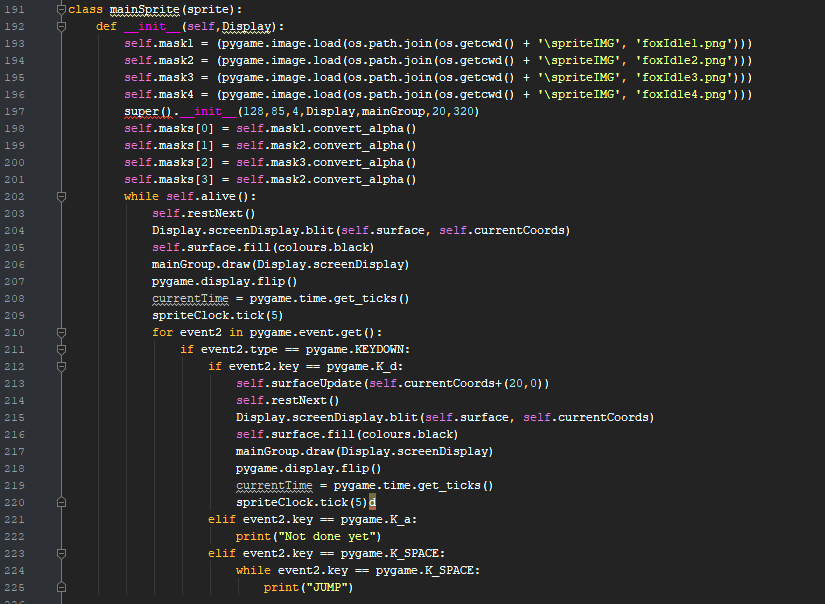
****

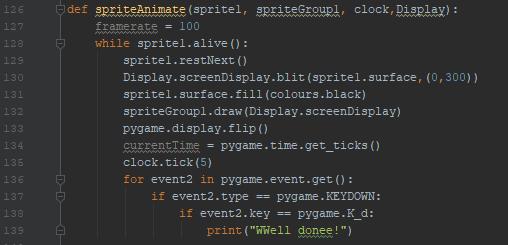
****

**V9**

Changelog:

* Began developing movement
* Development of keys events in both spriteAnimate/mainSprite class. Testing which would be better for this purpose.
* Movement animation in development too

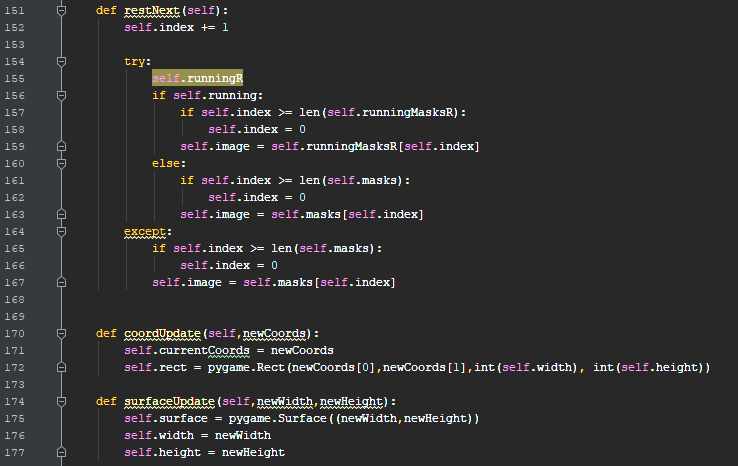
****

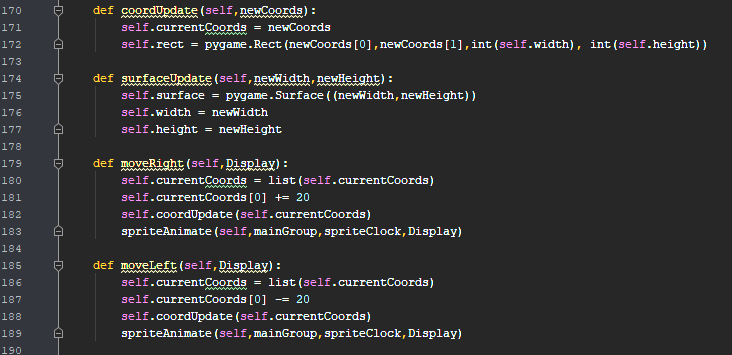
****

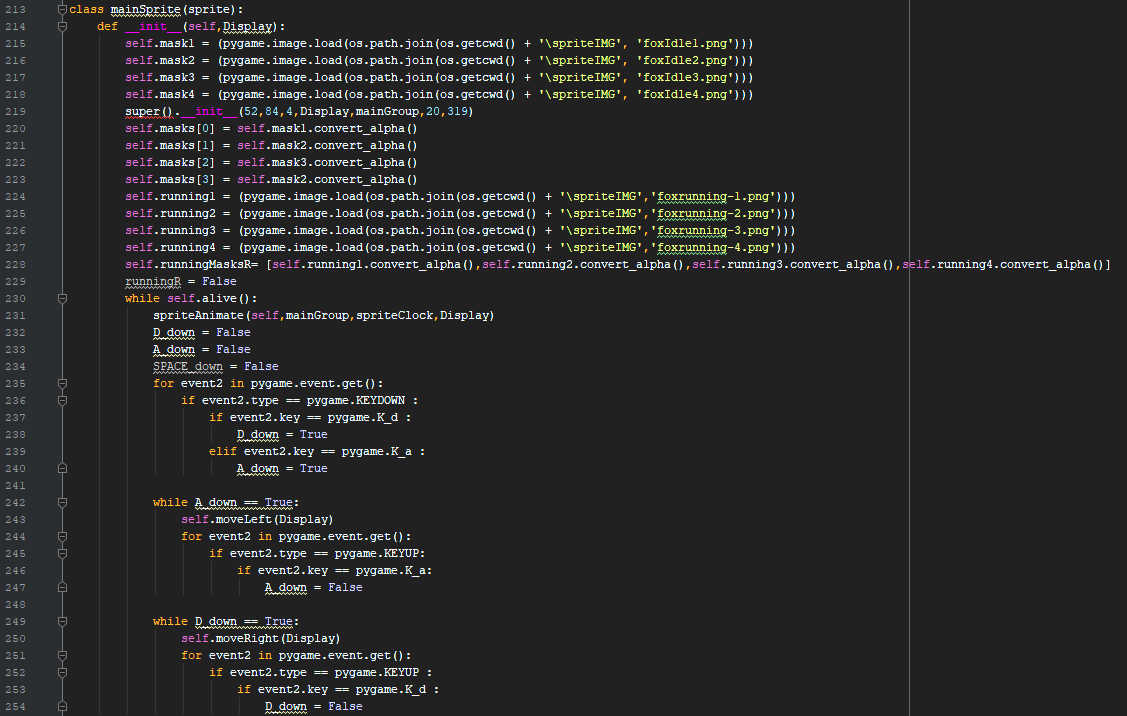
**V10**

Changelog:

* restNext method in sprite class changed to attempt to accommodate running animation
* coordUpdate method added to sprite class
* moveRight method added to sprite class
* moveLeft method added to sprite class
* runningMasksR attribute given to mainSprite class
* Way in which mainSprite moves is changed

****

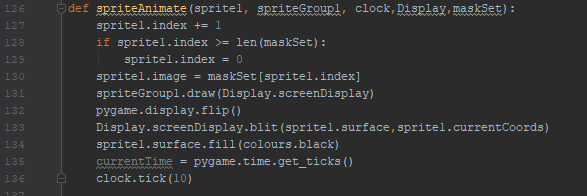
****

****

**V11**

Changelog:

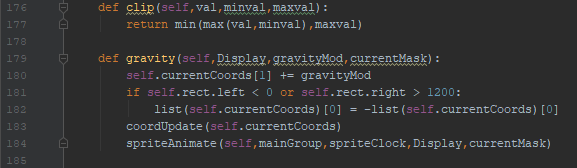
* runningMasksL added to mainSprite class
* spriteAnimate given parameter ‘maskSet’ so spriteAnimate animates everything not just a specific sprite mask set
* next() in spriteAnimate replaced with the contents of the old restNext function, for above reason
* clock.tick() changed from 5 ticks to 10 to quicken animation



**V12 (Final version)**

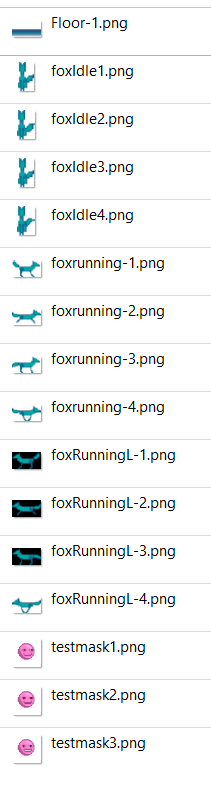
Changelog:

* Clip and gravity methods added to sprite class, in development

****

*Shortly after this, it was decided to abandon this project.*

*Sprite Masks*



*Justifications for changes and initial decisions for Previous game*

Key principles for good user centred design include:

* Intuitive design
* Efficiency
* Memorability - remembering controls
* User satisfaction

The design/pseudocode for the previous project is nowhere near as refined or detailed as the project was abandoned midway through. Pseudocode for project was deleted after project was abandoned.

Justifications for Wireframe Decisions/pseudocode decisions

Decided colour scheme, for user satisfaction as it looks prettier:

Light blue: RGB(124,176,255)

Mid-blue: RGB(89,103,181)

Dark blue: RGB(29,41,81)

Black/Background Colour: RGB(0,0,0)

Decided font family: Candara

*Main Menu*

* Main menu’s title is large and the three buttons on the main menu have clear labels and also have large text for an intuitive and accessible interface for the end users.
* Main menu adheres by colour scheme and font family for user satisfaction
* Main menu was required so save/load features could be implemented for intuitive use (See functional requirements)
* It was also required due to the SQA’s need for a clear user interface.
* The no save file notification will cover over the continue button so the continue button can no longer be used, as there is no save file to use. This increases efficiency.
* In original plans there was an option to close the error message generated when the continue button was clicked without a sav file, but it was scrapped as it was not deemed necessary for the program. (V3)

*Main Game*

* Main game was abandoned due to the time frame and the amount of time left until hand-in
* Gravity was never achieved in game
* Surface update and coord update were made methods of the sprite class to make it easier to move and animate sprites
* Many of the changes were due to unfamiliarity with pygame, so there was a lot of playing around. And testing how pygame worked.
* Escape menu was made to be small enough to look less significant than the main menu and more significant than the game behind it
* It probably had too many options and was too ambitious
* The main game wireframe also was way too ambitious, much of the features included in the wireframe could never have been implemented in the main game.

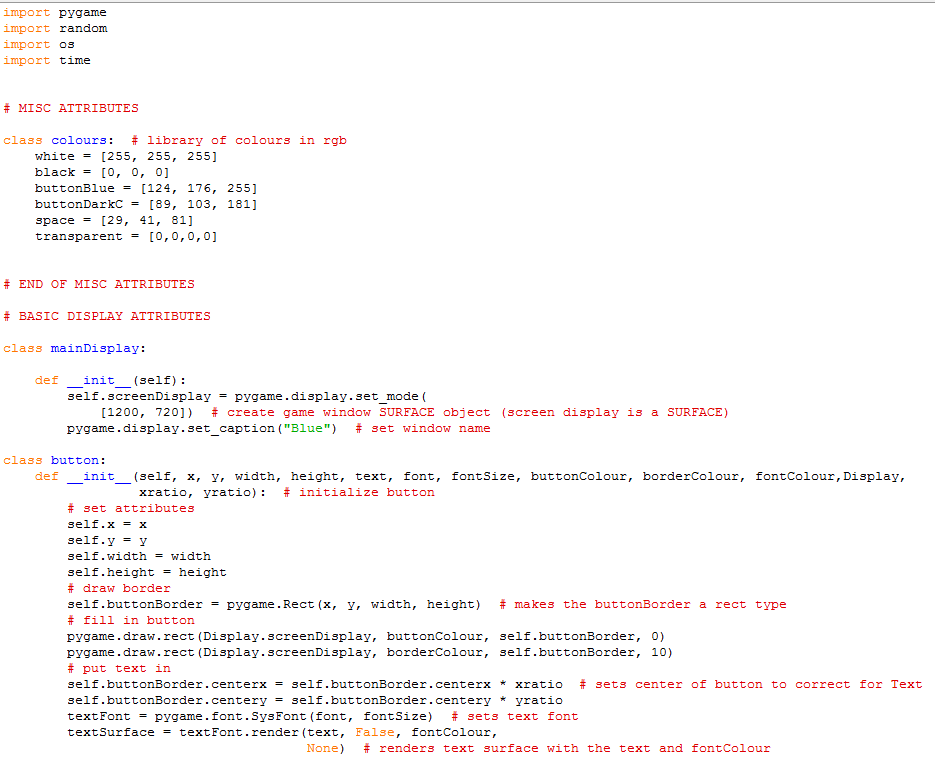
***Maze.py***

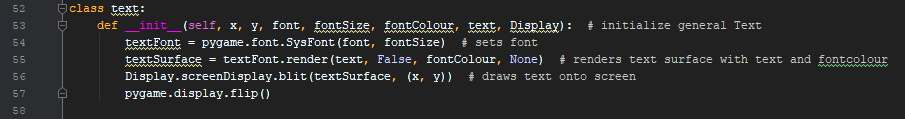
* Changing colours of screenshots are due to different interpreters being used for screenshots, as data dumps were done over time
* Detailed annotation of program is present on V8

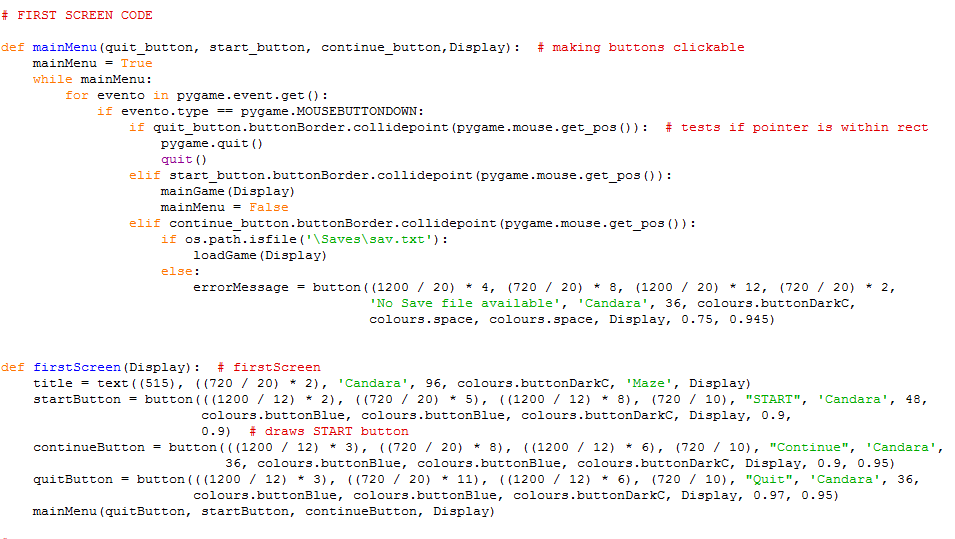
**V1**

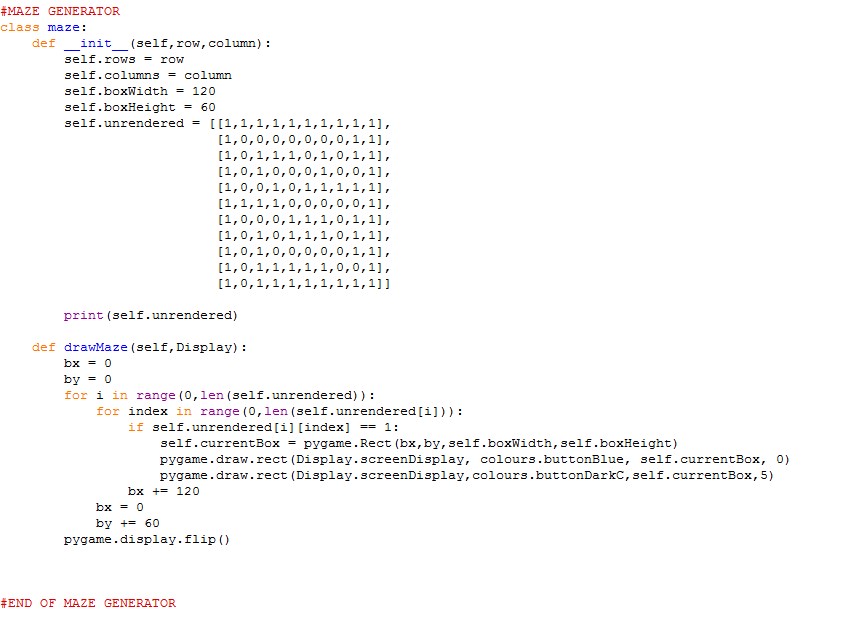
Changelog:

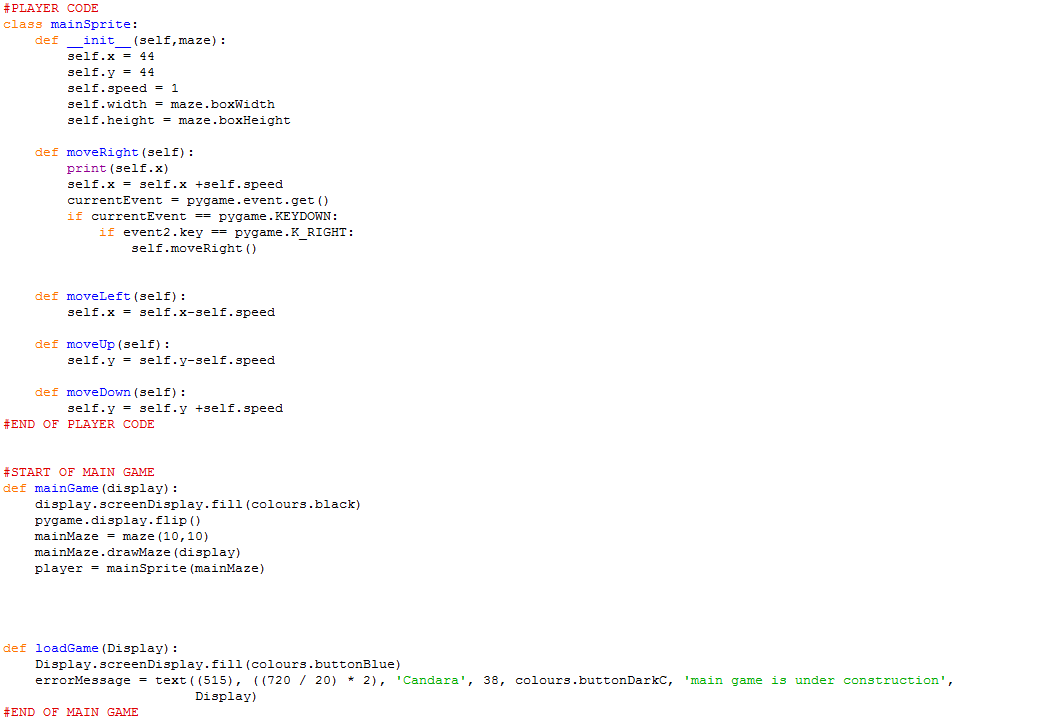
* Imports, colours, mainDisplay, button class, firstScreen,main function, mainMenu and text all copied from previous project
* Maze class implemented with drawMaze method
* mainSprite class implemented, with base attributes and moveLeft,moveUp,moveRight and moveDown methods which are not fully developed
* mainGame implemented, which creates an instance of a maze class and an instance of a mainSprite class. And draws a maze.
* Unrendered attribute of maze class uses ***2D Arrays*** and drawMaze manipulates them

****

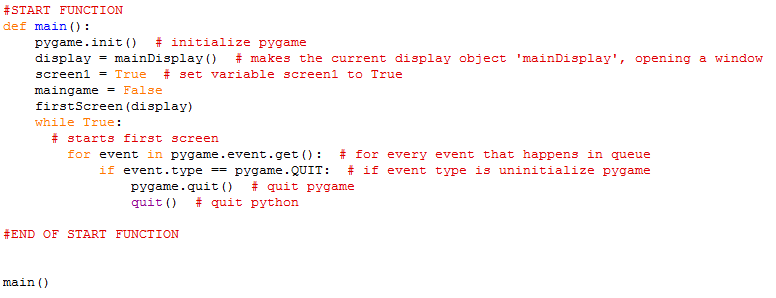
****

****

****

****

***Note that mainSprite class’ movement parts are not fully developed, nor is loadGame***

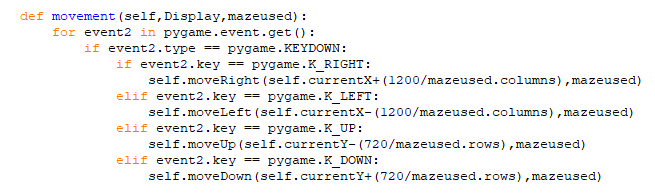
****

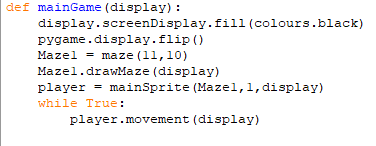
**V2**

Changelog:

* mainSprite class moveRight/moveLeft/moveUp/moveDown methods updated to include ***recursion***. Animation not implemented yet
* mainSprite has method ‘movement’ implemented so that moveRight/moveLeft/moveDown methods are called when certain keys are pressed
* mainGame now calls mainSprite.movement() while True



****

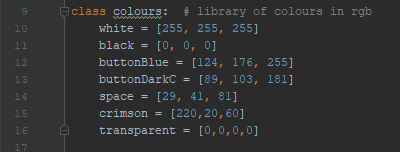
**

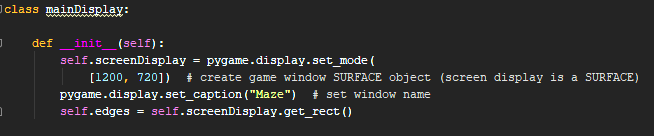
**V3**

Changelog:

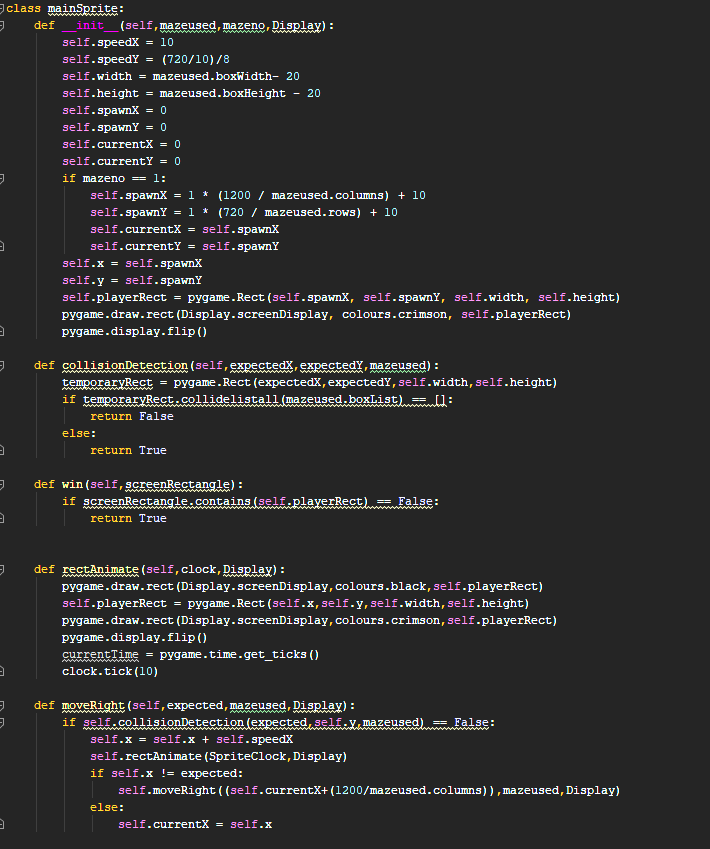
* Colour ‘crimson’ added to colours class
* Edges attribute added to mainDisplay class
* Global spriteClock added to monitor FPS
* mainSprite now has new attributes
* mainSprite now has collisionDetection method
* mainSprite now has win method
* mainSprite now has rectAnimate method
* mainGame function now calls mainSprite.win to test if player has won the maze

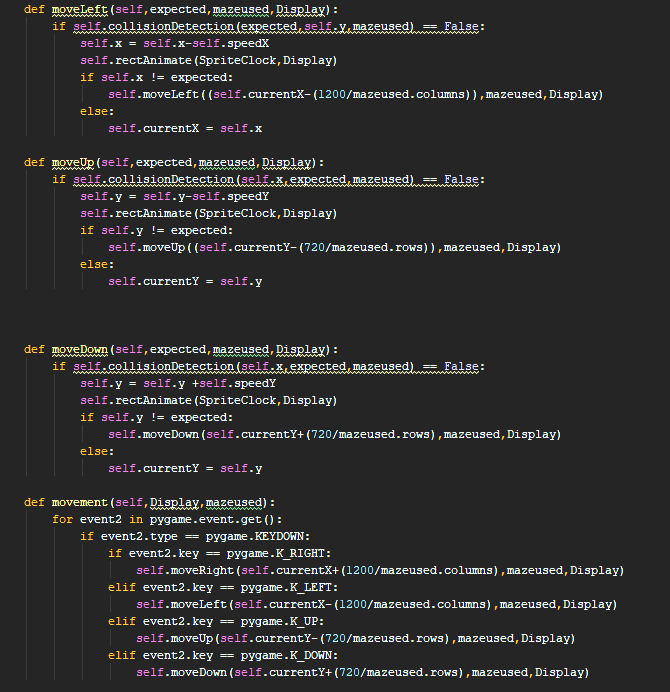
***Changed components:***

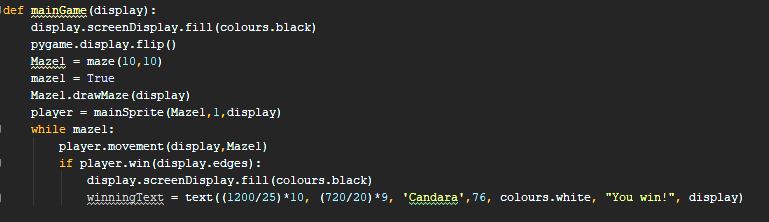
******

**

**

**

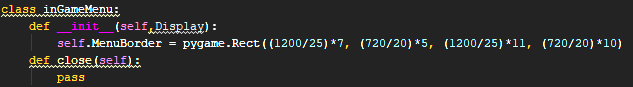
******

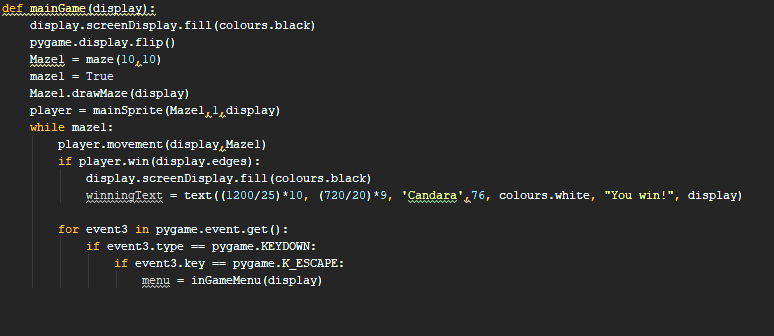
**

**V4**

Changelog:

* In-game menu class added, for now a placeholder
* mainGame function now detects whether the escape key is pressed or not

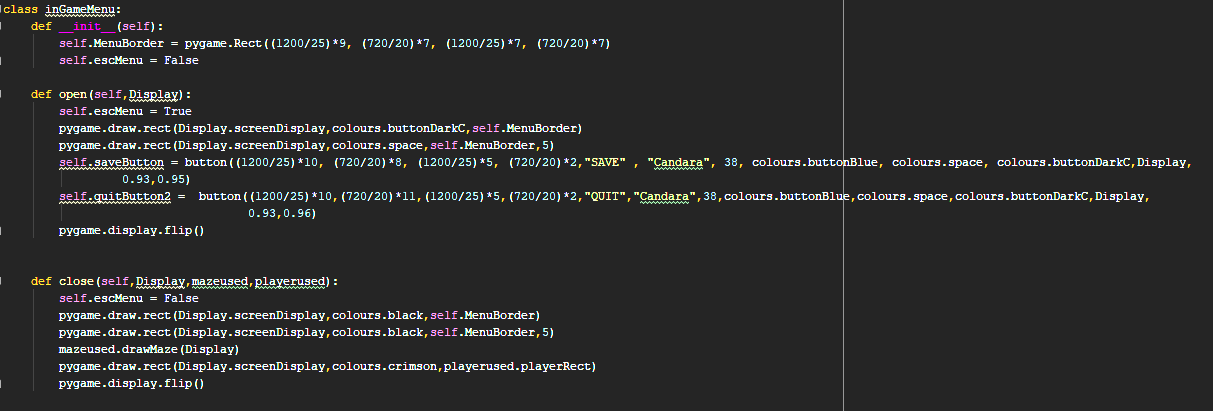
******

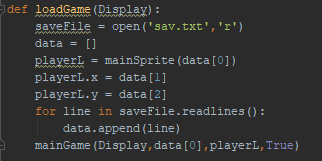
******

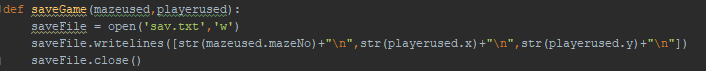
**V5**

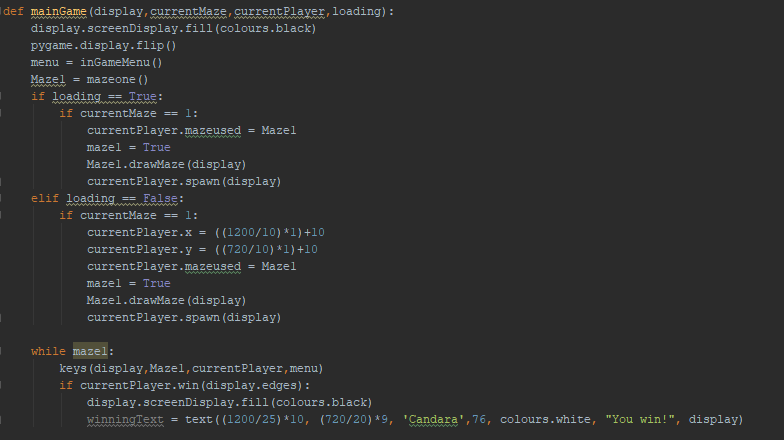
Changelog:

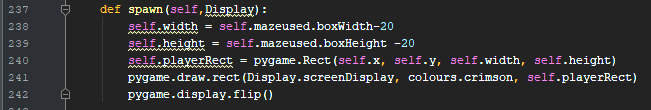
* inGameMenu class now has close() and open() methods, which are fully realized
* loadGame function and saveGame function now load and save data to a file called ‘sav.txt’
* mainSprite now has ‘spawn’ method to control exactly when and where player spawns
* Keys function implemented, so that all events related to the keyboard are in one place
* mainGame function is now changed to accomodate loading;
  + If game is loading, spawn player wherever its x and y coordinates are
  + If game is not loading, spawn player at beginning of maze
  + Player instance is created outside of mainGame function now, in order to accommodate new loading functions.
  + New ‘mazeone’ subclass of maze class created

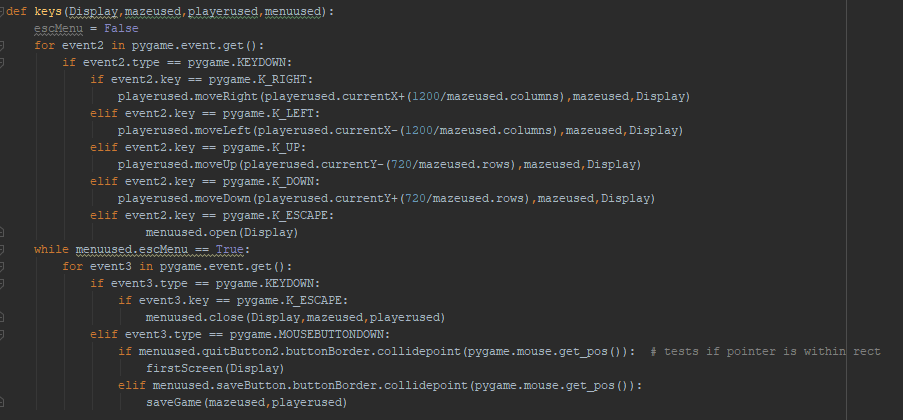
**

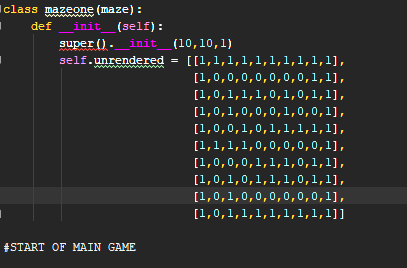
**

**

**

**

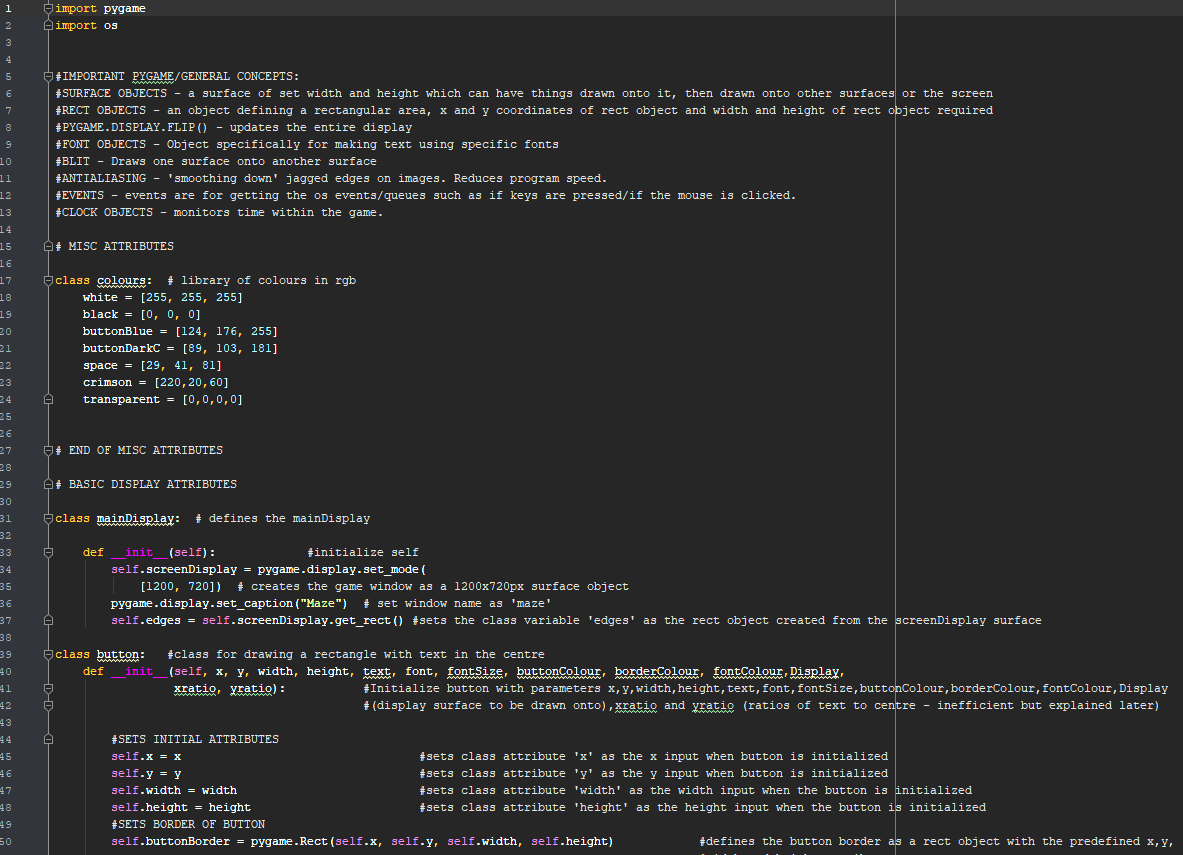
**

**

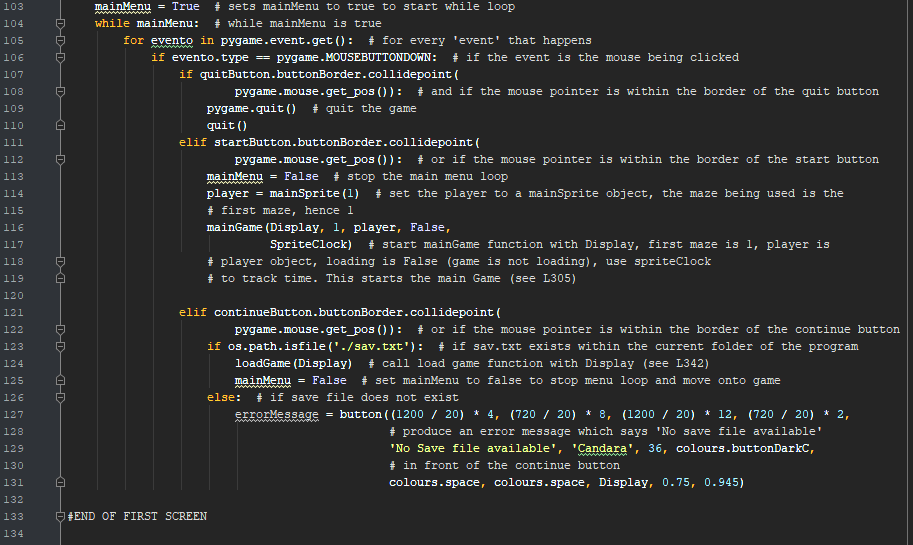
**V6**

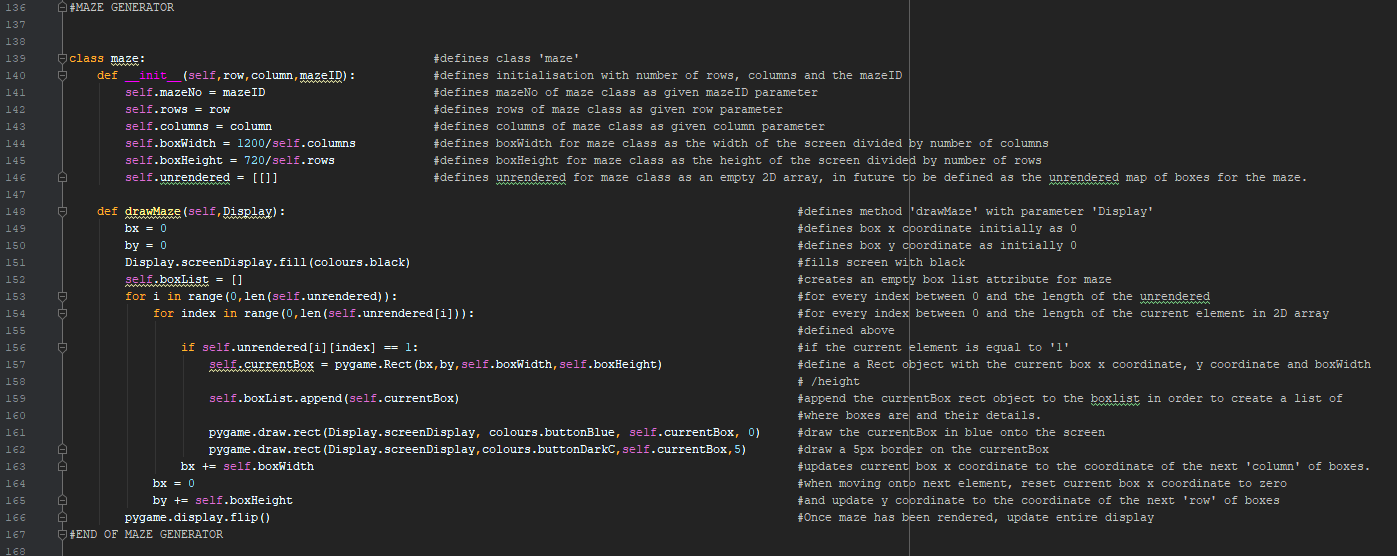
***V6 includes full code annotation. View this version for fuller understanding of code as a whole***

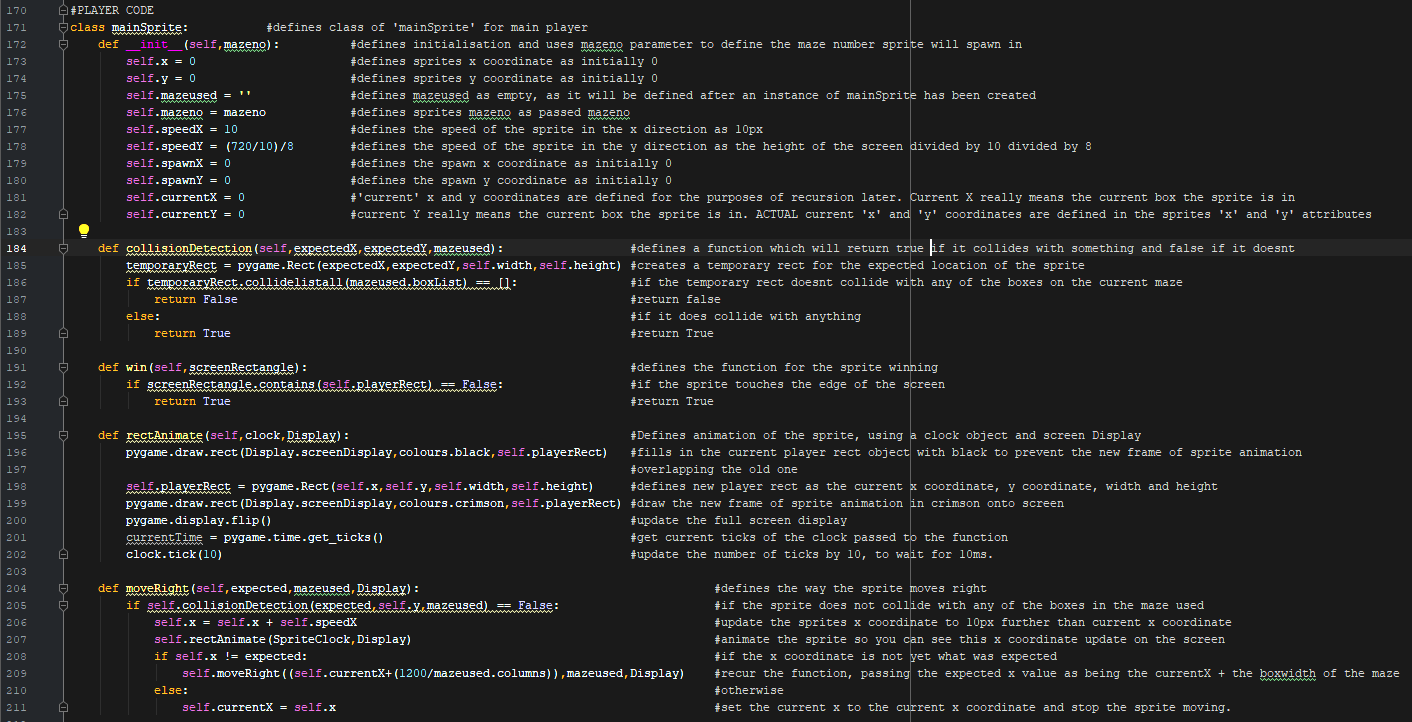
***All features unchanged, this is just the full program with annotation - this is apart from the firstScreen function which was originally the firstScreen function and the mainMenu function, but has been combined.***

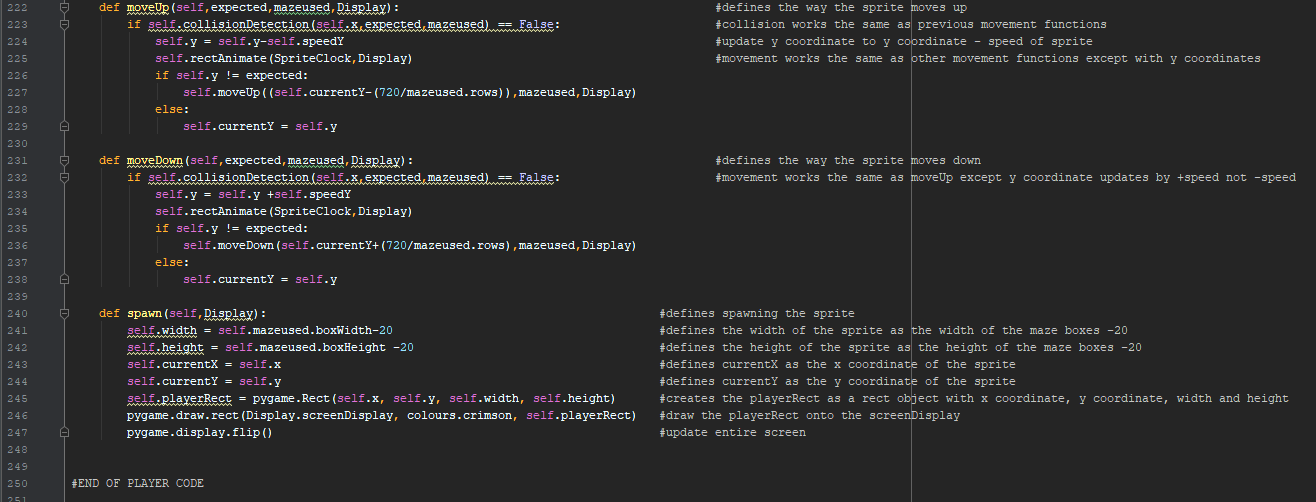
****

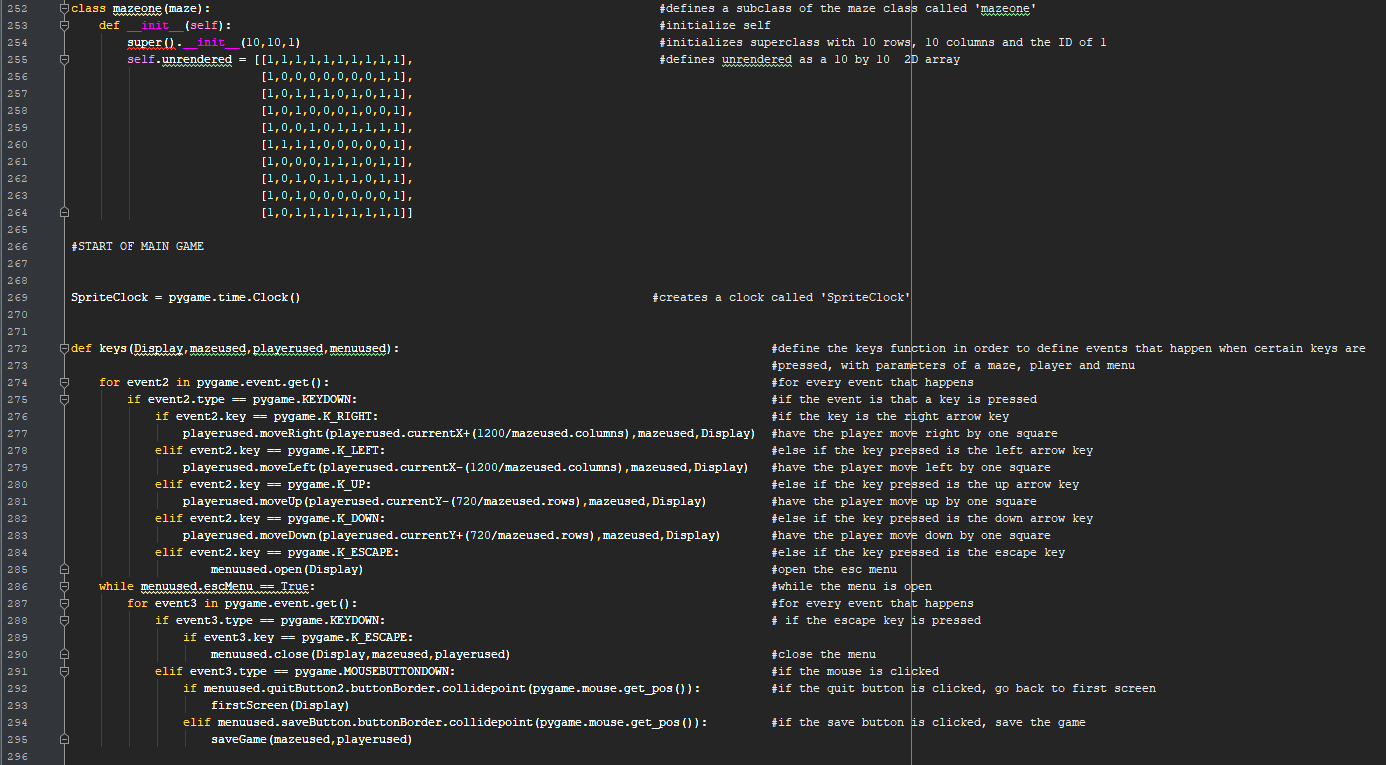
****

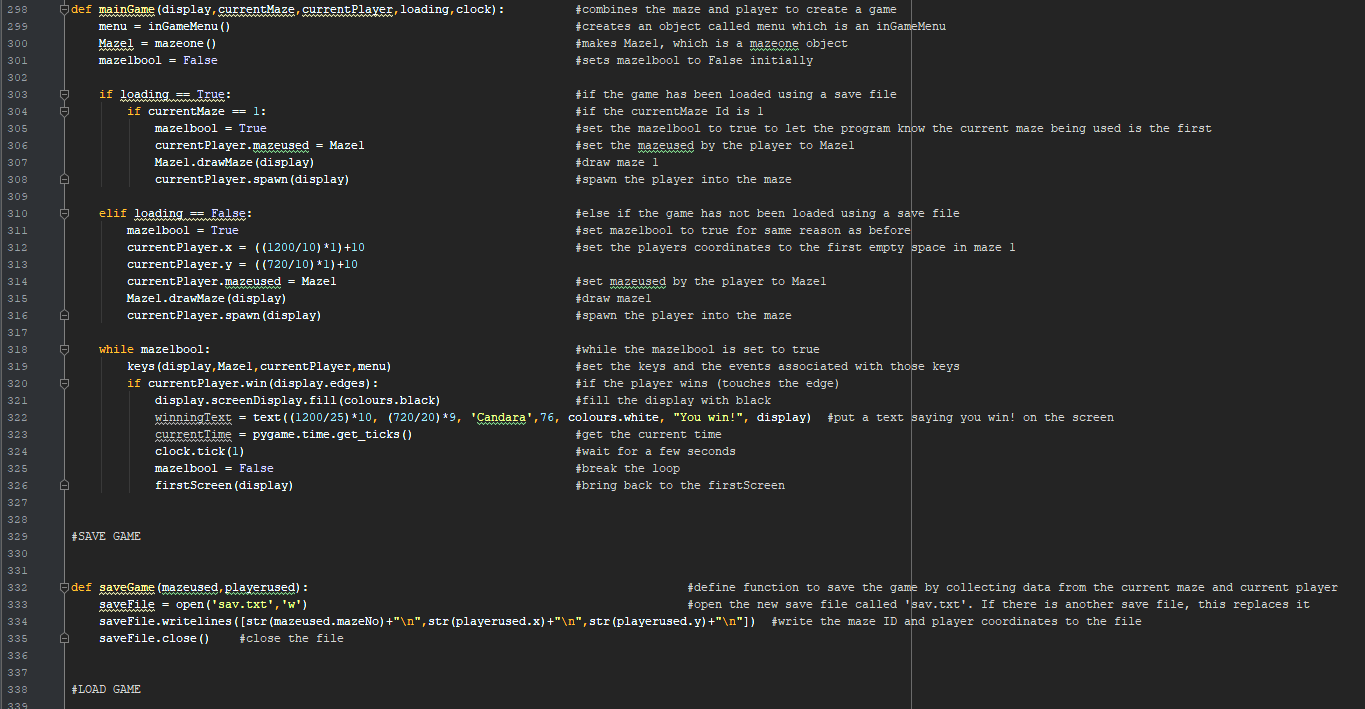
****

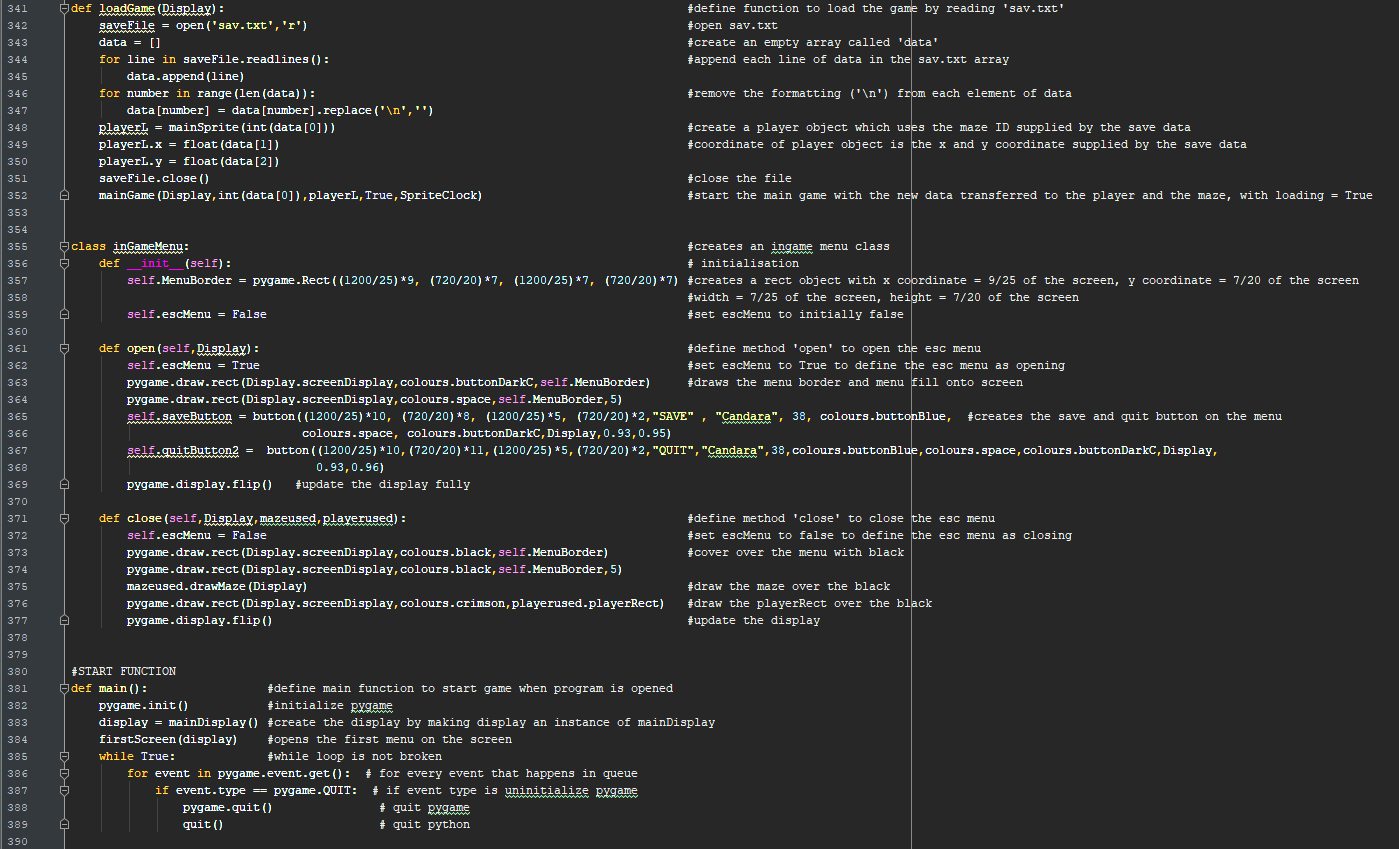


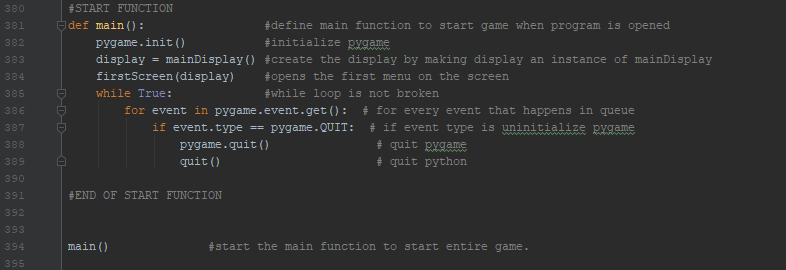
****

****

****

****

****

****

*Justifications for changes and initial decisions for maze game*

Key principles for good user centred design include:

* Intuitive design
* Efficiency
* Memorability - remembering controls
* User satisfaction

Justifications for Wireframe Decisions

Decided colour scheme, for user satisfaction as it looks prettier:

Light blue: RGB(124,176,255)

Mid-blue: RGB(89,103,181)

Dark blue: RGB(29,41,81)

Crimson: RGB(220,20,60)

Black/Background Colour: RGB(0,0,0)

Decided font family: Candara

*Main Menu*

* Main menu’s title is large and the three buttons on the main menu have clear labels and also have large text for an intuitive and accessible interface for the end users.
* Main menu adheres by colour scheme and font family for user satisfaction
* Main menu was required so save/load features could be implemented for intuitive use (See functional requirements)
* It was also required due to the SQA’s need for a clear user interface.
* The no save file notification will cover over the continue button so the continue button can no longer be used, as there is no save file to use. This increases efficiency.

*Maze Game*

* The maze is 10 by 10 due to requirements specifications
* The maze colours adhere by the colour scheme previously defined. This is to increase user satisfaction. The same applies for player colours.

*Esc Menu*

* The esc menu adheres by the colour scheme and font family for user satisfaction
* Button font is large and each buttons is labeled for intuitive design and accessibility
* Esc menu was required so save/load features could be implemented for intuitive use (See functional requirements)
* Save feature also exists so that in further development (Past the beta development stage that will define the end of the project), the game can be saved and loaded easily.

*Winning Screen*

* White on black text to make it easily readable
* Win screen returns to the main menu after a few seconds so the user can restart the game.

*Changes made*

Buttons:

* Buttons text centering system was changed due to xratio/yratio system being inefficient
* Buttons font parameter was removed due to all buttons having same font
* Drawing of button was separated from initialisation to make it easier to use.

Justifications for pseudocode

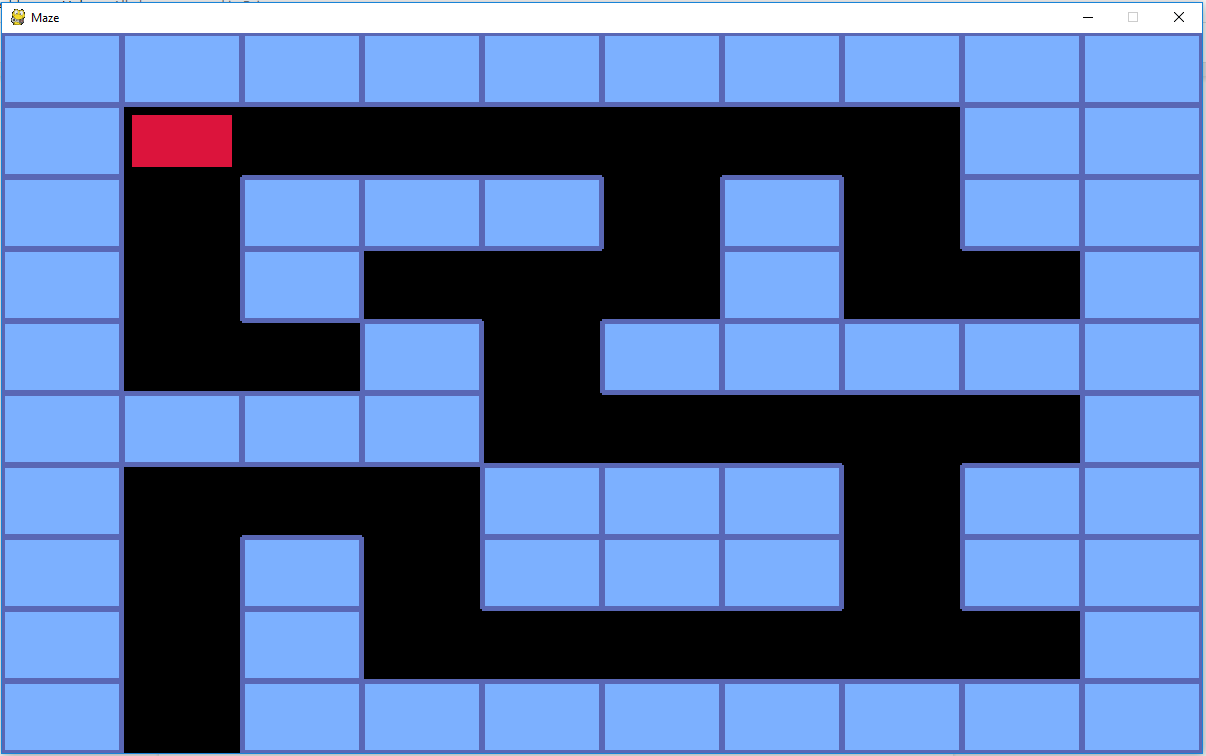
* The x,y coordinates and width,height variables are in the form (1200/24)\*x,(720/20)\*y to create a grid based coordinate system. This is to make it easier to transfer from wireframe to code.
* Wireframes have grids of 24\*20, which shows where on the grid each button/text needs to be. A grid was more efficient to create a user friendly user interface (as required in requirements specifications.
* Mazeone was made to be a subclass to make it easier to integrate other mazes in the future. For future end user enjoyment.
* All keys were assigned in one function so that more than one event loop had to be acknowledged at once.
* Firstscreen was initially in two parts: firstScreen and mainMenu. I realised it would be more efficient to integrate the two.
* Movement of mainSprite was initially a method in mainSprite. However, due to the way python detected events, it was decided more efficient if all key inputs were put into one function (i.e keys() function in V5)
* Creation of mainSprite instance was changed to be from within the mainGame to the loadGame or mainMenu functions, since the attributes of the two players depending on whether game is being loaded or not is different
* mainSprite given spawn method so that the spawning of the player can be more easily controlled.

***End Game screenshots:***

Main menu, continue not clicked



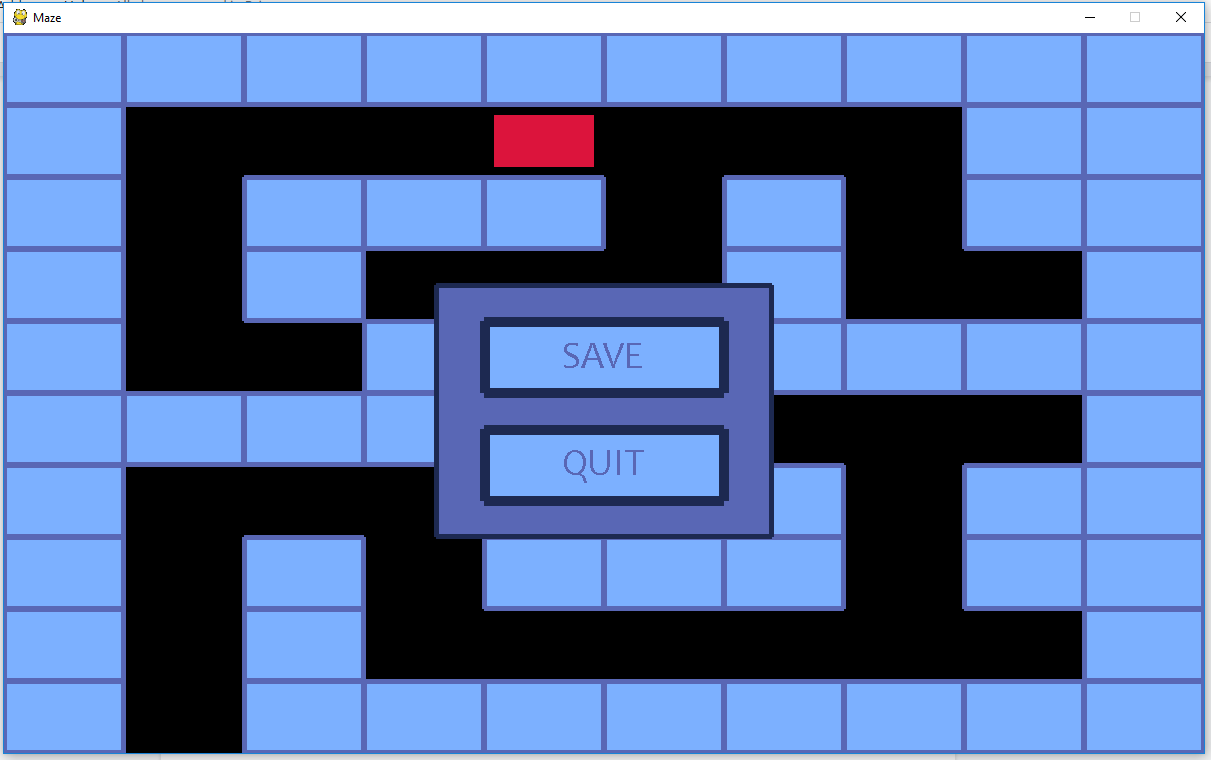
Maze, with player in spawn position



Main menu, with continue button pressed without a save file



Game with in-game menu open.



You win! Screen

