**GROUP PROJECT, GROUP 3**

**DATE: 14 November 2018**

**TIME: 13:50 – 17:30**

**ATTENDEES** Tom Gibbs, Henry Crofts

**LOCATION:** Discord, Voice Call

**Minute Taker: Tom Gibbs**

**Overall aims of the current sprint *(Detailed tasks, user stories and time allocations are tracked on JIRA)***

* Begin development of Unity prototype
* Create functionality for Unity project base classes
* Continue working in a Studio Environment

**Meeting:**

All team present. Henry advised Tom earlier in the morning that the studio-jam would have to be moved due to an urgent doctor’s appointment. Because of Henry’s commute time this would make a studio-jam unreasonably short. Meeting was moved to a discord voice call, following Henry’s appointment, so the team could continue the sprints Jam sessions.

Meeting start slightly delayed to account for Henry’s appointment.

Team began by discussing presentation feedback, received from tutors via email this morning.

Team began by reflecting on presentation feedback email (containing Dave’s feedback, additional brief feedback from Rob).

Email shows the group presented the concept in a suitable manner to allow for tutor understanding and shows tutors believe team to have successfully applied research design theories.

Tutors advise team to now develop gameplay and begin honing time variables – which team agree will be crucial to the player experience.

Dave provided key elements to precisely define:

* Precisely how long is a round?
* How many rounds will make up a typical play session?
* How often are tasks added to the list?
* How long do the players have to accomplish each task?
* How many tasks will be active at any one time?
* How do these variables change from the start to the end of a round?
* How far can the team push the players into madness and hilarious helplessness and panic? (Think headless chickens.)

Both tutors stressed the importance of getting a playable prototype made quickly and beginning playtesting cycles as fast as possible.

Dave suggested beginning with a version of the game that is too manic for players to possibly manage, gauge player response, and begin adjusting the game variables from this position (reducing difficulty until a relative relationship is found).

The team will take this advice on board and use it to guide development over the coming weeks – team admit they would likely have begun playtesting with a slow-paced level to test player interactions with mechanics, but agree Dave’s advice of beginning the process at high pace will yield much more telling results.

Tom had managed to complete all modelling tasks over the previous evening/this morning. Team reviewed models, their renders and confirmed suitability within the Unity project.

Team continued with development, able to make changes as per the update level design.

Team found that as both were working on the same scene, GitHub conflicts were occurring frequently. This continued and soon time spent resolving merge conflicts overtook time spent on assigned tasks.

To mitigate this risk in future, team agreed that going forward work will be completed using Unity Collab, then committed at the end of the studio-jam to GitHub.

Unfortunately, these merge conflicts had a significant impact on the jams productivity, leaving incomplete tasks to still be completed.

Both team members are confident that the team is fully aware of other members remaining assigned tasks and expected task outcomes.

No further studio-jam sessions required this week. Any outstanding tasks will be completed independently.

Next team studio jam to be held Monday 19/11/18 following Rob’s group project session.

***Detailed tasks, task descriptions, user stories and time allocations are tracked on JIRA.***

**Tasks for the current week:**

**Tom (12 Hours 50 Minutes):**

* **Edit Unity build to make compatible with Xbox Controllers (20m)**

Set project up to enable input from 4 Xbox controllers at any one time. Edit existing ‘PlayerController’

script to allow for same script to be applied to each player, with each taking input from a different controller simultaneously. Push changes to repository.

* **Edit ship model as per changes discussed in group meeting 12/11/18 (1h 30m)**

Boat model must be widened (along x-axis) without distorting the features of the model.

Side of the ship must be raised to uniform height.

‘U’-shaped cut outs for cannons along the side of the ship.

Addition of mast ‘ring’ to denote lose condition regarding water level.

Add changes to repository.

* **Create ‘grate’ model to cover the ship hold (1h)**

Create a model of the hold ‘lid’ which will be placed over the hold opening during gameplay. Add the model to the repository.

* **Create Treasure Island model (1h)**

Create a model of the island, with rowboat on shore. Island should be decorated with rocks, plants, palm trees. At the center of the island should be a chest ready to be opened to show rewards, with shovels next to it to indicate its recent discovery. Chest lid should be hinged so can be animated to open. Add the model to the repository.

* **Create Whale Tail model (1h 30m)**

Tail of whale only. Create the tail limb so that it is curved to prevent the tail beginning raising above the sea level when it is animated. Add the model to the repository.

* **Create Wood UI icon (3D image) (30m)**

Create 3D model of wood (planks), take screenshot of render to be used as UI icon. Upload to repository.

* **Create Cannonball UI icon (3D image) (30m)**

Create 3D model of cannonball, take screenshot of render to be used as UI icon. Upload to repository.

* **Create Barrel UI icon (3D image) (30m)**

Create 3D model of barrel, take screenshot of render to be used as UI icon. Upload to repository.

* **Create Rock UI icon (3D image) (30m)**

Create 3D model of barrel, take screenshot of render to be used as UI icon. Upload to repository.

* **Create base interface class for non-interactable hazards (1h)**

Create script containing lowest common denominator virtual functions. Upload to repository.

* **Create GameManager class to handle game scene (1h 30m)**

Create script to handle events and variables that will dictate gameplay. Upload to repository.

* **Create Whale script (45m)**

Create script to handle the ‘large wave’ event. Upload to repository.

* **Create Seagull script (45m)**

Create script to handle the ‘clean mess’ event. Upload to repository.

* **Create Rock script (45m)**

Create script to handle ‘avoidance of rock’ event. Upload to repository.

* **Create Wheel script (45m)**

Create script to handle ‘steer ship’ event. Upload to repository.

**Henry (12 Hours):**

* **Create base interface class for interactable tasks (1h)**

Create script containing lowest common denominator virtual functions. Upload to repository.

* **Create mop script (1h)**

Create script to handle the ‘clean mess’ event. Upload to repository.

* **Create cannon script (1h)**

Create script to handle the loading of cannon with powder, with cannonball, with cannonball&powder, and firing of cannon. Upload to repository.

* **Create player controller (2h 30m)**

Create script to handle player movement and player ‘action’ to activate corresponding states in other interactable objects.

Create player states within playercontroller to be used to determine how interactable objects receive input from player.

Create player respawn functionality.

Upload to repository.

* **Create gunpowder script (1h)**

Create script to handle selection of gunpowder, carrying by the player, effect state change within player, and loading of cannon. Upload to repository.

* **Create torch script (1h)**

Create script to handle selection of torch, carrying by player of torch, effect state change in player, firing of cannon. Upload to repository.

* **Create bucket script (1h)**

Create script to handle selection of bucket, carrying by player of bucket, effect state change in player, bailing of water if any present on deck. Upload to repository.

* **Create enemy script (1h)**

Create script to handle spawning and movement of enemies. Enemy health. Enemy return fire. Upload to repository.

* **Create damage script (1h 30m)**

Create script to handle deck damage as a result of enemy fire. Upload to repository.

* **Create repair deck script (1h)**

Create script to handle the repair of damaged deck through use of wood. Upload to repository.