**PROJECT POSTMORTEM SUBMISSION FRIDAY 4TH MAY 2018**

Once you have made your final presentation WE NEED YOU TO SUBMIT THE FOLLOWING COMPONENTS UPLOADED TO A SEPERATELY LABELLED GITHUB FOLDER

1. A SINGLE PAGE OF A4 (MAXIMUM) WHICH LISTS THE OVERVIEW OF THE ASSETS YOU HAVE PRODUCED FOR THE PROJECT, WHETHER THEY HAVE MADE IT INTO THE FINAL GAME OR NOT.
2. A COMPLETED REVIEW OF THE PROJECT **USING THE TEMPLATE PROVIDED BELOW**. PLEASE REMEMBER THAT THE MORE DETAIL YOU ADD TO THIS COMPONENT THE EASIER IT IS FOR US TO JUDGE YOUR WORK. SO AVOID SINGLE LINES OF TEXT. **EXPLAIN WHAT YOU MEAN**.

|  |  |
| --- | --- |
| **STUDENT NAME** | Tom Gibbs |
| **PROJECT NAME** | Dillama |
| What do you think went well on the project? | At our first meeting I was immediately pleased with the team’s level of collaboration. Each member was respectful of one another’s input, responded well to constructive criticism and built on ideas as a team.  This was further aided by frequent communication. The team were made aware of the importance of clear updates and how they form the foundation of a successful team project.  With only rare exceptions the team managed to maintain high levels of clear emails throughout development, advising the team of their availability, task status and anticipated attendance.  Team identified their availability in an excel spreadsheet. This allowed other members to predict when their work would be logged and their availability for meetings.  Completing tasks as early in the sprint as possible was stressed to the team. Despite having busy schedules detailed in the spreadsheet, the team managed to clear the majority of tasks within the first half of the sprint.  This allowed time to highlight any issues, discussed them and resolve them within the same sprint.  All team members arrived for meetings in a timely manner, and any absences were made known to the team as far in advance as possible, allowing time for the remaining members to adapt.  The team discussed strengths, weaknesses and experience with various software in the first meeting. This allowed the team to efficiently arrange which tasks would be delegated to each member ahead of time. At key stages of the project (when initial design was decided after prototyping, strengths were re-assessed at team meeting to anticipated tasks. This was done again at game redesign).  Using each team member’s advice regarding previous work and strengths estimated completion times began relatively accurately and improved over the duration of the project.  Team was able to adapt well to feedback received, without blame on any member. When it became necessary, the team acted upon advice ad criticism to fundamentally redesigned the project in a very short amount of time.  A group jam was held at the start of each sprint (on the Wednesday). Early sprint jam allowed the priority tasks to be tackled as a team and allowed greater understanding to all team members regarding each other’s tasks. As manager, I was also able to see how each member worked, assessing their capabilities meaning I was better able to gauge how long each task would take to complete. |
| What do you think needed improvement on the project? | Unfortunately, occasional Tuesday sprinting was logged by team members. While some instances were unavoidable due to unexpected issues, others were due to poor time management. Tasks were completed in priority order so the negative impact was reduced, though Tuesday sprinting could have had negative knock-on effects for all the team’s tasks.  While generally reliable, at the end of one sprint Jack advised the team he would be unable to complete his remaining tasks. This notification was sent to the team at 23:30, Tuesday evening – leaving the team insufficient time to adapt and compensate for the unfinished work.  Keeping fellow team member’s knowledge of your situation in mind while working on tasks would have helped prompt members to share their current task status with the team.  Inaccurate updates, with specific regard to JIRA. Inaccurate logging of worked hours lead to inaccurate estimated completion times being assigned. There were also times when members would forget to move a started task from ‘to do’ into ‘in progress’. This lead to multiple team members attempting to commit work to the repository at the same time, causing git conflicts which had to be resolved.  Real time task updates would have eliminated this problem.  Response to the brief – despite producing and playtesting multiple early design implementations, we focused too much on the games mechanics rather than giving primary concern to player experience. We later found this error undermined the development process, necessitating a redesign. This redesign restricted available development time and meant some stretch goals had to be removed to guarantee completion.  While limiting the project scope proved to be the right choice, as manager I should have foreseen this becoming an issue in development and continued the initial design process and playtesting to ensure appropriate mechanics and the intended player emotions were evoked before progressing the team. |
| What do you think of your own contribution to the project? | Strong.  I was responsible for all functionality included in both the final build and all early prototypes, as well as creating all scenes and importing assets into Unity.  Without building of these early prototypes we would not have reached the final design used.  Guided the first-year designers through Unity where appropriate, as they were unfamiliar with the engine.  As the project progressed and designers required increasing access to the Unity build to amend variables. I learned that to ease collaboration within Unity, I should rename and restructure the build to make it as intuitive as possible.  I also kept in mind that if I should be absent while the team had an issue with code and required the help of a tutor, having well-structured and commented code would assist the team in understanding the problem.  Created a spreadsheet detailing the team’s availability which they were requested to populate during the first sprint.  This was revised and repopulated for the Easter period, to ensure all members remained aware of each other’s availability.  Ensured clear and thorough communication throughout all means of recording information (emails, group meetings, meeting minutes, JIRA tasks), to ensure all members understood all other member’s availability and expected outcome of tasks. This assisted the team greatly on all collaborative work throughout development.  This was done at the beginning of each sprint.  I scheduled team jams for every Wednesday (the start of the sprint). This allowed the team to tackle the priority tasks together, and gave the first year’s opportunity to ask for help if needed. It also allowed me, as manager, to assess their abilities further to what had been initially advised in group meetings and gauge their efficiency of work, improving the accuracy of allocated task time.  If any tasks were still outstanding on a Monday, I would remind the team of their outstanding tasks and request an update as well as anticipated completion time.  I set members deadlines for tasks, depending on the task priority and whether its completion hindered the ability to progress other game elements.  When a member appeared to be struggling with a task, or I doubted their reliability, I set an early deadline for completion of the first third of task time allocated. This allowed me to confirm work was being completed, and prevented any unnecessary time wastage as I could check whether work was completed to the required standard and if not, the remaining task time could then be spent rectifying any issues or allocated to a different task.  Jack occasionally proved unpredictable in terms of quality of work and whether all tasks would be completed. This approach of early deadlines for the first third of tasks proved to be an effective way to combat this behavior.  Appropriately limited the scope of the project, completely removing certain elements and setting others as stretch goals. Provided structure for the entire development process, defining objectives for each sprint.  When it became necessary to redesign the project approximately half way through development, I prevented the designers from including the majority of the proposed features to ensure a playtested final project would be completed before the final deadline, uncompromised by overambitious elements.  Once the team has playable prototypes, ensured playtesting tasks were set each week so that the iterative process was informed by feedback. |
| **OVERVIEW** |  |
| **Thinking about the project you have worked on this year, what are the important lessons that you will take away from the experience for your next group project?** | Holding a minimum of one jam a week would enabled the team to work together on high priority tasks. Learn how each team member operates and provide an environment for in person discussion regarding task issues, as they happen, as opposed to over email.  Opening the team to a discussion of each members experience and capability with various software/tasks at the start of the project will help identify an appropriate scope for features to be included within the time frame, as well as a more informed range of allocated hours per task.  It is necessary to have a more extensive initial design phase to ensure the game is designed with player experience in mind rather than pure mechanical response the brief. Spending more time in the design phase, developing primitive prototypes and obtaining feedback would have been of huge benefit – giving a greater chance of a successful response to the brief earlier on, reducing the potential need for fundamental redesign in the later stages of development, saving wasted time and yielding a more polished final product.  Using previous knowledge to limit the scope of the project to a level where a player experienced can be refined and polished repeatedly.  That maintaining a continuous awareness of others’ tasks is key to successful collaboration. Each member being aware of what each other member is currently working on helps to avoid conflicts in committed work, and understanding what the mutually expected outcome of tasks will be ensures that all work produced adheres to the same theme and function as intended.  If members provide an anticipated availability schedule, and update the team with regular status updates, workload can be assigned accurately, and confusion in any areas clarified.  Consistent general updates, even to say work is proceeding as expected will help all members understand the status of tasks and help team members anticipate potential issues, preparing for them earlier and more thoroughly.  That any tasks left incomplete, or team members not contributing necessary levels of effort can have significant negative knock-on effects on other team members tasks. This is made worse if tasks are high priority or blockers.  It is crucial that the team is thankful for constructive criticism and is able to use feedback to inform iterative development.  Voice recording meetings can provide a mean for production of more accurate minutes, as well as giving members not able to attend a better foundation for understanding missed elements of the design process.  Inaccurate JIRA updates, have a negative impact and should not be allowed to occur. Be it embellishment of task quality/quantity interfering with later tasks or inaccurate logged hours meaning future tasks are estimated incorrectly. |

**Tom Gibbs, S184605 – assets produced:**

Unity scripts:

* GameManager.cs
* AudioManager.cs
* ScenesManager.cs
* MenuManager.cs
* Vibration.cs (plug-in)
* VibrationFunction.cs
* PushController.cs
* PlayerController.cs
* CameraShake.cs
* CountdownTimer.cs
* PowerUp.cs
* algorithm script (earlier method of negative-feedback loop)
* various scripts to provide prototype build functionality

Responsible for entire Unity build and all prototype builds

* Scenes, art placement, layout, project structure
* All archived builds
* All archived .apk files