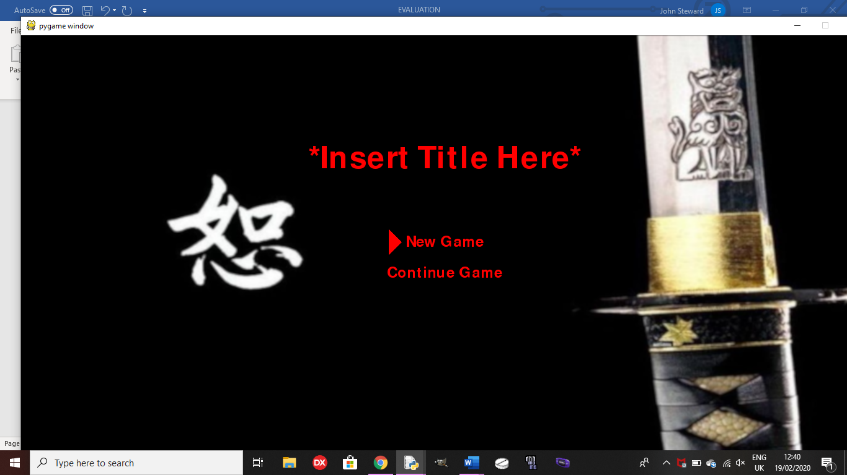
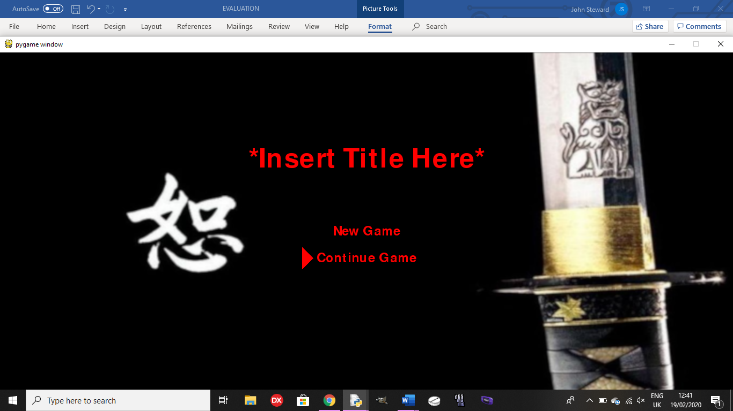
**Evaluation**

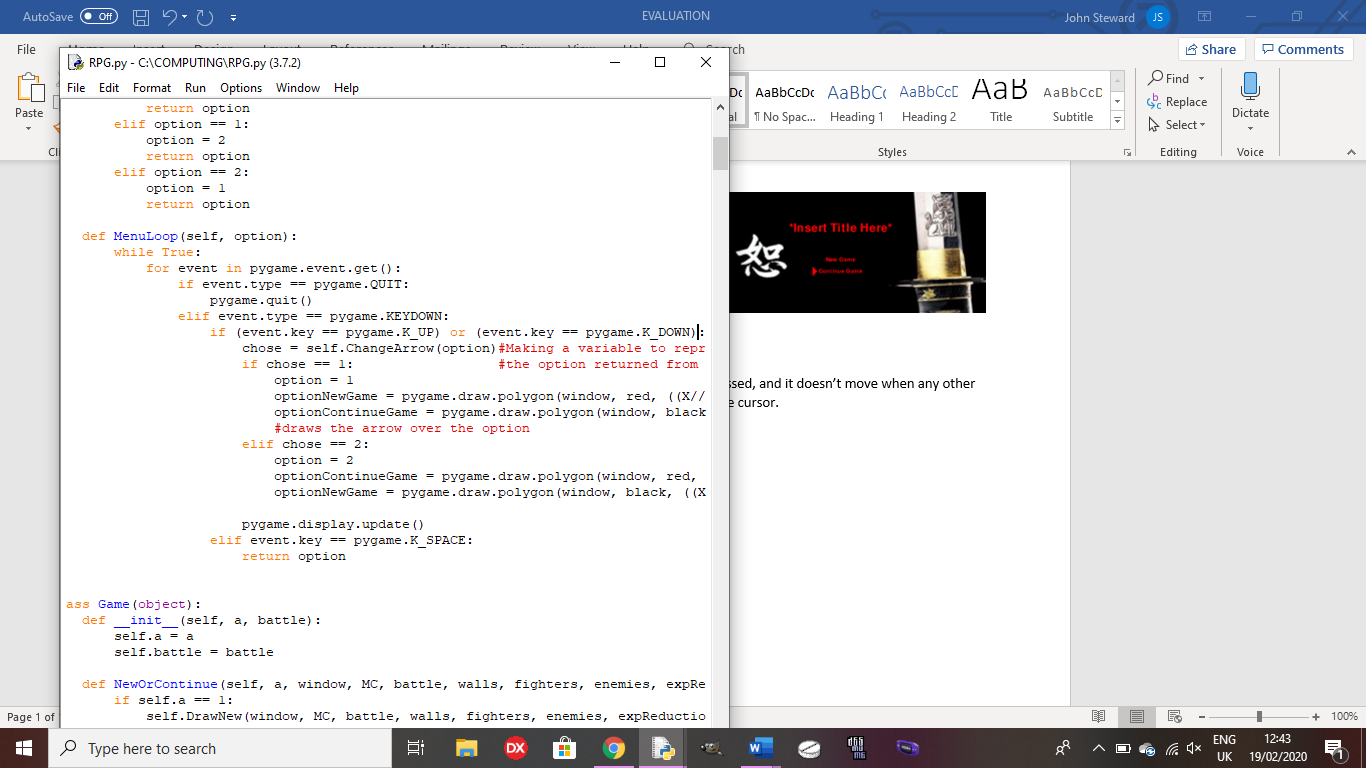
**Functional Testing**

Test 1: Is the main menu drawn and does the cursor move?

Outcome:



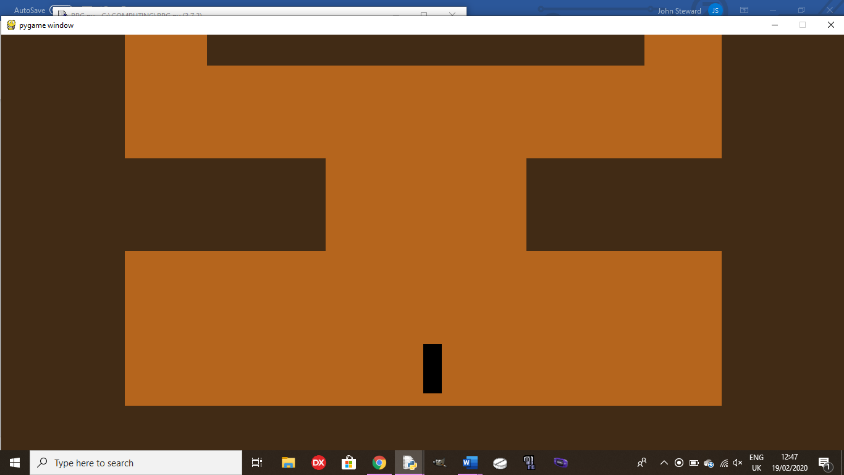
The cursor moves when the up or down arrow keys are pressed, and it doesn’t move when any other key is pressed as it only checks for these keys for moving the cursor.

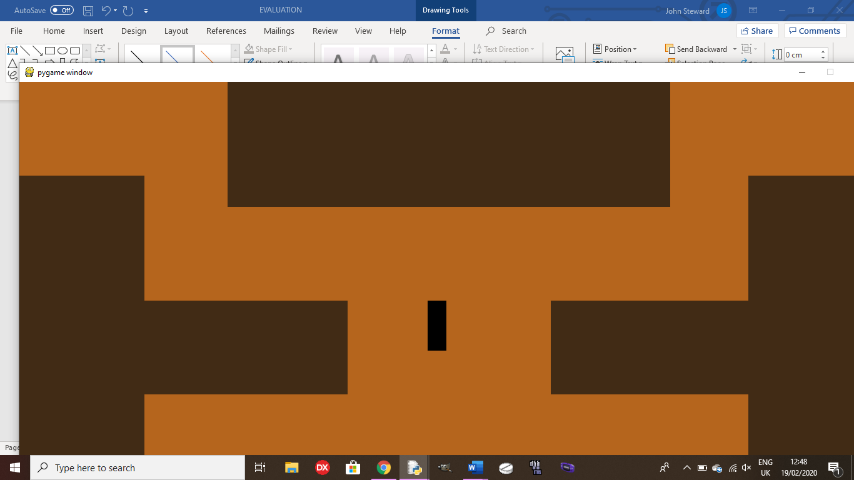


Test 2: when the player chooses an option on the menu, does it go to the correct function?

Outcome: See test 5 in testing for development.

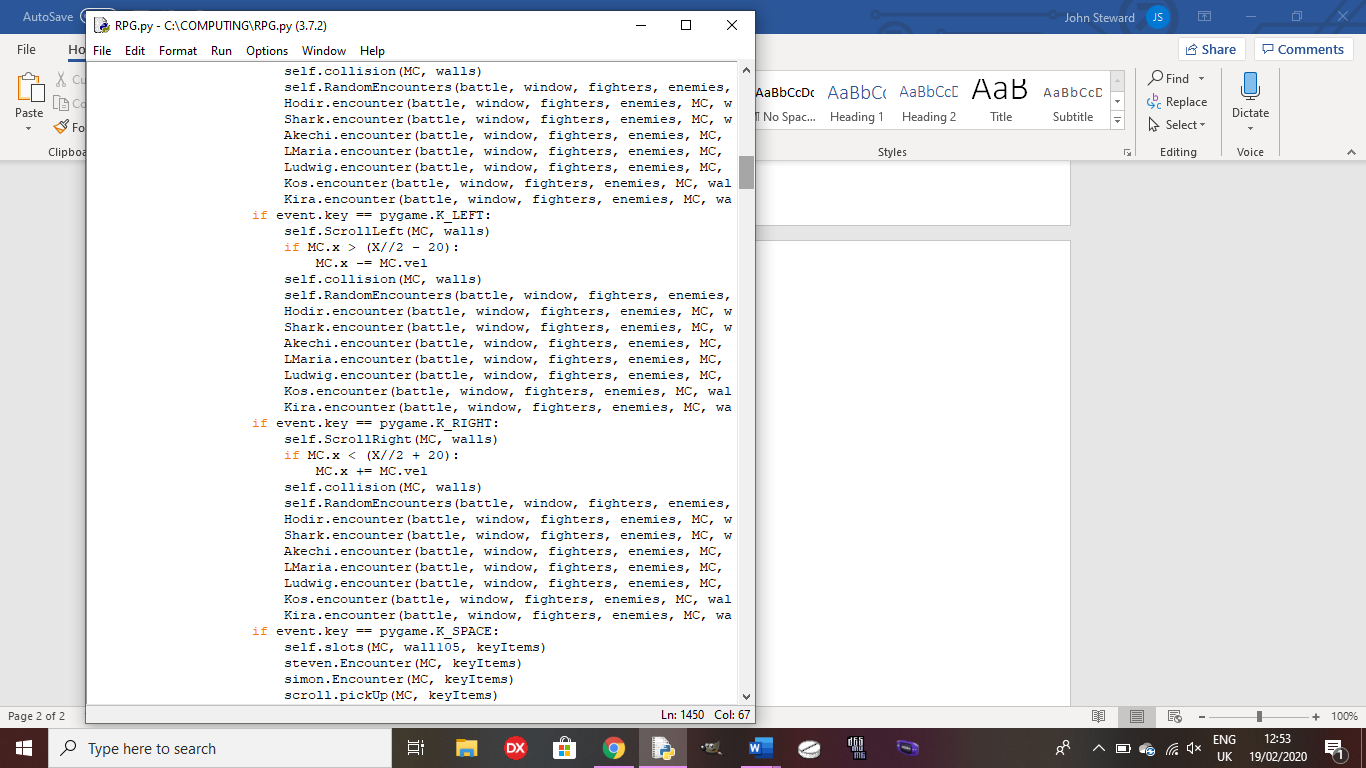
Test 3: Does the player move only if the arrow keys are pressed and do the walls scroll?

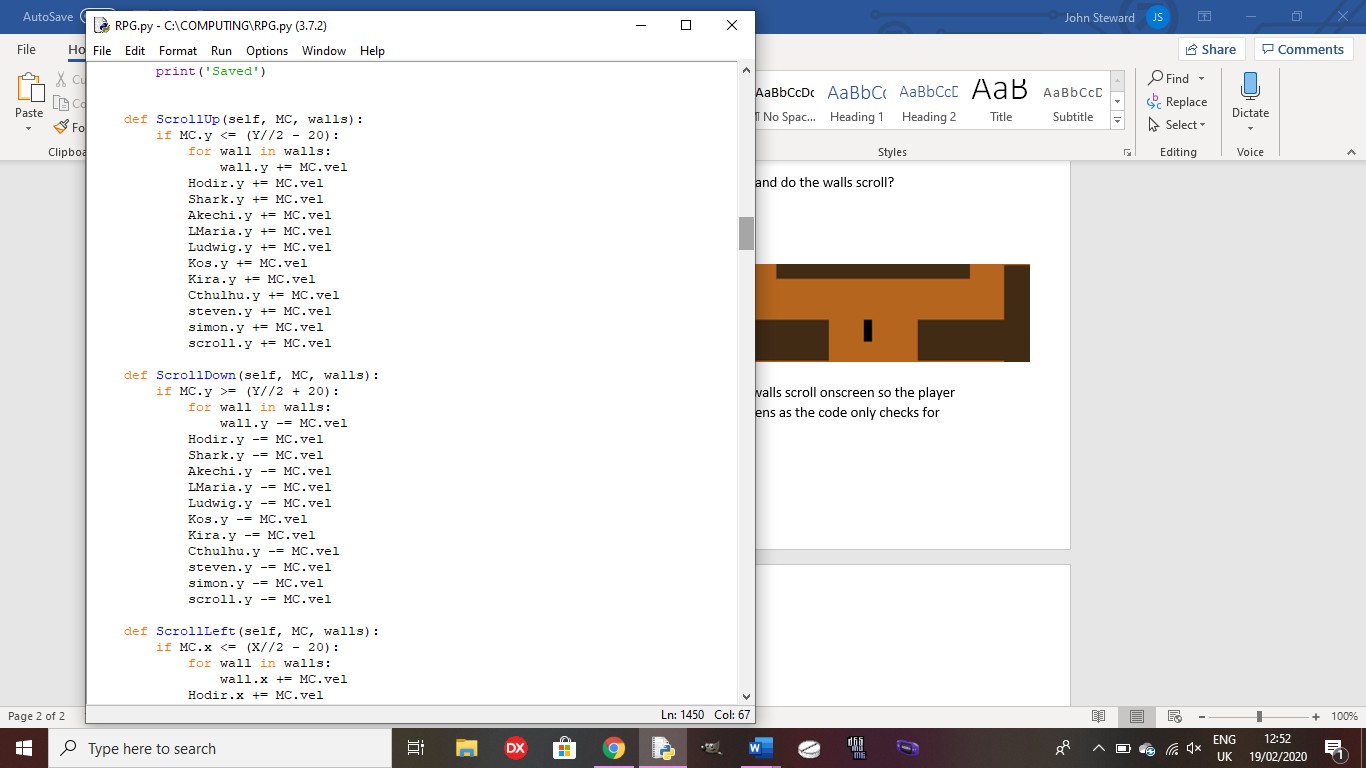
Outcome:



The character moves when the arrow keys are pressed and the walls scroll onscreen so the player doesn’t move offscreen. If other keys are pressed, nothing happens as the code only checks for specific keys, so if other keys are pressed, nothing happens. Links to my HIPO chart for movement being controlled by arrow keys.

Scrolling: Moving:

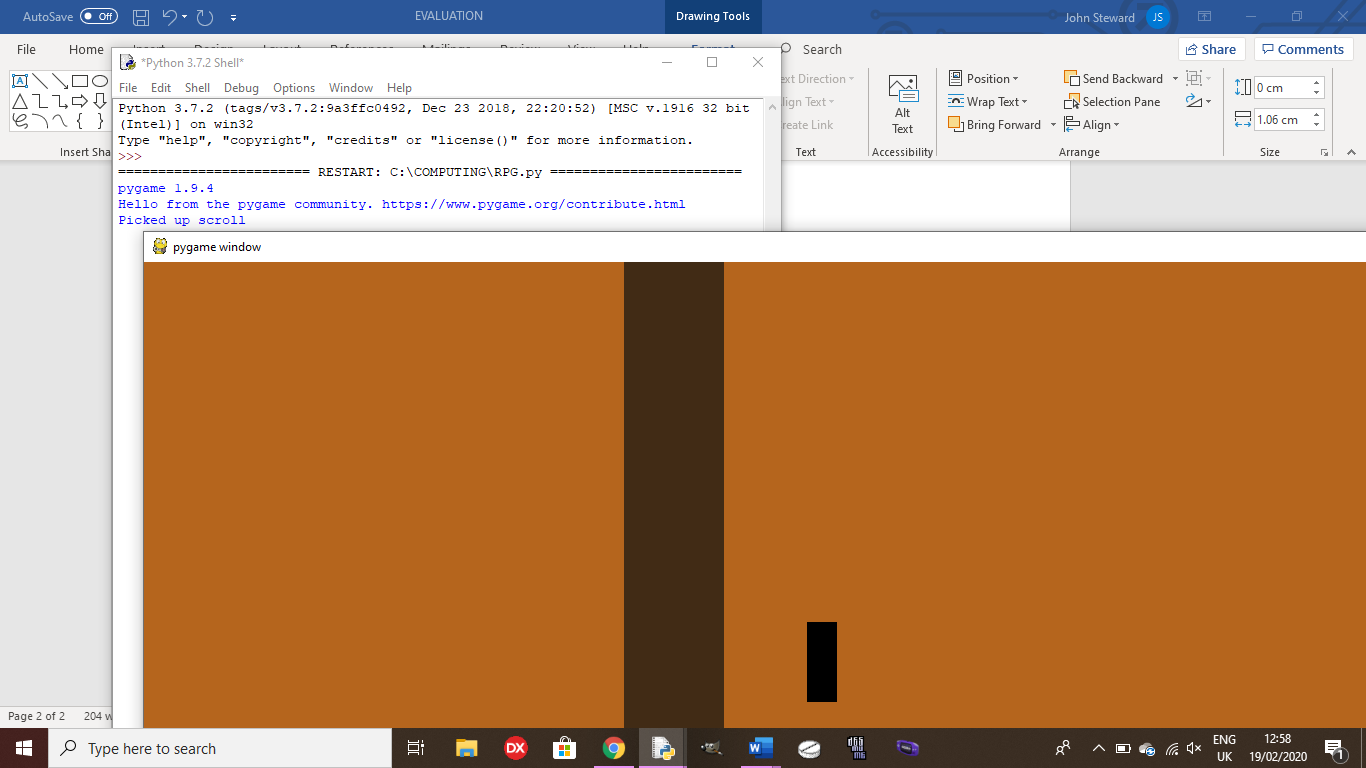


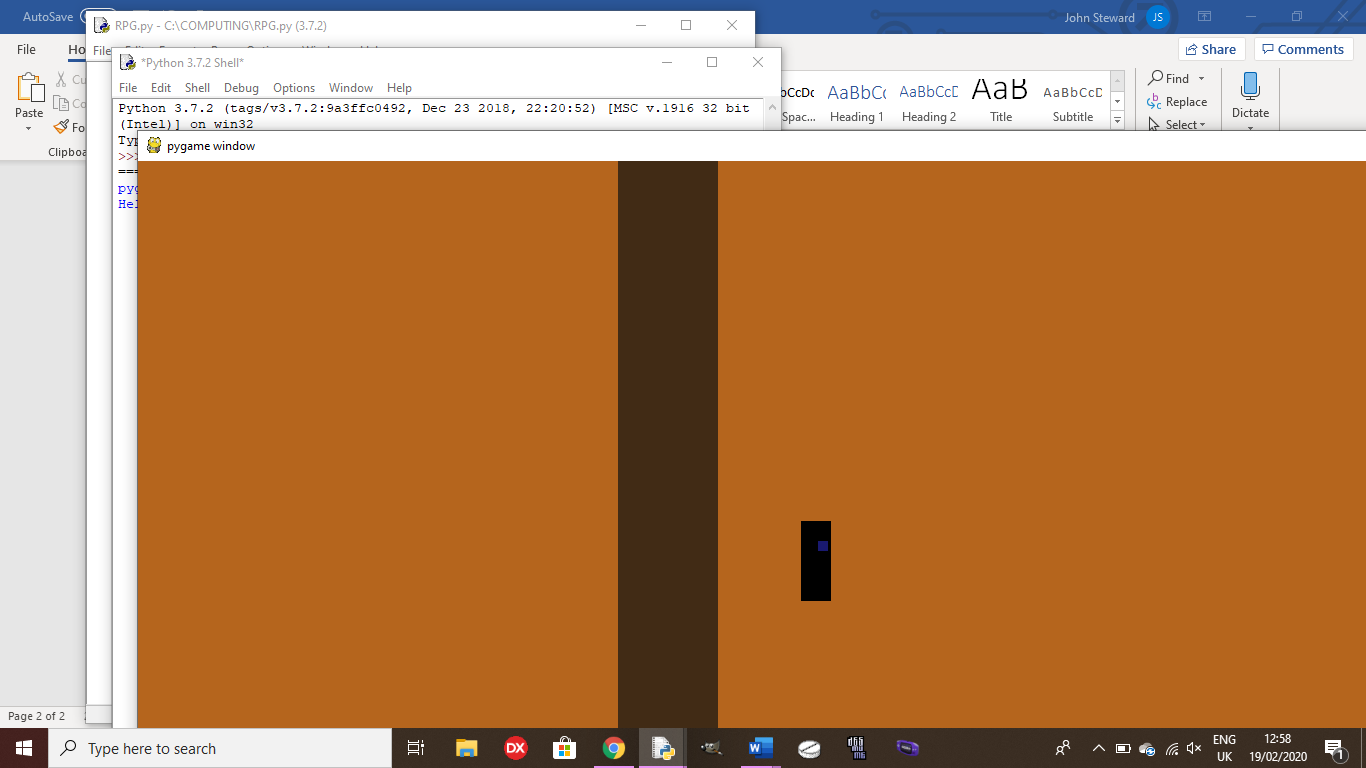
It checks for scrolling before moving the character so, if they have moved too far, only the walls will move, not the player, creating the illusion of movement from the player.

Test 4: Does the collision on the walls work?

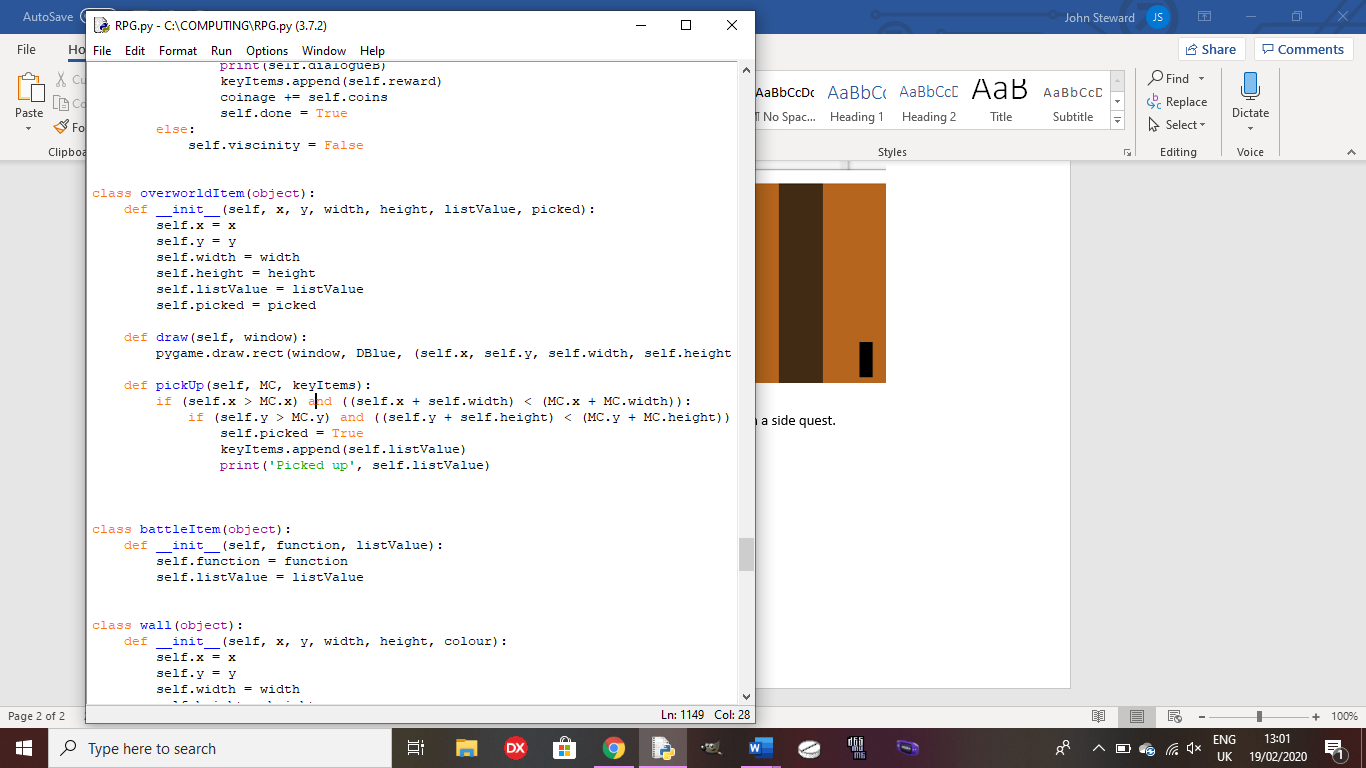
Outcome: See test 8 in testing for development. Links to my HIPO chart which says: ‘cannot move through solid objects’.

Test 5: Does the player pick up items when space is pressed while standing over one?

Outcome:



The scroll is added to the player’s key items array for later use in a side quest.



Test 5: Can the player talk to NPCs and complete side quests?

Outcome: See test 20 and 22 in testing for development. Links to my HIPO chart for dialogue from NPCs shown on the python shell when completing a sidequest.

Test 6: Does the player trigger boss fights when they have moved near the overworld sprite of the boss?

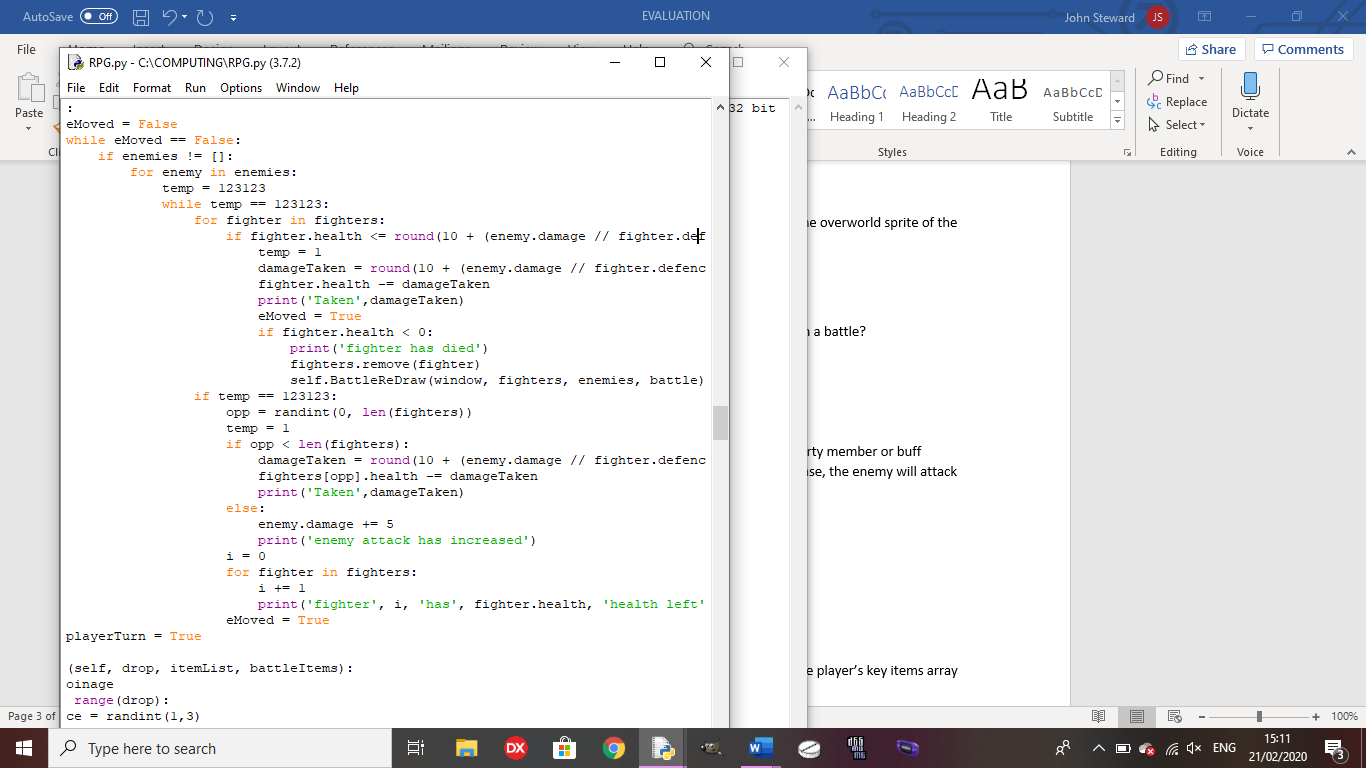
Outcome: See test 17 in testing for development. Links to the scripted events part in my HIPO chart for boss fights and dialogue.

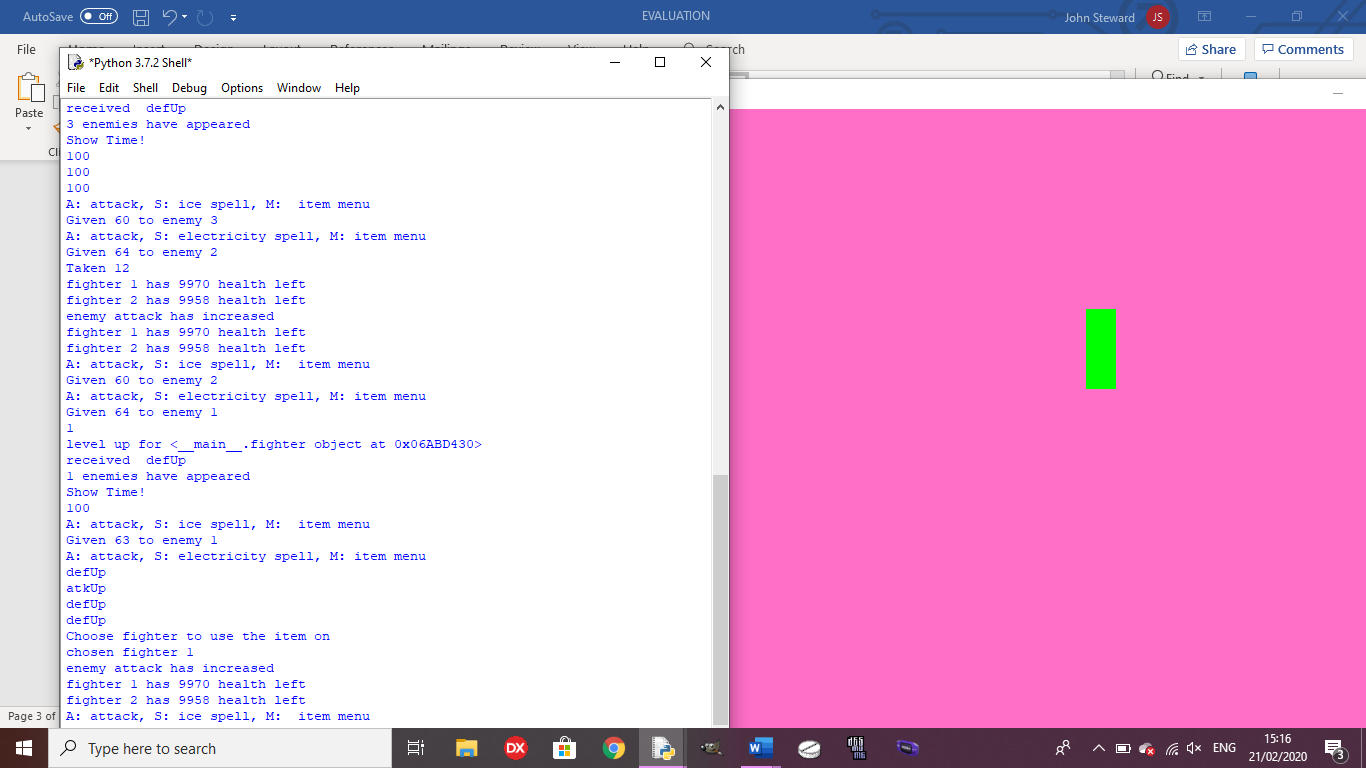
Test 7: Does the battle screen get drawn and allow the player to attack in a battle?

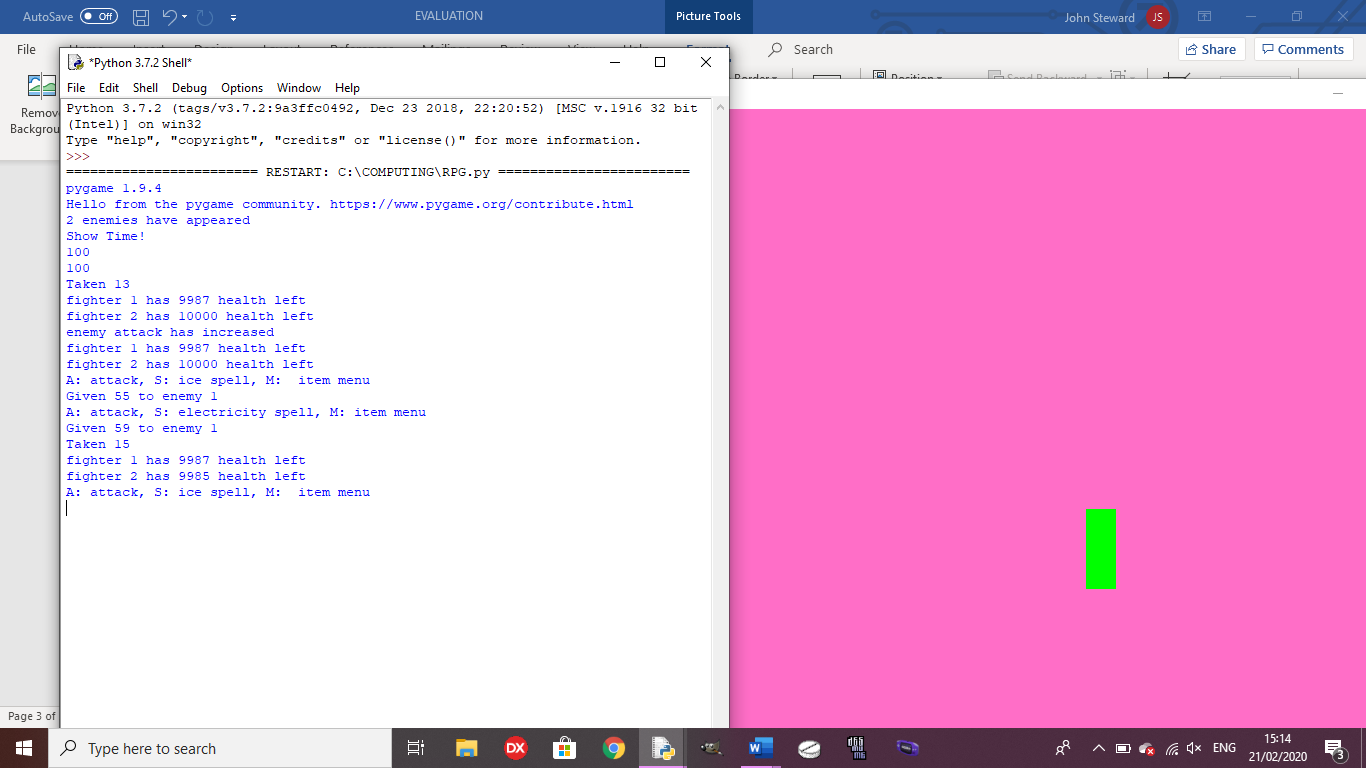
Outcome: See test 9 in testing for development.

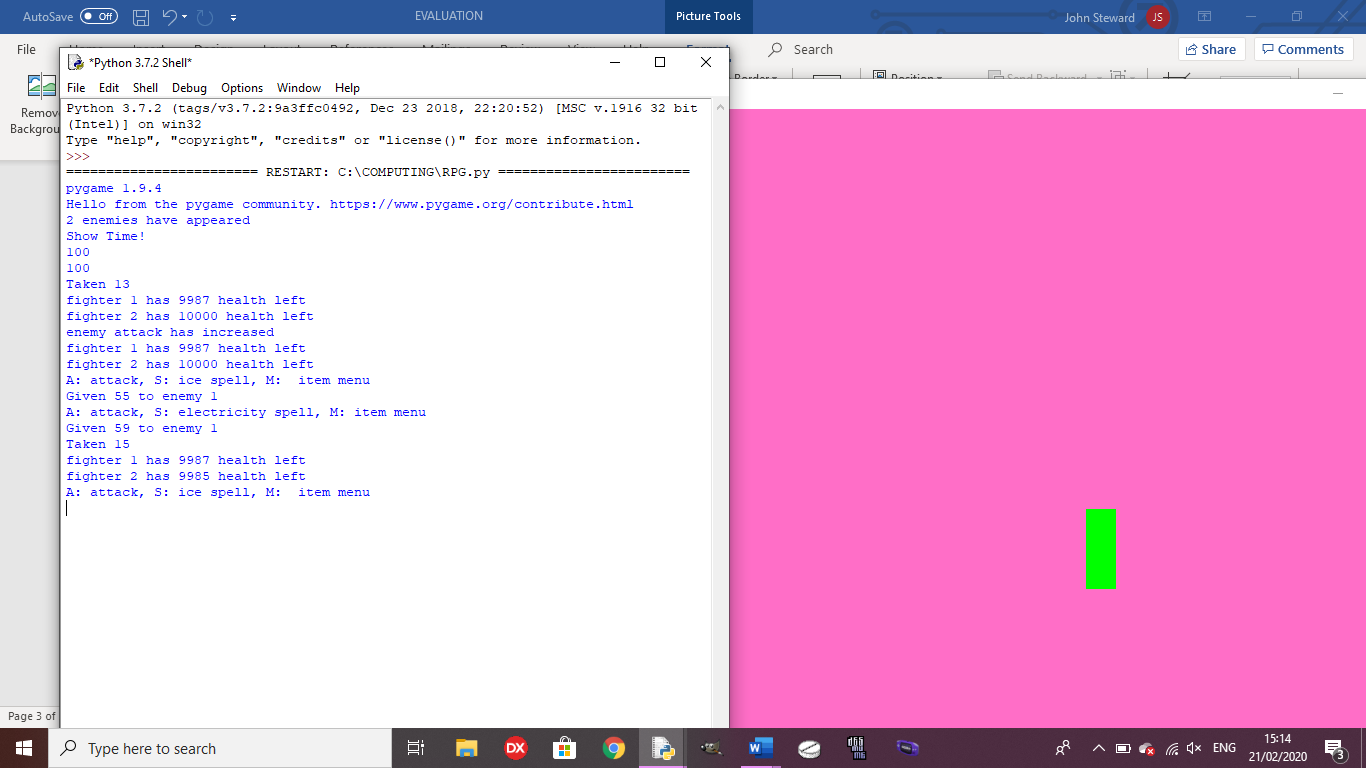
Test 8: Do enemies attack the player or buff themselves in battle?

Outcome: When the enemies have a turn, they will randomly attack a party member or buff themselves, unless one of the party members has low health, in which case, the enemy will attack that party member, defeating them. Links to battles in my HIPO chart which shows the different commands for enemies and players.









Test 9: Does the player gain EXP and level up upon gaining enough EXP?

Outcome: See test 13 and 14 in testing for development.

Test 10: Does the correct ending trigger depending on the contents of the player’s key items array and the number of party members?

Outcome: See test 24 in testing for development. Links to my flow chart which shows that there are different endings depending on the items held by the player.

Test 11: Does the game save and return to the correct point when re-loading the game?

Outcome: See tests 25 and 26 in testing for development.

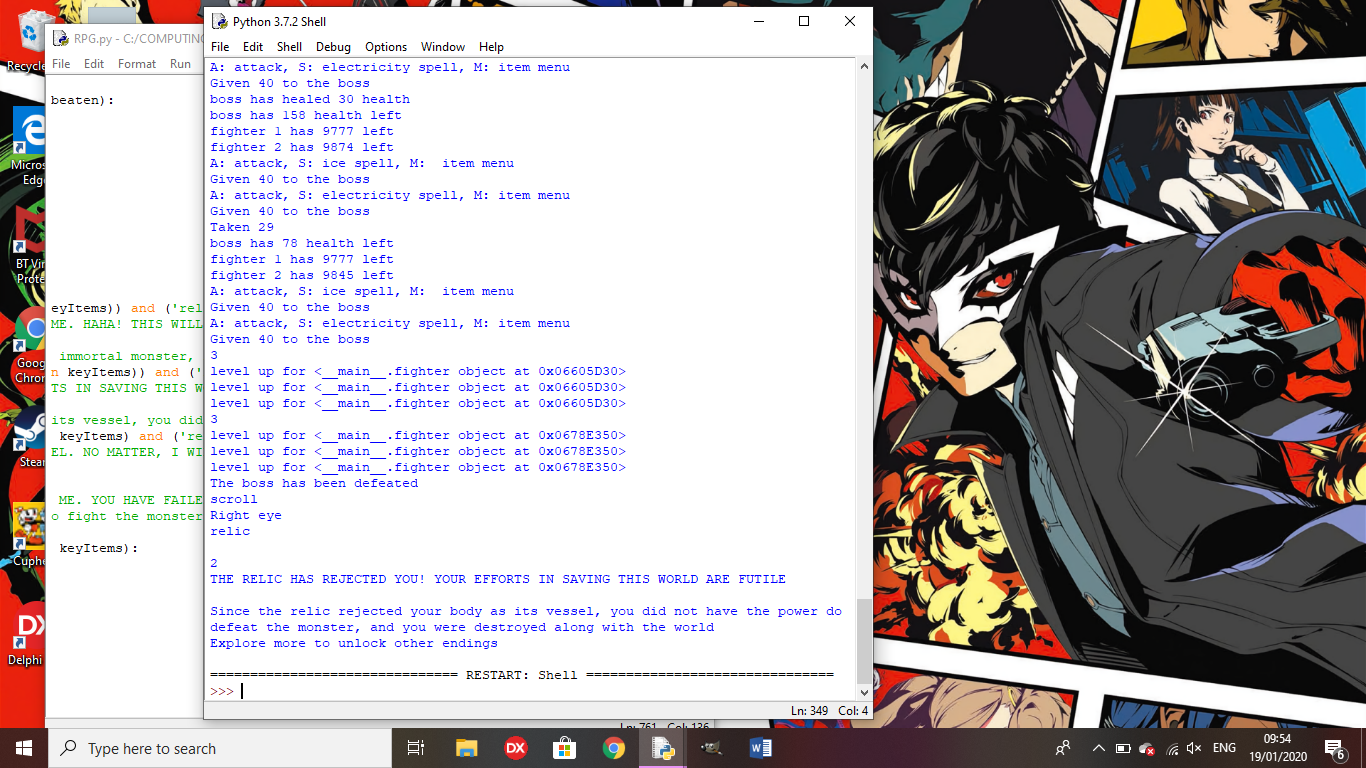
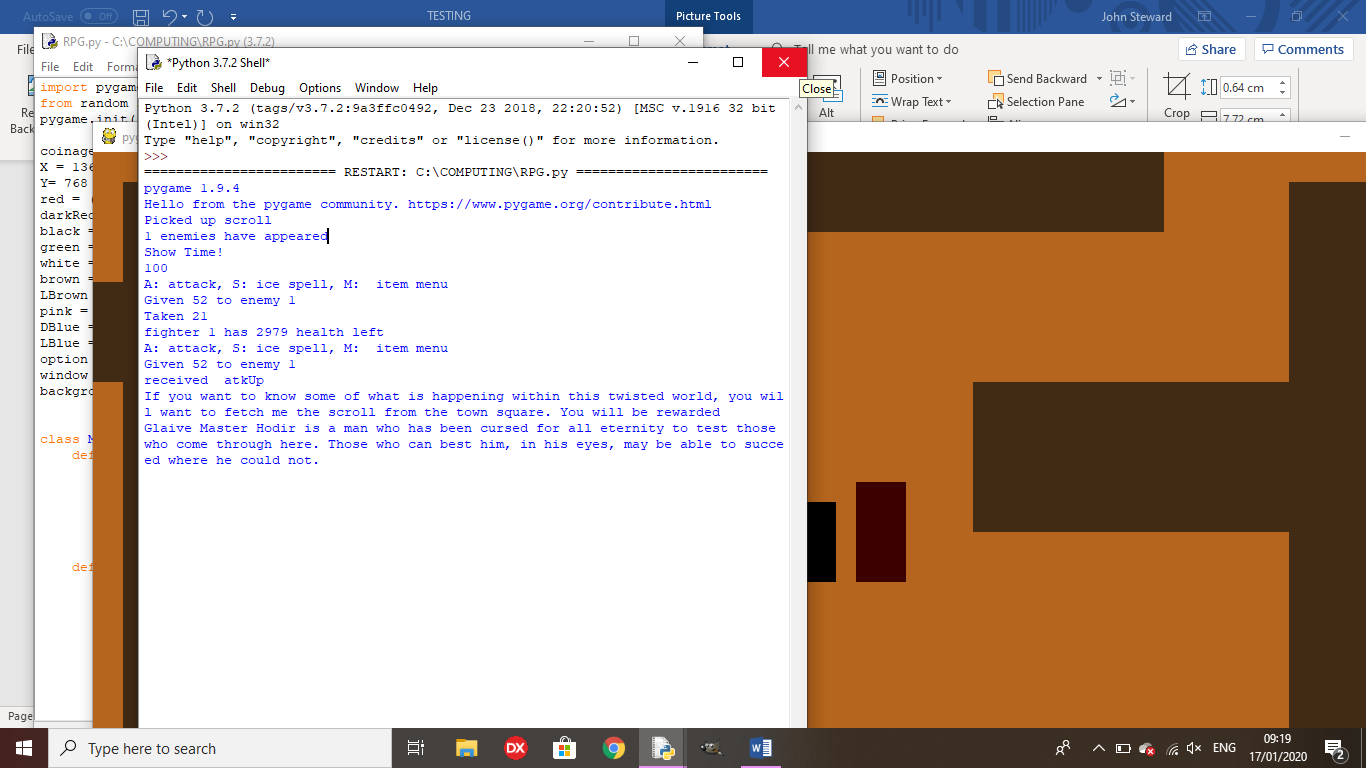
**Usability testing**

Test 1: Is the movement of the character fluid and quick?

Outcome: The player has to press the arrow keys each time they want to move, so they cannot hold the button down. This is because, if I build the code so the player can hold the button down, the game crashes without an error in the code, making it so the player cannot do anything. In order to make it more fluid, I will need to find a way for the player to be able to hold down the arrow keys to move without crashing the game.

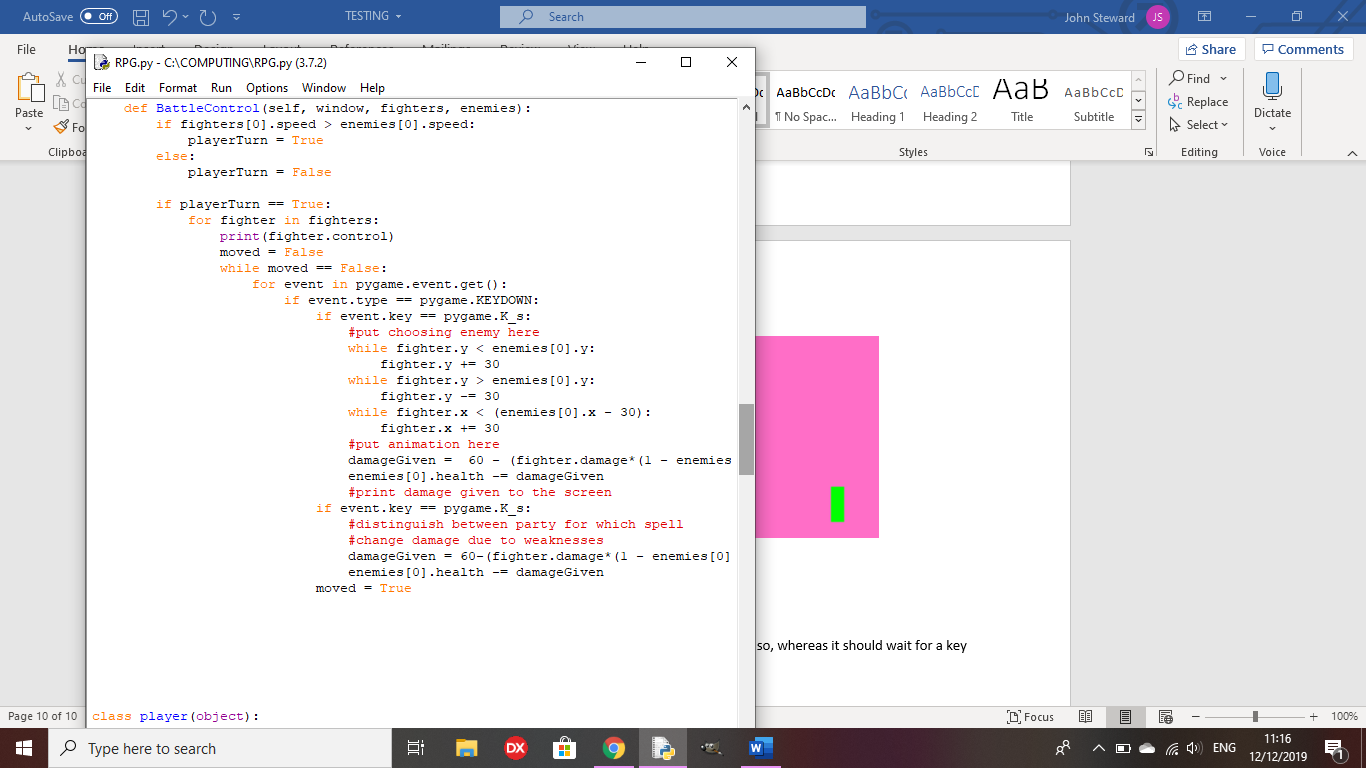
Test 2: Are dialogue and battle controls presented in an easy-to-read way?

Outcome: The dialogue and battle controls are shown on the python shell, making it easy for the player to read as all dialogue is in the same place so there is no need to look for other places to read dialogue.



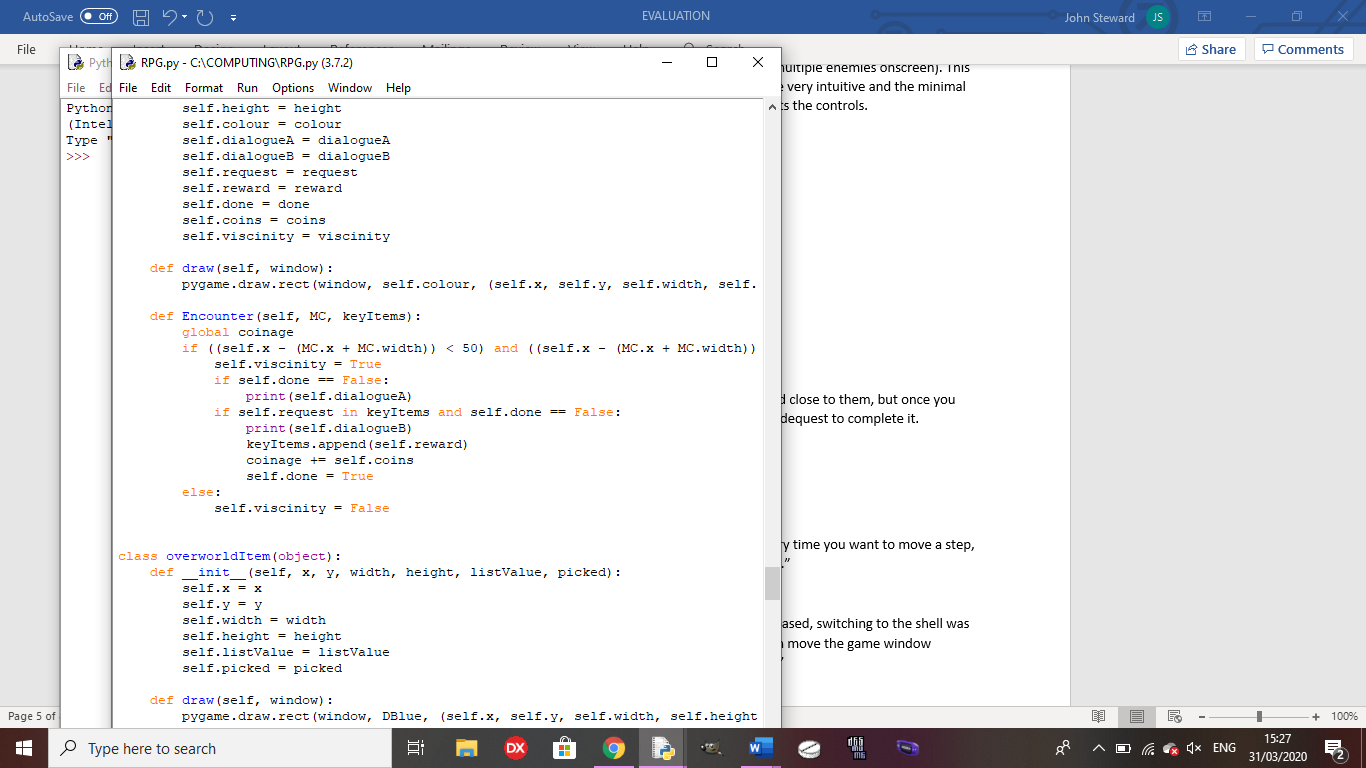
Test 3: Are the battle controls easy to understand and use?

Outcome: The controls are easy to use as they only require a single key press to choose the attack, and another press to choose the enemy to attack (only if there are multiple enemies onscreen). This means there is no requirement for a tutorial as all of the controls are very intuitive and the minimal controls are shown onscreen for if the player gets confused or forgets the controls.



Test 4: Are NPC sidequests easily accessed?

Outcome: You can only talk to an NPC if you are on their left side and close to them, but once you hae talked to them, it is easy to know what you need to do for the sidequest to complete it.



**Stakeholder Testing**

How easy was traversal?

“The traversal was a little annoying since you had to press a key every time you want to move a step, so I would want to be able to hold a key down to move continuously.”

How did you find switching from the game to the shell for dialogue?

“Given the relative lack of details in the game due to it being block-based, switching to the shell was not an issue as you could switch between them quickly since you can move the game window slightly to the side and can see part of the shell before clicking on it.”

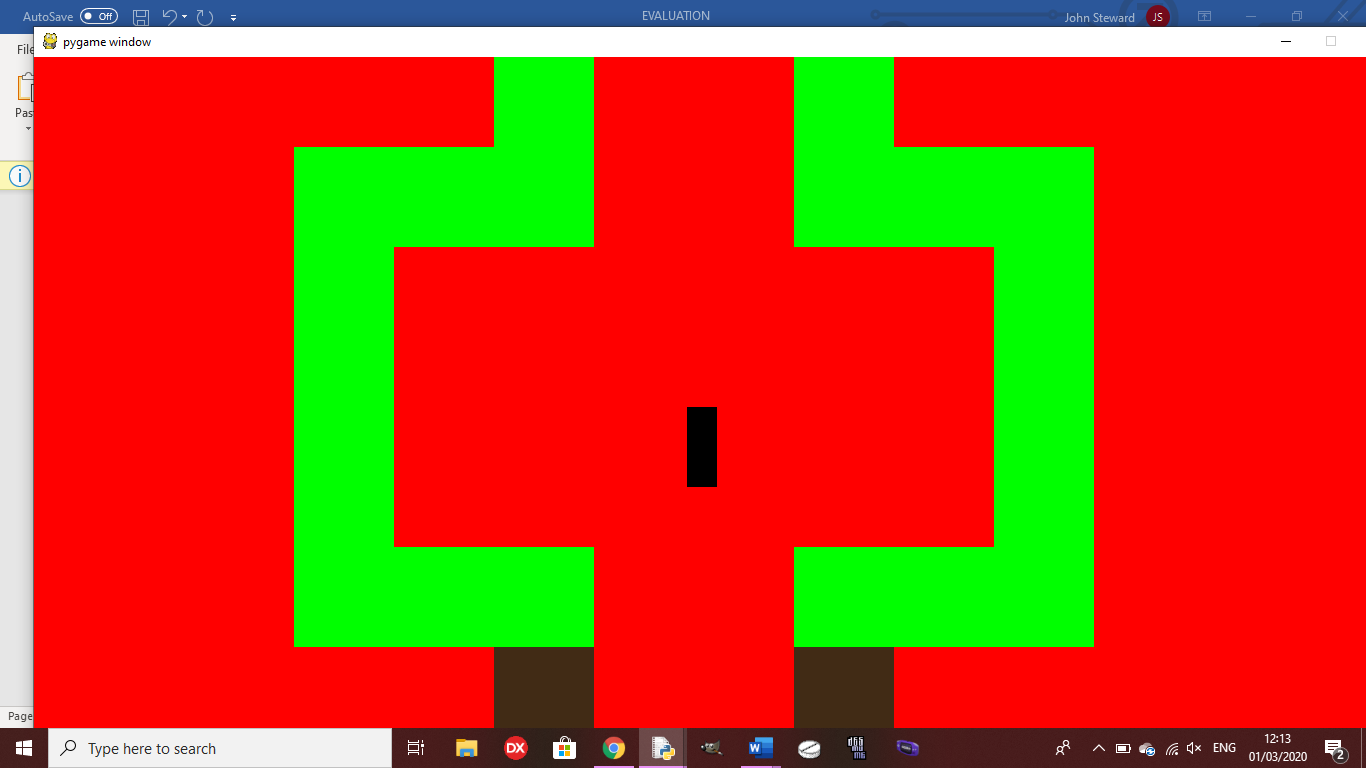
How easy were the battle controls to follow?

“Since the controls were given to you onscreen and are all a single key press, it was very easy to control the characters in-battle.”

How easy was it to access sidequests?

“They were easy to access, but it is annoying that you can only talk to them when standing on their left. Later, it should be altered so that you can talk to them from any side”

**Success Criteria evaluation from analysis**

1. ** Create all character and background sprites to be implemented into the game for the character to traverse the world while always being able to see where they can go. Each sprite for characters will be unique, and each area will be unique:** This criteria was not well met since, due to time constraints, I was unable to create sprites and animations for any characters or backgrounds. However, each character is a different colour from any other, and the areas are distinct as they have different background colours, walls and they have different side quests. For example, the first area has a fetch quest, the second area has a slot machine and the third area has a mini boss.



1. **Create a battle system to be used on the same window with random encounters:** This criteria was mostly met as each step the player takes has a random chance to trigger a battle, with a random number of 1 to 3 enemies (see test 7), the player can amass up to 4 party members throughout the game to aid them in battle. Also, players and enemies are able to buff themselves during the battle. However, the enemies do not have weaknesses or resistances to any spells.
2. **There will be optional content to reward the player for exploring well, like an optional area based on Bloodborne with a boss fight and good opportunities to gain experience and items:** This criteria was well met since the Bloodborne area is only required for the true ending and, to access it, the player must win the slots and find a path in the second area and it has 3 boss fights and a mini boss, with items to contribute to the true ending.(See test 23 in testing for development)
3. **Create multiple endings based on whether or not the player gained the necessary items throughout their playthrough to encounter the true final boss:** This criteria was well met as there are 4 endings based on how many of the key items the player received during their playthrough, and how many characters they have in their party. (See test 10)
4. **Create 2 sidequests in each area for the player to get invested into, if they want a break from the main story to do new things with interesting characters, fleshing out the world more and making each area have more life to it:** This criteria was not fully met, as there is only one side quest in each area as opposed to the two mentioned in the success criteria. However, these side quests are varied as one is a fetch quest, one is a slot machine and the last is a mini boss.
5. **Create a NG+ feature to add to the replayability of the game, making enemy and boss encounters harder, but the player keeps the equipment that they built up over the course of the previous playthrough. It will also have new bosses to encounter and 2 extra sidequests to incentivise the player to do another playthrough, keeping the player interested in the game for longer:** This criteria was not met as, once the game has been completed, it is done so the player will only be able to go back to their last save and do side quests to get other endings if they haven’t already.
6. **Create a save system that stores the state of all variables into a separate text file to be called to when the program is re-run to save the state of those variables back into the program, so the player can start exactly back from when they saved, unless they choose to start a new game, in which case, the values called will be the default values for each variable:** This criteria was very well met as, at any point in the game, they player is able to save the game to a text file and, if the game is reloaded and the player picks ‘Continue Game’, they will return to exactly the same place as they saved and, if they pick ‘New Game’, the player starts at the beginning. (See test 11)
7. **There will be 4 different areas for the player to explore throughout the story, 1 being optional, with 2 exclusive enemies for each area, and some that appear in multiple areas:** This criteria was somewhat met as there are 4 areas with one being optional and one to house the final boss, but there is only one enemy type in the game.
8. **The game will have a balanced difficulty, so it won’t be too easy to plough through, but it won’t be so hard that it is impossible to get through without grinding levels for hours:** This criteria was well met as the boss fights require strategy and good item usage, and there is no necessity to grind levels for a long time to be able to beat them. If you grow a few levels throughout the area, you should be able to beat each boss.
9. **The program needs to be efficient, taking up as little space as possible, with no load times as the world will be fully interconnected to keep the player as immersed as possible:** This criteria was well met as the full program is object-oriented, with reused functions as methods inside the classes, making it efficient, and the only load time is to load the title screen. Everything else, from loading the backgrounds to battles, is seamless.

**Limitations and future development**

The most important change for future for me would be making sprites and animations to make the game more visually appealing to hook new players into playing it. As it is currently, the game is not very visually appealing as it is only made up of coloured blocks, so adding unique sprites and animations would make it a lot easier for the player to get invested. I would also add in NG+ to add more replayability to the game (Shown in my stakeholder questionnaire) and allow the player to get more endings without reloading a previous save. It would also add extra difficulty to the game and a programmer would not find it hard to create new sidequests and potentially a new ending since everything is object-oriented so they could just add new objects for NPCs and items. They would also find it relatively easy to add new enemy types as the enemies are a class so they could make objects for the enemies and add them to the existing enemy list.

To add further enjoyment to the game, in the future, I will add more enemy types into the game with harder enemies in later areas. It won’t be too hard to implement since the enemies are their own class so I can just make more instances of the class with higher stats for later areas.

The only criteria that a programmer may need some time to meet is adding enemy resistances and weaknesses as they would need to add a new set of variables and alter the battles to either increase or decrease damage taken by the enemies based on the spell they were hit by.

Since the only image file is for the title screen, as long as another computer has Pygame installed, another person is able to run this game quite easily on their computer, as long as they download the source code and the image file for the title screen.

**Usability Evaluation**

According to my stakeholder, the movement in the game was annoying due to having to press the arrow key to move each step. In the future, I would need to find a way to allow the player to hold down a key to move continuously without causing the game to crash to make it easier for the player to play the game without getting frustrated.

It would also make it a better experience for the player if all the dialogue and damage values in battle were displayed on the game window, so the player doesn’t need to keep switching from the game window to the shell.