**GROUP PROJECT, GROUP 3**

**DATE: 6 February 2019**

TIME: **14:40 – 15:00**

**ATTENDEES** Tom Gibbs, Henry Crofts

**LOCATION:** *PROGRAMMING ROOM*

**Minute Taker: Tom Gibbs**

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

* Work towards final functionality of game manager
* Work towards final functionality of water level manager
* Work towards final functionality of UI manager
* Begin work/design of tutorial level

**Meeting minutes:**

Both members present.

Team requested a meeting with Chris to review structure and implementation of game and event managers.

Team explained current implementation to Chris who confirmed that logic approach would work, but not efficiently.

Chris explained a far better implementation using a struct within the event manager class with properties to define weighting, max occurrences, cooldown periods and trigger conditions.

Chris explained how this implementation would then allow the weightings of each struct to be summed, a random number selected and a check to see if the selected random is within tiered threshold values which can be used to define the next event.

This implementation can also be expanded to include the existing functionalities for cooldown periods/removing tasks if already present – team predict forcing the whale/rock events to occur less frequently will be necessary, though playtesting will confirm this.

After discussing with Chris it also became apparent that the game manager and evnt manager implementations are no longer needed and methods can be condensed into a single ‘eventmanager’ class.

Next jam scheduled for Thursday 07/02/2019 @ 11:00.

**Tasks for the current week:**

**Tom (12 Hours):**

* **To be completed as part of studio jam, Ensure Event Manager Works Correctly (2h)**

Ensure that the event manager handles the Dictionary correctly holding which events are currently active and removing them correctly from the Dictionary.

* **To be completed as part of studio jam, Ensure Water Level Manager Works Correctly (2h)**

Team should work together to ensure that the water level manager works correctly ending the level when the water level is too high, also raising and lowering the water as required.

* **To be completed as part of studio jam, Ensure UI Manager Works Correctly (2h)**

As a team check that the UI manager works correctly showing the correct events. Spend time fixing any issues with linking into the event manager.

* **To be completed as part of studio jam, give each activity a duration property (30m)**

Team members must work together to ensure a shared understanding for the remainder of development.

Completed task should associate a time with completion of each hazard/task. This should be visually represented by the active players UI indicator. All variables must be exposed in the unity editor for rapid future amendments.

* **Reserved task time, will be split task to be allocated during sprint (all 5h 30m assigned specifically)**

Task to be subdivided into specific tasks once highest priority tasks have been completed.

Team believe manager tasks may overrun anticipated assigned times. All remaining time will be used to address remaining priority issues.

**Reserved task time broken into (tasks also specified within JIRA sprint):**

* **To be completed as part of a studio-jam, review, amend and implement event manager (2h 30m)**
  + Team have found implementation of this manager and it’s interaction with existing managers troublesome. Task used as extension to existing task to continue work on manager.
* **To be completed as part of a studio-jam, bug fix deck interactable behaviour (30m)**
  + Issue identified when items are withdrawn from the ship hold. Team must review script to ensure wood item and cannonball item are interactable on first retrieval.
* **To be completed as part of a studio-jam, bug fix UI manager (1h 30m)**
  + Ensure that managers functionality to select the next available UI image is robust and that sprites are applied to corresponding UI images as dictated by the event manager’s selected event.
* **To be completed as a team, seek tutor advice to restructure event manager (30m)**
  + Request meeting with programming lecturer to discuss current implementation and logic of event manager.
* **To be completed as part of a studio-jam, use tutor advice to restructure event manager (1h)**
  + Using feedback from programming tutor meeting, amend existing event manager implementation to allow for dynamic weighting of events.

**Henry (12 Hours):**

* **To be completed as part of studio jam, Ensure Event Manager Works Correctly (2h)**

Ensure that the event manager handles the Dictionary correctly holding which events are currently active and removing them correctly from the Dictionary.

* **To be completed as part of studio jam, Ensure Water Level Manager Works Correctly (2h)**

Team should work together to ensure that the water level manager works correctly ending the level when the water level is too high, also raising and lowering the water as required.

* **To be completed as part of studio jam, Ensure UI Manager Works Correctly (2h)**

As a team check that the UI manager works correctly showing the correct events. Spend time fixing any issues with linking into the event manager.

* **To be completed as part of studio jam, give each activity a duration property (30m)**

Team members must work together to ensure a shared understanding for the remainder of development.

Completed task should associate a time with completion of each hazard/task. This should be visually represented by the active players UI indicator. All variables must be exposed in the unity editor for rapid future amendments.

* **Reserved task time, will be split task to be allocated during sprint (all 5h 30m assigned specifically)**

Task to be subdivided into specific tasks once highest priority tasks have been completed.

Team believe manager tasks may overrun anticipated assigned times. All remaining time will be used to address remaining priority issues.

**Reserved task time broken into (tasks also specified within JIRA sprint):**

* **To be completed as part of a studio-jam, review, amend and implement event manager (2h 30m)**
  + Team have found implementation of this manager and it’s interaction with existing managers troublesome. Task used as extension to existing task to continue work on manager.
* **To be completed as part of a studio-jam, bug fix deck interactable behaviour (30m)**
  + Issue identified when items are withdrawn from the ship hold. Team must review script to ensure wood item and cannonball item are interactable on first retrieval.
* **To be completed as part of a studio-jam, bug fix UI manager (1h 30m)**
  + Ensure that managers functionality to select the next available UI image is robust and that sprites are applied to corresponding UI images as dictated by the event manager’s selected event.
* **To be completed as a team, seek tutor advice to restructure event manager (30m)**
  + Request meeting with programming lecturer to discuss current implementation and logic of event manager.
* **To be completed as part of a studio-jam, use tutor advice to restructure event manager (1h)**
  + Using feedback from programming tutor meeting, amend existing event manager implementation to allow for dynamic weighting of events.

***Explanation of Task Split on JIRA.***

1. Select the group project from the dropdown menu.
2. Once the project is selected proceed to the backlog where you can see the current sprint and all issues in the backlog.
3. Right click on the issue that has been assigned for contingency in this case the task L6G3–260 and L6G3–261 both named *“To be completed as part of a studio jam, depending on the situation of the project once all other tasks have been completed. As a team assign new tasks and split this task up accordingly”*
4. From the dropdown menu that appears, select “*split issue”* this will then present you with another menu where you can change the task into other tasks without adjusting the scope of the sprint. (provided the number of hours remains the same).

