Level 2 - DEVELOP AN OUTCOME

AS91896

Heeju Shim

**Project details:**

Project Title: Quizpy

Github: <https://github.com/boomt1337/quiz-py>

Trello: <https://trello.com/b/45cEEQMR/py>

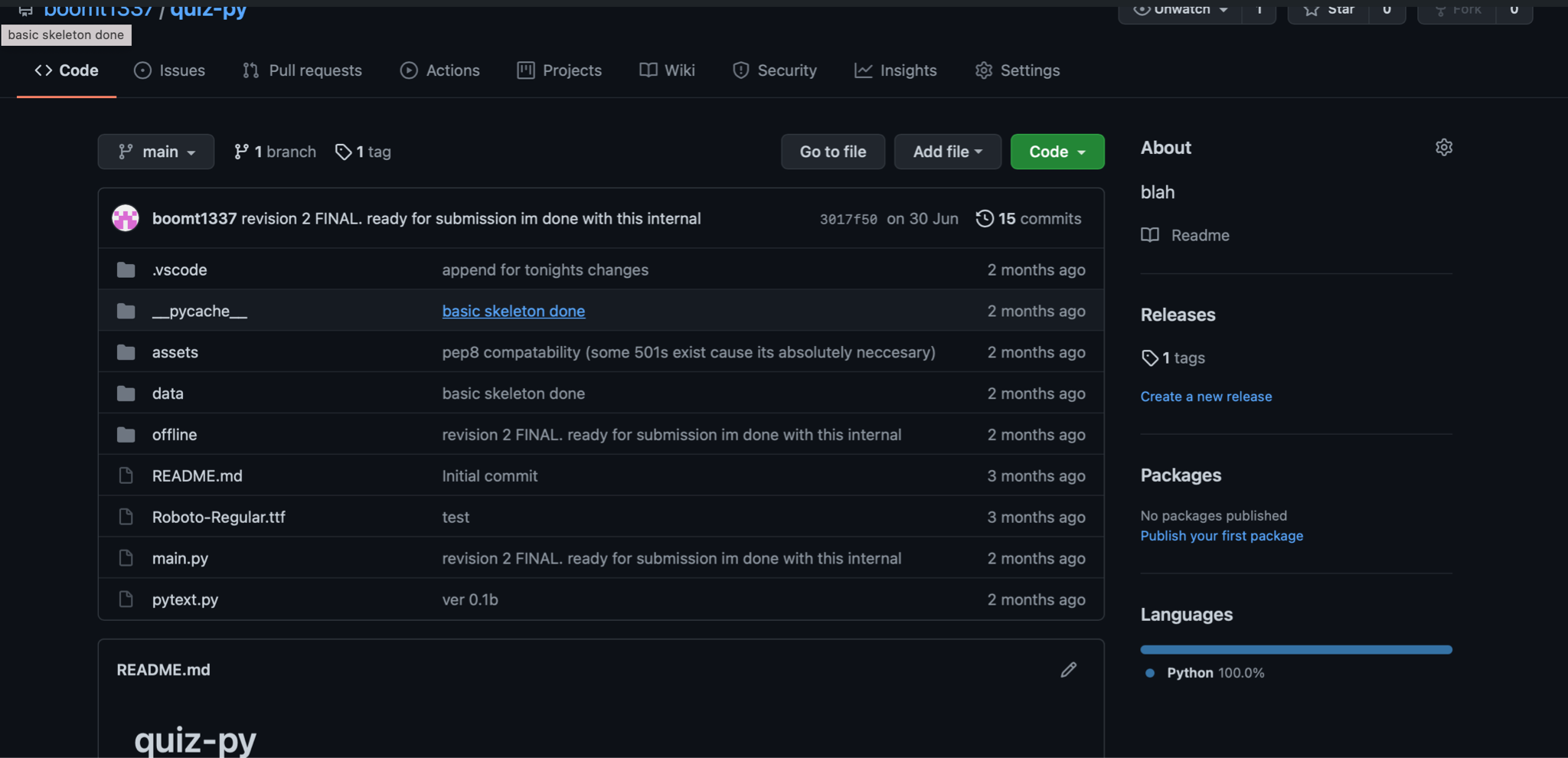
Project Management: Setup

* **Record evidence that you have set up project management tools to manage the development of your outcome.**
* **Record evidence that you have chosen project management techniques to manage the development of your outcome.**
* **Record evidence that you have set up/ chosen version control tools/techniques to manage the development of your outcome.**
* **Explain why you believe that these tools and techniques are appropriate for the development of your outcome.**

**Version Control tool:**

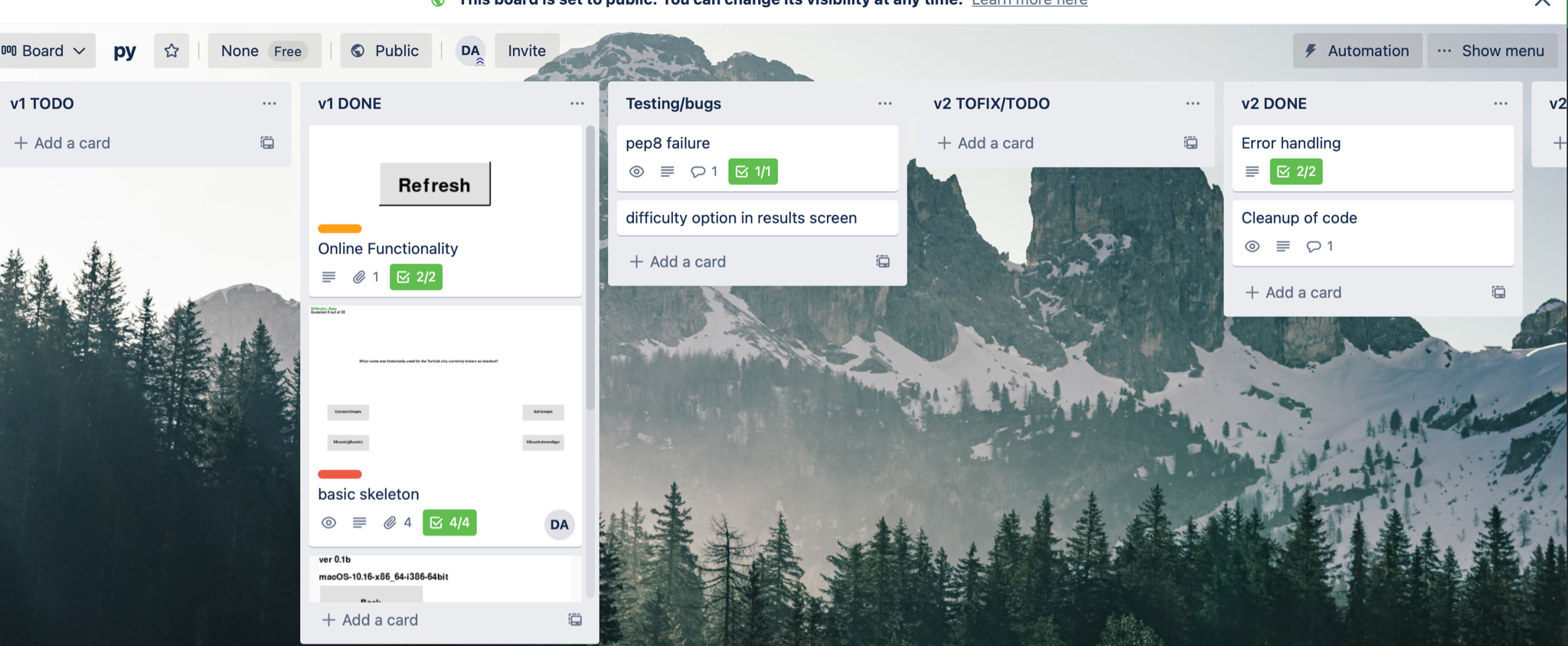
Tool: Github

<https://github.com/boomt1337/quiz-py>



**Project Management Tools**

Tool: <https://trello.com/b/45cEEQMR/py>



**Project Management Technique:**

I used Work Breakdown Structure (WBS) for the project. This was shown with the screenshot above. V1 (the working copy) was broken down with cards into smaller functions, like Online Functionality, the skeleton, and the different screens and quiz engines.

**Justification:**

* **Version Control Tool**

I used Github for version control for my project. Github is a service that enables me to upload and store different versions of my code in the cloud, using a repository that I commit to at the conclusion of the day’s work.

I opted for Github because it is:

1. **Simple to use.**

For me, Github is a very simple software to use. I found that with Github, I didn’t have to set up anything major on my system to upload, nor did I have to find that I had to set up my own servers to host my file compared to other software currently on the market. Therefore, this is essential to my project as it allows me to spend less time setting up.

1. **Open-source and free.**

Having a free host to host my revisions over time is appropriate for my project because I don’t need to worry about website hosting or tinkering with other alternatives, which allows me to save time planning out and setting up the version control tool.

1. **Has integration with my IDE (Visual Studio Code).**

With Github and my IDE, it allows me to easily upload my work with comments with a simple click of a button. This simplicity is what others such as CVS, Apache SVN and Google Drive don't bring. Therefore, this is appropriate to my project as it will help me save time in setting up for my project AND keeping revisions to debug and submit for feedback.

* **Project Management Tool**

For my Project Management Tool, I used Trello. Trello is a Project Management software that allows me to make cards in different categories of my choosing.

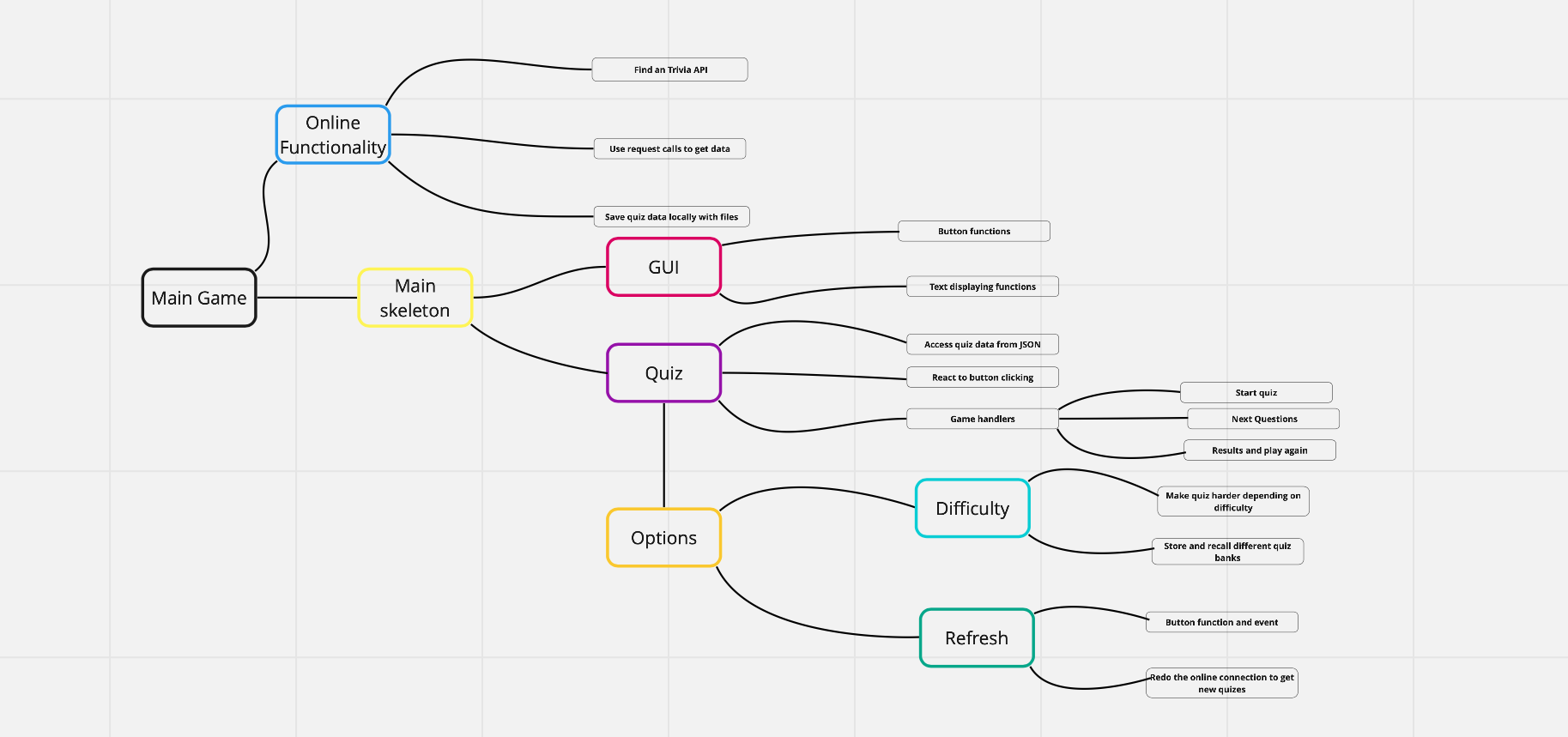
I used Trello because it allows me to break down my project into separate achievement criteria which allows me to plan what to do for each day’s work. Also, it gives me the option to put descriptions and checklists for those criteria, which I feel was important for my project as I can reassure myself more that I have really met the goals I set for myself.

I considered other project management tools like Asana, Monday.com and Miro, but I ended up opting for Trello because I felt that Trello has an more powerful suite, such as “cards”, checklists and the creation of separate categories for my two versions (v1 and v2), and just the simplicity of its User Interface that those software simply lacked.

* **Project Management Techniques**

As stated before, I used WBS (Work Breakdown Structure) for my project. This allows me to break down the full game into separate parts, such as the Online functionalities, the title screen and the game functions itself. Having this technique was appropriate for my work given its sheer size to complete, which gave me a plan in the timeframe that I had to slowly build up and allowed me to test more frequently.

Decompressing

****

**Flowchart**

Trialing Outcome

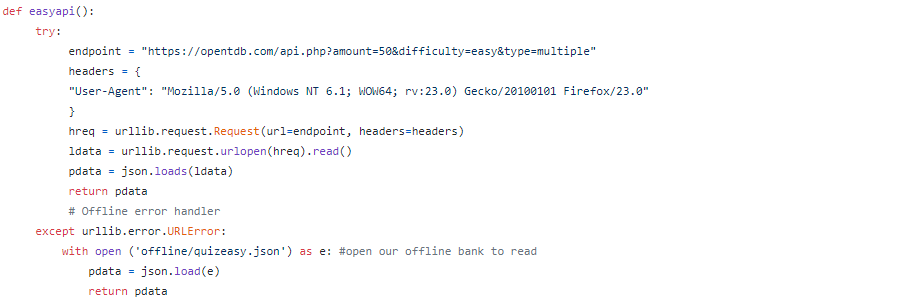
**Comparing alternatives/improvements**

* **Online functions**

****

Initially I was going to check if the machine was online by using a separate function that pings Google and returns true/false . True being online, false being offline. However, after trailing this it didn’t work, as I didn’t know how to incorporate it into my API calling functions, and I thought that continuing to use this component may harbor errors.

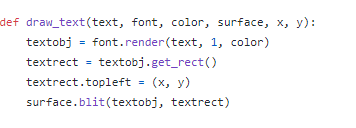
In the end, I incorporated error handling into the function itself, instead of doing it separately.



I used a try statement with an exception (except urllib.error.URLerror), which reads the offline bank. I decided to have the offline bank static, meaning that it will stay the same. I made this change because of the simplicity of the code base, and to reduce the overall complexity of the codebase, which ties into my sustainability + future proofing implication goal.

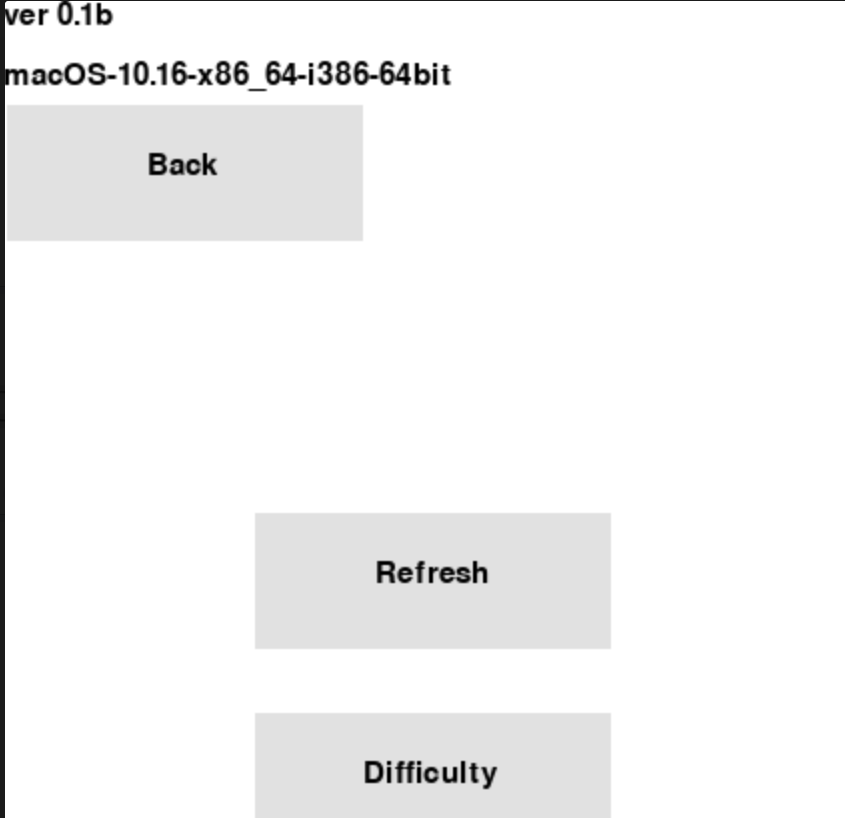
* **GUI - Text**

****

****

For text, I was originally going to use pytext, an external library that can print text onto the screen, however the trailing for this component revealed that it wasn’t printing the way I wanted it to be, often printing in awkward positions or displaying blanks.

In the end, I opted to use a function to handle text printing on screen. I chose this because it is more sophisticated and precise with x,y coordinate support, which improves functionality and usability.

Output: 

**Stakeholder Feedback**

**Version 0.1**

| **Stakeholder** | **Feedback** | **Improvements made** |
| --- | --- | --- |
| **1** | * The GUI is good but make sure to make the quiz playable | No changes were made as the game wasn’t playable when the program was distributed, however the next testing session will have a playable demo |
| **2** | * There should be music that plays during the game | For the next testing session, I will add backing tracks throughout the program based on this stakeholder’s feedback. |
| **3** | * I don’t like the black gradient of the buttons, change it so that it is one color only | I will also change the button class to remove the black gradients too, so that the button is one color and uniform |

**Version 1**

| **Stakeholder** | **Feedback** | **Improvements made** |
| --- | --- | --- |
| **1** | There is an issue with printing the questions on screen, it doesn’t display special characters properly | The issue has been identified, and for the next feedback session there will be a proper check to remove these special characters |
| **2** | There should be backing music on the results screen | I will add a short results theme from online for the next feedback session |
| **3** | The button placements on the GUI looks misplaced, especially the play again and the title buttons on the results screen | For the next feedback session, I will make sure that I have readjusted and realigned the buttons used for the results screen |

**Version 2**

| **Stakeholder** | **Feedback** | **Improvements made** |
| --- | --- | --- |
| **1** | There should be an offline bank used for the quizzes if the game is offline, instead of defaulting to the quiz questions made | This wasn’t implemented fully yet by the time this was made, so no improvements is made here |
| **2** | Looks good | N/A |
| **3** | Looks good | N/A |

Testing Outcome

**Test 1 (Version 0.1)**

| **Component** | **Expected outcome** | **Actual outcome** | **Pass/Fail** | **Fixes to make** |
| --- | --- | --- | --- | --- |
| Online function/Initialisation | Components should be able to initialise if files are missing, and create a json file. Should be able to default to offline bank if offline | Component makes json files, but errors if file already exists which means opening the game twice would crash | FAIL | Add checks to ensure game does not crash if an quiz bank is already created |
| UI - Buttons | Buttons should react | Buttons react | PASS | N/A |
| UI- Text | Text should be able to display on screen | Text shows on screen | PASS | N/A |
| Quiz - reading questions/answers | Able to read the files for the quiz folder the online functions have made before | Reads quiz folders/files | PASS | N/A |
| Quiz - Results and game functions | Result screen should function | It is blank | FAIL | Add in functions and code to handle result screen and grades (progress the game further) |
| PEP8 | Should be almost errorless | 60+ errors | FAIL | Fix this in Version 2 |

**Test 2 (Version 1)**

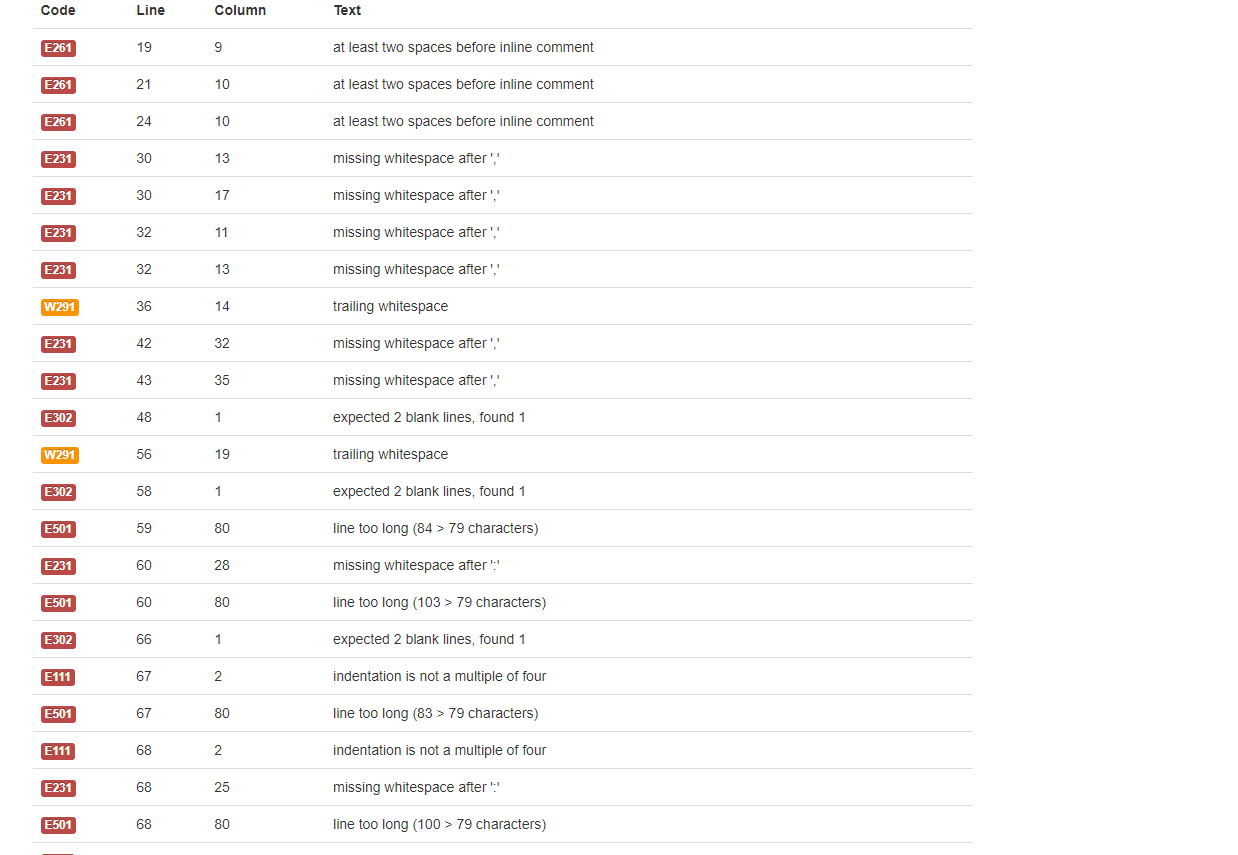
| **Component** | **Expected outcome** | **Actual outcome** | **Pass/Fail** | **Fixes to make** |
| --- | --- | --- | --- | --- |
| Online function/Initialisation | Components should be able to initialise if files are missing, and create a json file. Should be able to default to offline bank if offline | Component makes json files but errors out if the game is offline, unplayable if first launch if offline | FAIL | Add an contingency into the API functions to prevent game bricking/crashing with no internet |
| UI - Buttons | Buttons should react | Buttons react | PASS | N/A |
| UI- Text | Text should be able to display on screen | Text shows on screen | PASS | N/A |
| Quiz - reading questions/answers | Able to read the files for the quiz folder the online functions have made before | Reads quiz folders/files, but displays garbage on screen from quotes and special characters | PARTIAL PASS | Add detections to prevent garbage from printing on screen |
| Quiz - Results and game functions | Result screen should function, buttons for retrying game should function too | Game detects when game is over, displays result screen  Options are broken however | PASS | Fix the refresh/difficulty buttons on the option screen |
| PEP8 | Should be almost errorless | 50+ errors when checking | FAIL | Most of these are warnings so make sure to fix those at version 2 |

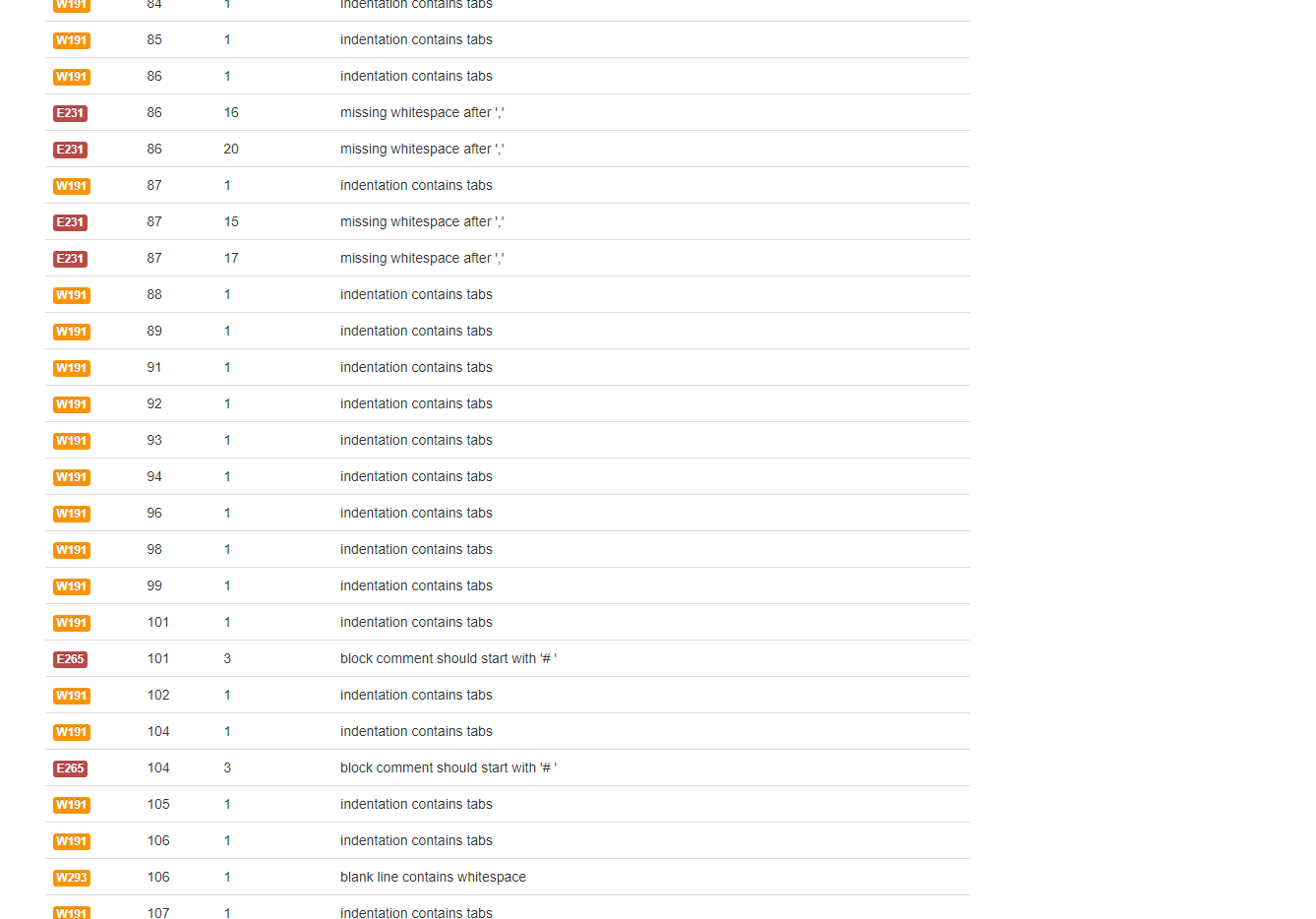
**Test 3 (Version 2)**

| **Component** | **Expected outcome** | **Actual outcome** | **Pass/Fail** | **Fixes to make** |
| --- | --- | --- | --- | --- |
| Online function/Initialisation | Components should be able to initialise if files are missing, and create a json file. Should be able to default to offline bank if offline | Component makes json files, reacts to offline situations and defaults to offline json bank | PASS | N/A |
| UI - Buttons | Buttons should react | Buttons react | PASS | N/A |
| UI- Text | Text should be able to display on screen | Text shows on screen | PASS | N/A |
| Quiz - reading questions/answers | Able to read the files for the quiz folder the online functions have made before | Reads quiz folders/files | PASS | N/A |
| Quiz - Results and game functions | Result screen should function, buttons for retrying game should function too | Game detects when game is over, displays result screen, successfully retries with clean slate | PASS | N/A |
| PEP8 | Should be almost errorless | 5 errors but those are links that can’t be spliced | PASS | N/A |

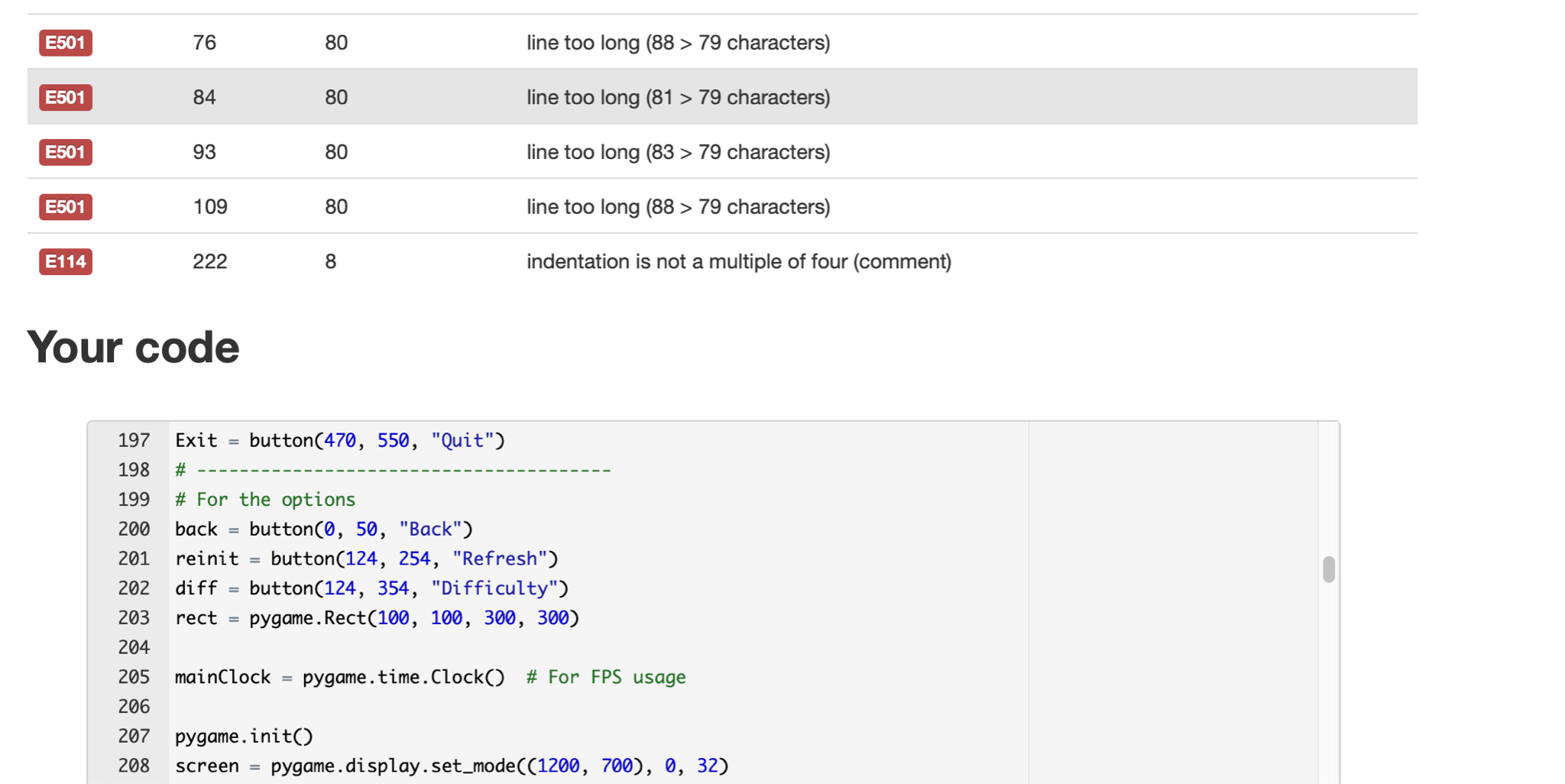
**Code Validation**

**Version 1:**

****

****

**Version 2:**

****

From Version 1 to Version 2, the amount of code validation errors have dropped significantly. I understand that there are errors on version 2, but fixing those broke my API calls or some essential line, so I decided to not fix it.

**Final Testing**

**Version 1**

Windows:

<https://streamable.com/3o8217>

Mac:

[**https://streamable.com/451sy6**](https://streamable.com/451sy6)

**Version 2**

Windows:

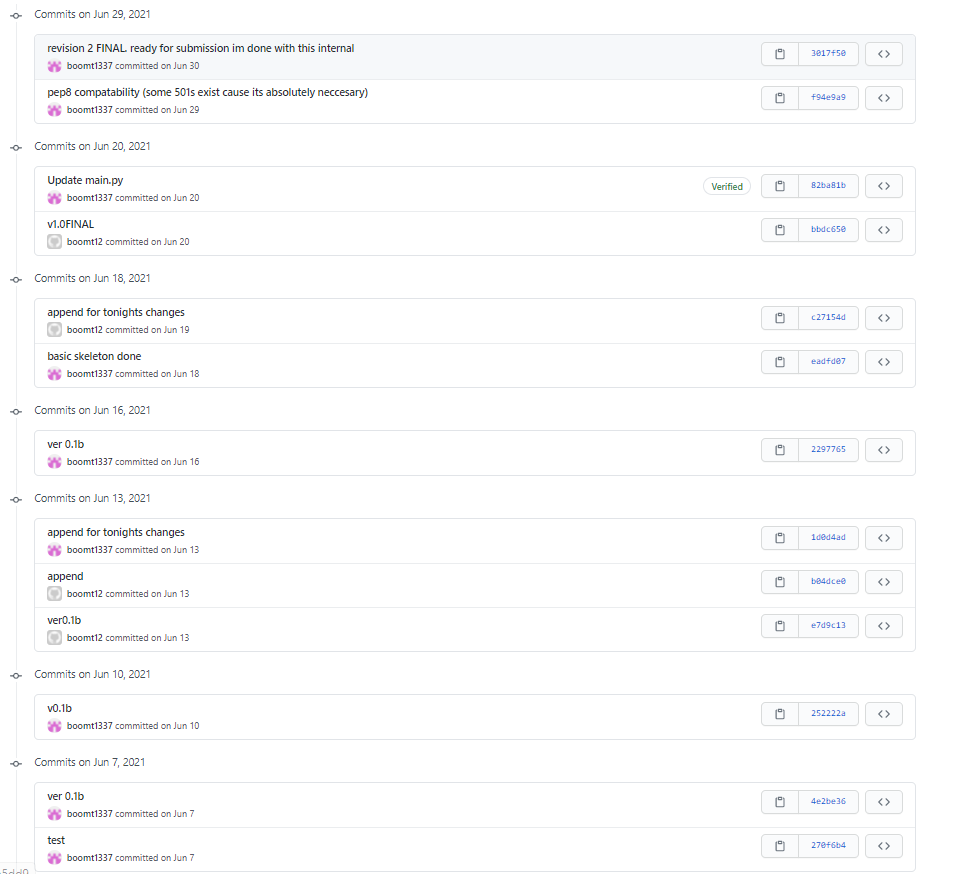
<https://streamable.com/im1qvi>

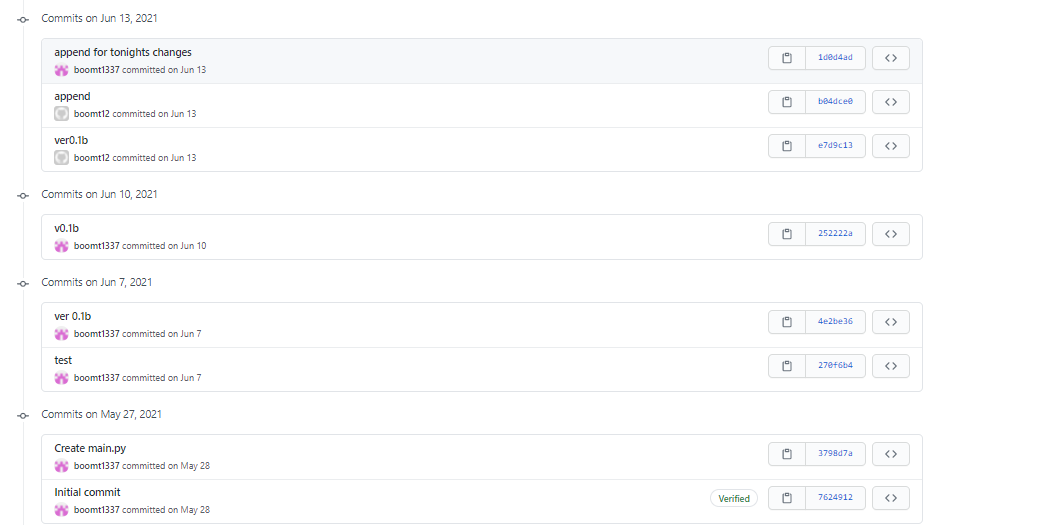
Mac:

<https://streamable.com/2kj7fw>

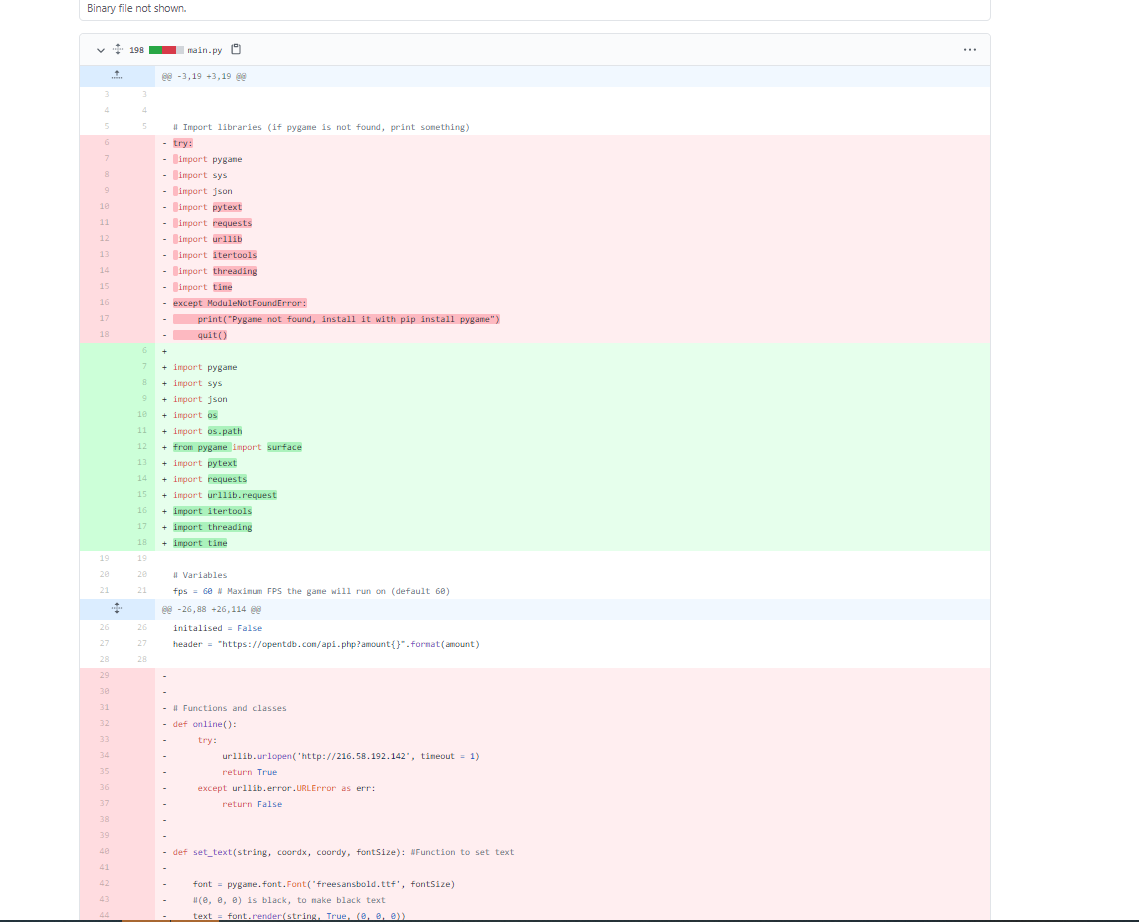
Version Control

Github was used, and I kept track of versions with codenames and comments (version 0.1b, version 1 and version 2)





Example of what each version looked like





Relevant Implications

**Legal**

Description of implication:

Programs should abide by the relevant laws of countries worldwide. This includes all copyright laws, privacy laws (GDPR, Privacy Act, COPPA) and Digital Communication laws.

How I addressed this in my project:

For my project, I addressed legal implications by making sure anything I use is free of use. This means assets like music, images and libraries used were under MIT or Creative Commons licences. I also made sure my project abided by digital communication and privacy laws by making sure I don’t make false allegations against someone else in my program AND making sure I don’t store any information which may violate privacy laws.

**Usability**

Description of implication:

Usability or “user-friendliness” is defined as the ease of using a product. With Usability, all elements within the program must be considered to ensure that the user can easily use the program.

How I addressed this in my project:

For my project, I addressed usability in my project by considering elements in my program, such as user interactions and the customisation to be as easy to use as possible. For example, I made user interaction as usable as possible by incorporating buttons instead of other means like typing inputs.

**Sustainability + Future proofing**

Description of implication:

Sustainability and future proofing is defined as the capability of a software to be easily understood, so that other people can improve or update your software later on.

How I addressed this in my project:

In my project, I addressed this implication by making sure different parts of my work are easily arranged by priority (initialisation to title to game to results). I also made sure to comment my code, particularly around the api links, as I know this can become redundant in the future.

**Accessibility**

Description of implication:

Accessibility refers to the amount of people who are able to access the project. Many users worldwide suffer from conditions like colorblindness or rely on braille or assisted hearing which may otherwise hamper their ability to use the program.

How I addressed this in my project:

For my project, I addressed this by making sure I design my project in a way that would help these people. My project is GUI based, so I made sure to make large buttons that people with motor disabilities can easily navigate to. I also made sure text is displayed instead of spoken, which would help people who are hard of hearing to understand for example the quiz’s question and answers.

Evaluation

Information from planning, testing and trailing has assisted me in making a high-quality outcome because I think having a plan using Trello and the Work Breakdown Structure (WBS) planning method helped me plan out my time across the amount of time that I have had, by simplifying an overall goal into smaller manageable timeframes and aspects of the program were managed. Using other planning methods may have led to a lower quality outcome because WBS is the planning method that I am very accustomed to and if I were to do this again I would use the same method that I used in this project.

Information from testing and trailing my project has also assisted me in making a high-quality outcome. I feel testing has helped me work better as it allowed me to catch some errors that may have led to a lower quality of work, such as pep8 validation or the functionality of the code, as that was a big leap to hurdle. I think developing as well as testing regularly has allowed me to save time worrying about code breaking later, and this is a habit I will definitely not change if I did this again.

Trailing my project, by incorporating alternatives and stakeholder feedback has helped me gain a sense of what other people found hard to navigate, or their own suggestions or concerns, which allowed me to refine my project further. However, I have only done 3 stakeholder feedback for this project. If I were to do it again, I think I would expand how many stakeholders I have in the future, which would allow me to make my program more refined than it is currently, as I didn’t need to refine it much in the final outcome.

Generally, if I were to do this project again, I would make the GUI more visually appealing, as I feel like I didn’t do the best job of blending simplicity with aesthetics. Aesthetics was going to be an implication that I would address, however I lacked the knowledge to follow through.