**SDD HSC major work Documentation**

**Problem statement**

I realised that during every gamer experience that at one point in time they will face gamer

block, a phase where there is no good game out or don’t know what to play. The solution GamersGateway, using flask and visual studio code I will create a web app that will randomly select a game from a list of thousands on steam using request to API that returns the appid and name, then it will go and use the steampowered api to send a request out for the information and return it in JSON format. Additionally the app will use a web scraper that will search on bing for the image relating to the game and then return the first image result that pops up The Applications features will involve the user generating a game at random or if you have a specified genre, you wish would play the app will allow the user to enter an input of the specified genre they want to play. The information about the game being returned will consist of an image the name and developer, trailer, description of what it is and price. However, there are some limitations due to time constraints like I was planning to add a feature of saving the games they like but with time it was too difficult to implement.

**Legal and Ethical issues**

**Legal**

There are two main legal issues within the app GamersGateway are the use of intellectual property such as copyright law, and when planning to monetize the app. The use of intellectual property is to tie pieces of software to a person and stop it from being copyrighted, in my app the game description, trailer, or images might be copyrighted. And if I display these assets it could infringe on the rights of the content owner, an example in my app is that using steampowered API has links to videos and images relating to the specified game. Where developers may have copyrighted the file and only given access to steam. Additionally, if I plan to monazite my app and make advertising, I will have to make sure that the owner of the external data accepts the use of data on commercials and that the APIs and content providers complies with the commercial use policies.

**Ethical**

There are 2 main ethical issues within the app GamersGateway and that is firstly an algorithmic bias opinion, secondly Content moderation through relevant information. Algorithmic bias is a key ethical issue as within the app as a developer I could show favouritism towards specific game developers, meaning the idea of randomly generating games would be tarnished through prompting more frequent displays of certain developers. Furthermore, the use of content moderation as the API being sent data request could contain false information about the developer or game to deter the user from buying the game, which would ruin the reliability of the app and potentially lose user interest.

**Functional and non-function requirements**

|  |  |
| --- | --- |
| Functional | Non-functional |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Algorithmic description**

Fetch\_Game\_Trailer() function:

1: creates variable trailers to store the specified area movies within the data dictionary of the random game

2: uses a for loop to request all the mp4 or webm URL for the trailer

Development Log entry

|  |
| --- |
| **Development entry 1** |
| Date 9th May |
| Week 2 |
|  |
| **Summary of work done;** I have found an achievable idea of solving gamer block through GamersGateway. What has been achieved is that I was able to figure out how I was going to find the games through an API called Steam spy which enables me to run line of code to randomly select a game. |
| **Challenges And Solutions:** The challenges I faced was that when I found the problem I was confused on how I was going to find APIs for the game but that was easily solved through steamspy API that took a couple days to find. Additionally, the way to return the text I didn’t know what JSON was or request so with YouTube videos I learnt how the request function work and how JSON works. Lastly another challenge is that I was originally going to import the API and use its functions, but I found that using the website URL I can just model my code to return the information myself and it helps through not being limited to the Steam spy API functions. It also enables the information to be passed around the code more easily as some functions might not interlink between python and cause future errors |
| **Milestones achieved:** found the API that will assist in getting the valuable information related to the app that will be persistently used in the future as it is easily usable and manoeuvrable |
|  |
| **Development entry 2** |
| Date 15th May |
| Week 3 |
|  |
| **Summary of work done:** Setting up a website template using Flask that in future will display all of the information and functions that GamersGateway will show to the user, such as search the specific game you want, search through genres or just clicking a button and returning a random game. |
| **Challenges and solutions:** A major challenge that I had is that I didn’t know how to code using HTML as I hadn’t of used it in a long time which ment I had to go and watch videos on how HTML works. Furthermore, as I learnt what HTML does, I didn’t know how to interpret it to my code so I kept getting errors as I couldn’t get the buttons to work. The solution was that I never created a route in my main code which is key as when the button is pressed it needs a function in the main file to call to as any accessibility features are to be defined within it. |
| **Milestones Achieved:** I have now been able to start displaying the information regarding the Games been generated which was key as I now have learnt how to code in HTML making development later on a lot more easier when using HTML within the project. |
|  |
|  |
|  |
|  |