Battleship: Team Crossover Edition
RETROSPECTIVE

Team

- Chase Stump
- Marco Borth
- Jack McClure
- Rachel Elting
- Jarod Davis

Log of all meetings

Meeting 1 Summary

Date: 9/6/2019

Location: Eaton 2

All members present

We discussed how to approach the battleship project. Options considered were creating a battleship game on Unity Engine with C# scripting or on the web using JavaScript. We discussed how to implement the game using various components such as parent/child objects, 2D vs 3D rendering, making the board interactive with mouse click, etc. Members were encouraged to review basic Unity tutorials. Only one member was familiar and fluent with C# and Unity engine. The team decided to communicate via Discord when unable to meet in person.

Meeting 2 Summary

Date: 9/9/2016

Location: Eaton 2

All members present

Unity was the chosen development tool to complete the project, unanimous decision amongst the team members. The discussion at this meet was identical to the first, with further consideration as to how to build the game from the ground up, and how to set up the hierarchy of objects within the game.

Meeting 3 Summary

Date: 9/11/2019

Location: Fishbowl

All members present

At this stage some of the basic components have been made. Boards have been created and are interactive via mouse. Basic mouse hovering and clicking causes color change for the individual tiles. The main menu is currently under development, which will allow users to select how many ships to start the game with, 1 - 5 ships, as well as the option to quit. Ship design and scoring is under development. Version control with GIT is being worked on and a repository is being set up for all members to push/pull to/from.

Meeting 4 Summary

Date: 9/16/2019

Location: Fishbowl

All members present

Boards have been updated to allow for images to show after clicking on the board. We are currently working on getting the program to restrict user input to one board at a time rather than both for turn based play. The UI has been updated to show menu options and text. Currently working on displaying/hiding buttons and options when certain selections are made by the user. Ships have been updated and contain skeletons for the functions needed for them to work. As the ships are further developed, the scoring system can be developed. We discussed documentation method and agreed on Doxygen as the tool used for documenting the C# scripts.

Meeting 5 Summary

Date: 9/18/2019

Location: Fishbowl

All members present except chase - out sick, and communicated via Discord

Boards have been updated to allow interaction one at a time for turn based play. UI scene has been integrated with the boards scene bringing the project closer together. Ships have been updated, references to the components have been made so that scripting can be done between the UI, boards, and ship components. Now the remainder of the components need to be merged into one scene for a single project.

Meeting 6 Summary

Date: 9/19/2019

Location: Fishbowl

All members present

Bugs on the boards have been fixed, allowing for single fire on a single turn before switching boards. A second UI grid was created to allow for separation between menus. Ships can now be instantiated when selected in the menu before the game begins. A bug was fixed that didn't allow for rotation of the ships for vertical orientation. Ships now need collision detection implemented in the individual ship parts to communicate with the board.

Meeting 7 Summary

Date: 9/20/2019

Location: Fishbowl

Members present: Marco, Rachel, Jarod

Short meet in the evening. We are approaching the finishing steps for the game. Final ship drag and drop feature is being implemented, and colliders are being added to set the ships on the board. Doxygen is up and running and proper documentation can now be done. Scoring system is the last component to be completed, aside from minor fixes if time allows.

Meeting 8 Summary

Date: 9/21/2019 & 9/22/2019

Location: Discord/Online

Members present: All

Final sprint to complete the game and all required documentation. Retrospective was was updated to include a works cited. Most of the bug testing took place over these two days and most were fixed. Ships now move with the mouse correctly when dragged and dropped. Fixed ship size consistency when playing on different resolutions, game UI consistency when switching between players.

Along with the major meets, communication took place in class and in Discord.

Work Split

- **Board design** - Jack McClure
- **User Interface** - Marco Borth
- **Ships** - Chase Stump & Jarod Davis
- **Scoring** - Chase Stump & Rachel Elting
- **Documentation** - Set-up completed by Rachel Elting, all members added proper comment formatting.
- **Git** - Setup done by Jarod Davis, all members committed and pushed to the repository.
- **Bug Finding and Testing** - Primarily playtested by Rachel Elting, Marco Borth, and Jarod Davis.

Challenges

- The Unity engine and C# scripting was new to everyone but one of our members. This slowed development progress. Frequent communication between members kept progress from halting altogether. We used Unity tutorials and C# references to help resolve issues.
- Gaining familiarity with the branching and merging aspects of Github have proven to be a difficult task. Some members have taken the habit of deleting and re-cloning the repository rather than pulling.
- Bad push to repo caused the project to fail. The team worked to revert to a previous working commit and implement correct changes.

Features that didn't make it

- Bonus items in game to provide tactical advantages to players in finding enemy ships or protecting their own ships.
- We planned for aesthetic additions such as ship sprites, water, sound effects, and music, but delays in development resulted in too little time to implement these ideas.
- By the end of this project we realized we'd need more than double the time to set up an online multiplayer version of the game.

What we would have done differently

- Using unit testing may have sped up the development process by helping us find errors or unexpected results faster.
- Having started the project on a single “scene” - or version of the project - in unity would have helped everyone stay in sync while developing.
- Scheduling more in-person meetings, especially in the earlier stages of development. Communication is much more clear in person.
- Ensure everyone has a task that can be worked on earlier in the project. One teammate’s major task was more relevant later in the project, but ended up melding into other tasks, making the work split less even.
- Add comments/documentation in earlier stages of development to promote readability between teammates.