|  |  |
| --- | --- |
| **STUDENT NAME** | Tom Gibbs, S184605 |
| **PROJECT NAME** | GROUP PROJECT: “Ship Happens” |
| What do you think went well on the project? | Both team member’s motivation and commitment to all aspects of the games develop since first meeting and willingness to collaborate respectfully. Throughout the project both team members were humble of one another’s design input, were not afraid to adapt their own ideas in response to criticism, allowing ideas to be shaped as a team.  The teams open and honest discussion of strengths, weaknesses and experience on previous projects/with different software allowed for definition of project scope and an estimation of the end product early on in development.  When necessary I was able to adapt to suit the needs of the team and research into 3D model production (which I do not have prior experience in) to a sufficient degree to produce and adapt our assets when development required it.  When the team had opportunity to take on an additional designer team member very late in development, despite taking a portion of development time away from a sprint to being Amy up to speed with our goals and progress, all team members were able to integrate into the new dynamic almost immediately and continue with game iterations.  Collaboration was aided by frequent communication.  Team were in near-constant communication from inception of the project to the final submission. Primarily through in-person studio-jams, frequent in-person discussion outside of meetings and via discord messages when the first two were not possible.  Completing work within jams whenever possible helped team to understand each other’s positions and efficiently complete work.  Almost all work was completed within studio-jams for the entirety of the project. This efficiency and ability to work fluidly to adapt as problems arose, working issues as a team continued to be a huge help throughout development.  The team created availability spreadsheets in addition to keeping members updated of their availability, which helped greatly in predicting capacity for work during periods off requirements from other modules or external commitments. This allowed the team to plan ahead to mitigate risk of reduced development time, sharing workloads and reassigning tasks to meet client deadlines if the need arose.  This constant awareness of each other’s work status also meant that when an issue with JIRA effected other groups development, our team was largely unaffected with no negative impact to completion of sprint tasks.  The transparency between team members also allowed for more accurate task time estimates as the project went on, boosting efficiency further.  Once a functional version was completed, team began public playtesting.  Playtesters were carried out in person using questionnaires and interviews, online with attached surveys and with public testers external to fellow students on our course for more candid feedback.  All feedback received proved invaluable for further iterations.  Through playtest feedback the team was able to identify weaker or inconsequential game elements, making the decision to remove several from the project scope to focus polish on elements fundamental to player experience.  The same cutting of content to achieve the best polish possible saw the team make hard, but correct, decisions to remove rewards schedules and several game mechanics from the project scope which a large amount of design and implementation time had already been invested in creating.  These choices allowed for further playtesting and feedback of more fundamental game elements which have proven to have a greater impact on player experience and has a hugely positive effect on the project as a whole.  This allowed the group to remain acutely aware of our limitations as a team of only 2 programmers, with a broadly overlapping skillset. When designing initial ideas, the team assessed our realistic abilities and imposed development restrictions (criteria such as a small finite gameplay area to reduce the amount of art assets needed) which all proposed designs needed to adhere to.  When presenting early designs to lecturers, the team took all advice given. Feedback indicated Lecturers had concerns that the team lacked the design knowledge necessary to effectively create the player emotions involved in our design.  The team adapted to this quickly, deciding to halt any code or asset production and spend the first phase of development researching and applying relevant design theories, then seeking design advice from lecturers before producing code.  The was one of the most beneficial periods of the project and using the feedback to inspire development dramatically improved the quality of the team’s final product.  Once the team had completed research and design of the project, the team produced a thorough design document, risk assessment and project timeline to completion which helped guide and track development during the project.  The design document proved to be a valuable resource when requesting feedback regarding project design from lecturers and the risk assessment allowed the team to mitigate several risks and look to plan ahead for the latter stages of the project.  Remaining aware of risk as two programmers, team collected modular asset packs to reduce the need to create our own game models, increasing capacity for development of game function and project polish.  With the advice of lecturers, the team produced a psychographic which further helped influence choices made during later iterations and kept the project on track.  All team member’s continued to follow the advice of lecturers following production of our prototype. Lecturer recommended reading material (of most benefit were “Casual Game Design” – G. Trefery, “Hooked” – N. Eyal and “The art of Game Design” – J. Schell) saw our designs regarding reward schedules and player investment improve.   * Unity master builder optisimation talk, helped us with project * Was able to model/edit existing models when asset packs would notsuit our needs   Team took pride in creation and delivery of presentation materials which received consistently positive feedback from tutors.  Team were able to improve project and working manner by arranging regular tutor meetings to discuss project management, design theory and code implementation.  The team also sought advice from a Unity master builder while lecturing at UoS who’s advice had a dramatic increase to game run-time efficiency. |
| What do you think needed improvement on the project? | While ultimately of huge benefit to the team, it was necessary to sacrifice development time at the start of the project to developing the team’s knowledge of design theory. This took many subsequent weeks of sprint research time until the team felt confident, though was essential as it was used as the foundation of our games design.  Use of version control software. Team have learned that a single product should be used (not Git AND UnityCollab).  While the team were able to respond well to time lost because of merge conflicts, the eventual outcome was only one member being able to edit the main build at a time until the issue was resolved, which resulted in time being wasted unnecessarily.  Before adopting a better version control approach, fixing merge conflicts was a time-consuming process for both members, taking development time away from developing the project. It also resulted in having to reassign tasks to recreate functionality which had been produced before the merge conflict occurred, further adding unnecessary development time.  While communication was thorough and constant this did lead to a few instances of brief conversations not being included within the groups records, although any significant points were appended to formal meeting minutes.    Team were occasionally guilty of inaccurate JIRA updates, where work would be completed as part of a studio-jam, and team members would not remember update JIRA in real time.  Team communication was constant so there were no negative issues internally, though this practice did risk potentially misleading project investors.  Although team members always contributed the minimum task time agreed during the sprint even if tasks were completed early, under/overestimation of task times lead to less efficient development, where accurate estimates would have allowed the team to better plan tasks at the start of the sprint.  Neither team member has previously developed a project with so much time spent playtesting. This improved the project greatly, though also revealed many game elements which were redundant/issues/performed better than expected – which did not always align with expectations of the team and meant priorities had to be reviewed.  This meant the project timeline often had to be revised with unexpected tasks needing attention, and many game elements which the team invested time into were removed from the scope of the project as needs changed.  Despite detailed attempts to plan code structure and project functions to work together and scale with projects needs as designs are iterated, the team found that often weekly tasks would involve the re-writing of existing code to either make it more reliable or compatible with other changing elements.  Similarly, playtesting would occasionally reveal testers attempting to interact in a way the team had not predicted, or intentionally trying to break the game. This lead to significant development time being assigned to ensuring all features were robust and hotfixing player exploits.  After experiencing very similar code problems repeatedly, the team agreed issues were not being solved effectively so scheduled multiple meetings with Chris to review our implementations, which while took time away from developing further game function did prevent the need to regularly adapt existing completed work.  The team spent a large amount of time in the initial phase of the project purely planning the gameplay experience and how players would be taught the game mechanics via a tutorial. While at the end of the project testers seemed to genuinely enjoy gameplay, the team struggled throughout the project to make the tutorial an engaging experience for two players.  With the late addition of a designer with experience in ‘player-onboarding’ the team were able to make significant improvements in latter development, but there was not enough remaining time to fully rectify the issue.  Many of the 3D model asset packs that the team had investigated as a way of mitigating the risk of asset production, became inadequate as designs were iterated. This meant that on occasion I would have to stop developing the functionality of game elements, and instead model/adjust existing models for game purposes, given that I was more familiar with 3D modelling than my team mate.  Because of my lack of experience in this area, simple tasks often took an inflated amount of time to complete beyond the team’s best estimates - further restricting ability to contribute toward game function within those sprints.  The majority of models I was responsible for creating/editing were comparatively non-performant. Models had high poly counts and did not allow for sections of the same model to have different Unity materials applied which would have meant production/sourcing of textures by the team which is further out our knowledge.  As such, I had to have subsequent tasks to re-edit models and learn how to appropriately configure models which was necessary to development but did take further time from developing functionality.  As the project neared completion, during the Easter period the team struggled to find playtesters to test the game and give interview feedback. While the team were able to test with a lower than usual number of playtesters, a higher percentage than normal were testers with previous exposure to the game.  The feedback obtained was vital to guiding development, but the team has found that a smaller number of testers can lead to outliers in feedback given.  The games collision detection is sometimes unreliable which occasionally gives false values where positives should return. The team learned of layer-based collision detection which is a more consistent and efficient alternative late in the project and decided that applying polish to existing elements was of greater benefit than risking an overhaul of integral parts of the project functionality. If more time to enhance the gameplay experience and reduce player frustration this would be updated.  Team rewrote code in a more optimized manner to increase efficiency mid-way through the project. The team should have accounted for the need of these data structures earlier in development which would have eliminated the need to spend further task time on already implemented functionality.  Not having a designer/visual asset producer on the team (prior to the addition of Amy) did mean that the two programmers had to adapt and learn elements of other disciplines more closely. This was time-consuming and took team members away from their strengths, slowing production. If a suitable designer had been available at the project start it would have been of great benefit to the team to being their knowledge onboard to allow team members to play to their strengths and maximize efficiency.  While the team adapted appropriately to the amount of time remaining, the team had overscoped the project during the initial design phase.  The team had spent a significant amount of time researching and designing reward schedules that ultimately had to be removed from the scope of the project to allow for high level of gameplay polish which the team prioritized. |
| What do you think of your own contribution to the project? | Very strong.  Throughout the project I took the advice of tutors on immediately. When advised that the project would benefit from a deeper understanding of design principles I halted all programming and spent time beyond that assigned in tasks to learning appropriate design theory and applying it to our games design before any functionality was added.  I was involved in all stages of the project’s design/implementation/iteration.  I proposed and implemented many game design and gameplay elements that have demonstrably improved playtester responses over successive playtest sessions.  My communication with my team members was continuous and constant throughout the project. Any issues with task, absences I had to have or reduced hours I anticipated needing were raised with the team in advance of negotiating tasks with the intent of mitigating as much risk as possible.  I continued to adapt to the needs of the team, learning how to 3D model and producing assets for the group as our design was iterated and asset packs no longer met our needs.  Every week I was involved in project management and deciding the priorities of the project – often making difficult decisions to adjust the current focus of the project and remove content when necessary.  Contributed as much time to the project I was able so that the team could be prepared to hold playtesting sessions to inspire iteration as frequently as possible.  Completed as much work as possible within studio-jams to maximize team efficiency.  Proposed creating admin spreadsheets to best identify team’s availability for studio-jams and completing tasks.  Produced an auditable history of group development including meeting minutes, photos of designs during studio-jams, voice/video recordings and creation and delivery of presentations. |
| **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?** | Team ability.  The importance of thorough and specific discussion of team member’s strengths, weaknesses, abilities and experience at the initial phase of the project to help define project scope.  When allocating tasks team member’s experience can also be used to inform anticipated completion dates and project timelines.  Initial design phase.  The immense benefit of thoroughly researching the design theory and application of research to the project before rushing into building a functional prototype.  This is the first game project I have worked on that has had such a significant design/research phase prior to development and the only project that has not required a fundamental overhaul during the project - and I believe it is because of the attention taken over the initial phase.    The importance of studio-jams.  As much development as possible should take place within a studio-jam. Ensuring that the development team share an understanding of what is expected from each task, where each member is in development and how each member works is crucial to working effectively and efficiently together.  Project polish.  This is the first large-scale project I have worked on where the team was able to reach a point in development where playtesting and iteration were able to be carried out for subsequent weeks for a significant percentage of the project.  As such this is the most player-enjoyed project I have ever produced and believe it is the first time a project has come close to eliciting the emotions in players that we had designed for.  While always knowing its importance, I have more appreciation for ensuring there is sufficient project time allocated to iterative playtesting.  Accepting feedback.  All feedback received is valuable and should be the driving force behind any changes made to the project.  It is crucial that the team is thankful for constructive criticism and is able to use feedback to inform iterative development.  Regular and continuous communication.  Without this our project would not have been nearly as successful. It allowed the team to adapt to changing, feedback and new goals while also giving us opportunity to spot potential approaching risks and allocate development tasks to avoid unnecessary risk.  Project scope.  Limiting scope of the project to realistically achievable amounts.  It is necessary to understand team capabilities/limitations of technologies as well as realistic timescales for completions of tasks. This will avoid future conflicts and prevent uneven expectations between investors and developers.  Staying continuously aware of other members workloads.  That maintaining a continuous awareness of others’ tasks is key to successful collaboration. Reduces likelihood of merge conflicts 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. It is a weakness for only a specific member to be able to do a specific thing, all members must be able to pick up work when needed.  Being made aware of anticipated group availability is also a great help when organizing workload and predicting whether other members are available to help compensate for other’s unavailability.  Inaccurate JIRA updates have a negative impact and should not be allowed to occur. All updates should be made in real-time for the benefit of internal team members awareness of others work, as well as to allow for monitoring from investors perspectives.  Setting up software appropriately to avoid merge conflicts  To be more efficient, the team tried to use two different types of version control software. However, this lead to many merge conflicts, resulting in lost work and wasted time.  In future I will look to avoid overcomplicating workflow as the potential risks outweigh the potential benefits.  Accurate and regular documentation of development.  Assists with internal understanding, ensuring development stays on track. Can be revisited to confirm (quality of) completed work.  Can be used as basis for discussion with client and is of great assistance in producing client presentations.  The team were fortunate to be able to have a designer join the team in the late stages of development, who was able to contribute more efficient revised assets despite the game design already having been completed.  In future it would be beneficial to have a team with more balance across varied disciplines.  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.  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. |