Interactive Media Teamwork

Managing Projects with GitHub

Thomas Abela

2HNDi

Task 1 – P1.1 – Identify Client Requirements

This project follows the development of my Unity computer game created for the CIDP assignment. From the beginning I already had an idea of what kind of genre of game I wanted to make, and so the functionalities I determined for the game were based on that decision. I wanted the game to be a sort of 3D platformer where you control a character that can explore a virtual world. These functionalities must reflect that genre. The following requirements were determined:

The game must have -

1. A well animated character with precise movement controls.
2. A relatively large, open area of terrain for the player to explore.
3. A time-based objective. The player must complete a task before a certain amount of time expires.
4. Different collectibles spread throughout each level.
5. A smooth third person camera which allows the player to zoom in/out of the character and rotate the camera around the character.
6. A Health Bar with various collectibles affecting its value (Positive or negative).
7. While the game is intended to be played with a keyboard and mouse, an alternative set of controls should be implemented for mobile. For this reason, the option of on screen UI should be included to accommodate touch controls.
8. The main menu should reflect what kind of game this is. Therefore it should not be a static image but be based in a 3D environment with animated elements.
9. Level music is an important element to 3D platformers of the past such as Crash Bandicoot or Spyro the Dragon, which this game takes inspiration from. The game must have upbeat level music and sound effects to accommodate this requirement.
10. The game must have some convey an educational message, which has been determined to be Recycling awareness.

Task 2 – P1.2 – Define and Analyze target group to identify user needs.

In order to familiarize ourselves with GitHub, I will define the following terms in relation to GitHub tools.

**Repository**

A place where your code is stored. Each project you create must be contained within a repository. To create one, go to your GitHub account and go to the repositories tab and click New. Repositories can be public or private, however free accounts are only allowed to create public repositories, meaning anyone is able to find and access the code. So make sure your projects do not contain any sensitive information.

**Commit**

Committing is the act of taking all the changes you have made to a project and then recording them. This can be done as many times as you like, after which you can then push those changes onto the GitHub server, where you project’s repository is stored. One of the benefits of tracking commits is that it allows you to roll back to a previous version of the project if necessary.

**Issue**

Issues can be questions, tasks or suggestions made by individuals relating to your repositories. Remember that if your repository is public then anyone can see your project, and they may have queries regarding certain areas of your project. Whenever an issue is created, it opens its own discussion forum where the repository collaborators are able to discuss the issue with the person/s who submitted it.

**Sync**

Sync is the act of making sure that your local branch matches your remote branch of the project. If your local branch contains commits that your remote branch has not incorporated, then it will push those changes onto the remote branch. Likewise, if the remote branch has additions that the local branch does not, it will pull those changes onto your local branch. Sync basically ensures that your local and remote branches mirror each other.

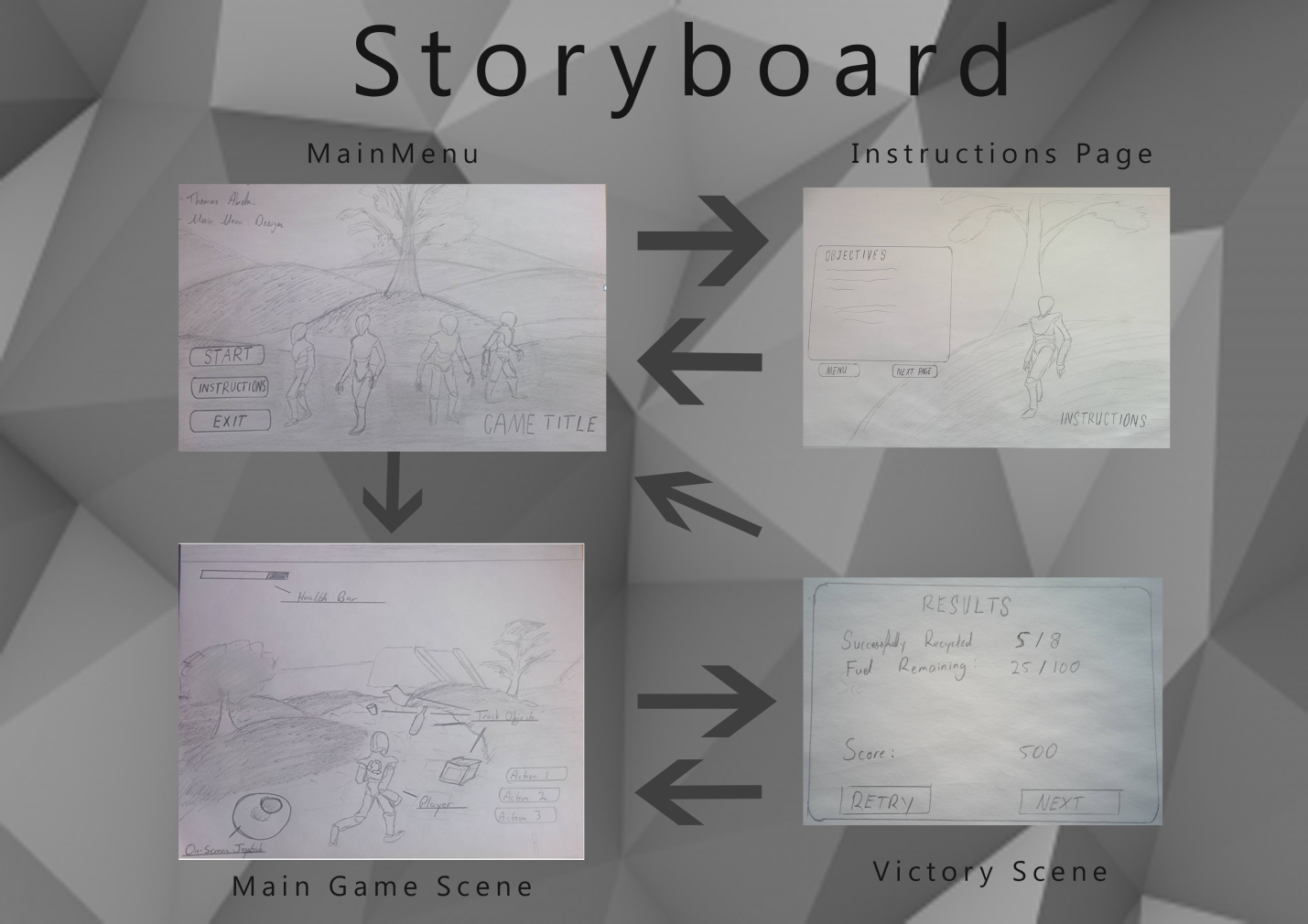
**Add**

Add is an option which allows users to add more buttons to a repository. Once this has been done, the user can commit the data and any changes to the remote branch via Sync.

**Pull Request**

Users can make changes to a project and then submit them to the repository collaborators as a proposed change to the repository. This is known as a pull request. The user who makes a change to the project can open a pull request, and the repository collaborators can then open a discussion forum to review the change. The collaborators can either accept or reject the change to the main repository.

Task 3 – P1.3 – Clarify creative intentions through recorded communication with client.



Task 4 – P2.1 – Identify and apply own area of expertise.

Task 5 – P2.2 – Clarify own role within team-driven development schedule.

Task 6 – P3.1 – Produce preliminary components for an initial prototype.

Task 7 – P3.2 – Evaluate and confirm prototype in relation to constraints.

Task 8 – P3.3 – Reflect and record on feedback from prototype phases.

Task 9 – P4.1 – Develop a fully working interactive media product that meets clients’ needs.