“Five Guys” Team Meeting 1 – 9/27/19

ATTENDEES: Rob, Qui, Afnan, Cole

LOCATION: Spahr Auditorium (Eaton 2)

* Need to figure out documentation method
* Team meetings can occur via Discord or in-person
* Afnan, Beau, and Rob will focus on AI portion of project
* Qui, Cole, and Cameron will focus on implementing new features into the project
  + Qui is likely going to be adding a scoring system
    - Hit percentage
    - Miss percentage
* Need to familiarize ourselves with the code-base we have inherited
* Need to familiarize ourselves with Python3.7.4 and PyGame
* Need to fix the inherited code-base to adhere to traditional rules of Battleship
  + Player should NOT get to attack repeatedly after hitting an opposing players ship

“Five Guys” Team Meeting 2 – 9/30/19

ATTENDEES: Rob, Qui, Afnan, Beau

LOCATION: Spahr Auditorium (Eaton 2)

* New additions/features + confirmation with Dr. Gibbons (COMPLETED)
  + Qui : Scoreboard
  + Cole : If you sink a ship, you can move one of your ships (BE CAREFUL TO NOT RUIN A GAMEBOARD)
  + Cameron : Sound Effects (MAKE SURE TO HAVE WORKS CITED)
  + Afnan : Special Shot (Nuke / 3x3 attack)
* Talk with group about when we can meet in-person and our upcoming schedules
  + Beau is gone 10/11-10/13
  + Afnan is gone 10/11-10/15
  + Qui is studying for exams (no vacation)
  + Rob has not made plans yet (might go on vacation)
* Need to familiarize ourselves with the code-base we have inherited
* Need to familiarize ourselves with Python3.7.4 and PyGame
* Need to fix the inherited code-base to adhere to traditional rules of Battleship
  + Player should NOT get to attack repeatedly after hitting an opposing players ship
* AI will use new game-state (Rob, Beau, Afnan)
  + EasyAI / MediumAI / HardAI
* ALL : Fix game rules
  + should NOT get to fire again after getting a successful HIT

“Five Guys” Team Meeting 3 – 10/2/19

ATTENDEES: Rob, Qui, Beau, Cole

LOCATION: Spahr Auditorium (Eaton 2)

* Talk with group about when we can meet in-person and our upcoming schedules
  + Beau is gone 10/11-10/13
  + Afnan is gone 10/11-10/15
  + Qui is studying for exams (no vacation)
  + Rob has not made plans yet (might go on vacation)
* Need to familiarize ourselves with the code-base we have inherited
* Need to familiarize ourselves with Python3.7.4 and PyGame
* Need to fix the inherited code-base to adhere to traditional rules of Battleship
  + Player should NOT get to attack repeatedly after hitting an opposing players ship
* AI will use new game-state (Rob, Beau, Afnan)
  + EasyAI / MediumAI / HardAI
* ALL : Fix game rules
  + should NOT get to fire again after getting a successful HIT
* Afnan, Beau, and Rob come to lecture on Friday with an understanding of our inherited code-base so that we can start to plan what we have to do.
* Qui will be at Engineering from 11am-6pm tomorrow
* Let’s get some work done on Friday (PLANNING)
* Make UML Class-Diagram for each file in the inherited code-base
* FRIDAY FUN DAY ON 10/4
  + TEAM BASED ACTIVITY

“Five Guys” Team Meeting 4 – 10/7/19

ATTENDEES: Rob, Qui, Beau

LOCATION: Spahr Auditorium (Eaton 2)

* Talk with group about when we can meet in-person and our upcoming schedules
  + Beau is gone 10/11-10/13
  + Afnan is gone 10/11-10/15
  + Qui is studying for exams (no vacation)
  + Rob will be in Overland Park/Lawrence for Fall Break
* Began laying out plans for how to implement AI w/ Game States
* Python3.7.4 and PyGame
  + Inherited project runs poorly on Mac 🡪 Runs GREAT on Windows machines
* AI will use new game-state (Rob, Beau, Afnan)
  + EasyAI / MediumAI / HardAI
    - Basic algorithm designs discussed
* FIX INHERITED CODE-BASE:
  + If player hits, then they get to keep firing until they miss
    - FIXED THIS ISSUE THEN...
      * If player misses, then they get to fire again
        + FIXED THIS ISSUE
* Discussed questions we had about the code-base between each other
* Qui began working on the Score Board
  + Formulas created
  + Basic display of Score Board discussed

“Five Guys” Team Meeting 5 – 10/9/19

ATTENDEES: Rob, Qui, Beau, Afnan

LOCATION: Spahr Auditorium (Eaton 2)

* Talk with group about when we can meet in-person and our upcoming schedules
  + Beau is gone 10/11-10/13
  + Afnan is gone 10/11-10/15
  + Qui is studying for exams (no vacation)
  + Rob will be in Overland Park/Lawrence for Fall Break
* Python3.7.4 and PyGame
  + Inherited project runs poorly on Mac 🡪 Runs GREAT on Windows machines
* AI will use new game-state (Rob, Beau, Afnan)
  + EasyAI
    - randomly generate I and J (ROW/COL) between 0 and 8
    - follow trackRects from “if isPoint...” on down
  + MediumAI
    - if(globalShipHitsAI is not null)
      * use fireAdjacent function to generate ROW and COL to fire on
      * if HIT
        + if SUNK

globalShipHitsAI = remove coordinates of ONLY the sunk ship

globalShipMissAI = null ???

* + - * + else

add coordinate to globalShipHitsAI

* + - * if MISS
        + add coordinate to globalShipMissAI ???
    - else
      * fire randomly
      * if HIT
        + add coordinate to globalShipHitsAI
  + HardAI
    - simple, just need to decide on order of hits (randomly hits ships? or goes smallest to largest?)
* Discussed questions we had about the code-base between each other
* Rob added the buttons to the welcome screen for VS EASY AI, VS MEDIUM AI, and VS HARD AI (clickable as well)
  + USE GAME STATES TO HANDLE AIs
* Created new functions:
  + trackRectsHuman
  + trackRectsAI
  + printRectsHuman
  + printRectsAI
  + trackPlayButton\_AI
  + setUpPlaceBoatsAI

“Five Guys” Team Meeting 6 – 10/16/19

ATTENDEES: Rob, Qui, Beau, Afnan, Cole, Cameron

LOCATION: Spahr Auditorium (Eaton 2)

* Python3.7.4 and PyGame
  + Inherited project runs poorly on Mac 🡪 Runs GREAT on Windows machines
* AI will use new game-states (Rob, Beau, Afnan)
  + EasyAI
    - randomly generate I and J (ROW/COL) between 0 and 8
    - follow trackRects from “if isPoint...” on down
  + MediumAI
    - if(globalShipHitsAI is not null)
      * use fireAdjacent function to generate ROW and COL to fire on
      * if HIT
        + if SUNK

globalShipHitsAI = remove coordinates of ONLY the sunk ship

globalShipMissAI = null ???

* + - * + else

add coordinate to globalShipHitsAI

* + - * if MISS
        + add coordinate to globalShipMissAI ???
    - else
      * fire randomly
      * if HIT
        + add coordinate to globalShipHitsAI
  + HardAI
    - simple, just need to decide on order of hits (randomly hits ships? or goes smallest to largest?)
* Cameron and Cole will be working together to implement Sound Effects
  + Either find and cite free, open-source sounds from online OR create your own sound effects
    - Make sure to include citation information if sounds are taken from online
* Afnan, Beau, and Rob will work on implementation of AI and AI testing throughout next few days
* Qui made progress on GUI scoreboard
* Bug fixes made to how the game runs
  + More bugs created in the process

“Five Guys” Team Meeting 7 – 10/18/19

ATTENDEES: Rob, Qui, Beau, Afnan

LOCATION: Spahr Auditorium (Eaton 2) & Fishbowl

* Python3.7.4 and PyGame
  + Inherited project runs poorly on Mac 🡪 Runs GREAT on Windows machines
* Cameron and Cole will be working together to implement Sound Effects
  + Make sure to include citation information if sounds are taken from online
* AI team (Beau, Afnan, Rob) meeting up today at about 4PM to fully implement AI and test the AI functionality
  + Will also work on bugs as we find them
    - AI is continually firing (extra loop)
    - Scoreboard is not updating properly anymore
* Austin has been making progress on Medium AI function fireAdjacent()
  + it’s been a difficult task... Ran ideas for his implementation by Beau and Rob
  + Austin pushed fireAdjacent() to master branch
    - for coord in reversed(shipHitsAI):
      * if(aboveCoord is on board and has not been shot at)
        + return aboveCoord
      * elif(rightCoord is on board and has not been shot at)
        + return rightCoord
      * elif(bottomCoord is on board and has not been shot at)
        + return bottomCoord
      * elif(leftCoord is on board and has not been shot at)
        + return leftCoord
    - if fireAdjacent is called, you already know that it WILL find a good place to shoot at
    - shipHitsAI = [(x\_1, y\_1), (x\_2, y\_2), ...]
* As of 530PM today, Human vs Human, Human vs AI (all difficulties) are functional and mostly tested
  + Documentation API was discovered and documentation has been written for the new functions we implemented
    - Afnan will download Sphinx Documentation API (<https://www.sphinx-doc.org/en/master/usage/quickstart.html>) as well as update the Docs files to reflect the new function documentation
  + showboatHuman was implemented and fixed
  + Scoreboard bug was fixed
  + Initial testing of Human VS Easy AI went great, no issues at all
  + Initial testing of Human VS Medium AI went well, no obvious issues
  + Initial testing of Human VS Hard AI went well, no issues at all
* Cameron and Cole still need to implement sound effects
  + Works Cited text file!!

“Five Guys” Team Meeting 8 – 10/20/19

ATTENDEES: Afnan, Beau, Cameron, Cole, Qui, Rob

LOCATION: Remote (Via Phone)

* Cameron and Cole implemented sound effects they found online into the game:
  + placing boats
  + changing players/turn
  + miss
  + hit
    - victory sound was playing on repeat so decided to not implement due to inability to prevent game states from always looping during the game
* Works Cited for the new sound effects files has been included in the Docs/WorksCited.txt file
* Documentation was completely finished last night by Afnan and Rob
* All testing so far has not uncovered any bugs with the program... further testing would be smart just for safety
  + game is 100% playable in all 4 game modes

“Five Guys” Project 2: You’re Fired!

Afnan Latif, Beau Hodes, Cameron Tomka, Cole Adams, Qui Phan, Rob Chirpich

**Retrospective Write-Up**

1. Description on how work was split between teammates:

When this project was assigned to us, we decided that Afnan, Beau, and Rob would focus on getting the inherited code-base up to speed so that we could begin to implement the three different difficulties of AI to play Battleship against. Once a few minor bugs had been fixed, the original codebase was a proper recreation of Battleship in GUI format. Rob also worked on updating the GUI interface to display and react to the input from the user.

For the new additions to the inherited code-base, the team was Cameron, Cole, and Qui. Qui decided to work on implementing a scoreboard that would display statistics on both players in all game modes so that the players could see how many times they attacked, how many hits they had, and how many misses they had throughout their game. Cameron and Cole implemented the sound effects together so that the player could hear certain sounds when they placed their ships, attacked and hit an opponent’s ship, attacked and missed an opponent’s ship, and also when the game is switching to another player’s turn.

For the Meetings Log and Retrospective Write-Up, Afnan and Rob worked together on this portion of the project. The Meetings Log document was updated regularly to reflect the details of our meetings and the Retrospective Write-Up was completed at the end of the project.

As far as testing is concerned, everyone participated in this and worked together to find and fix any bugs we encountered.

1. Challenges and how they were overcome or dealt with:

This project was a bit challenging in terms of working on someone else’s code and also some of us had to learn a new programming language to do this project. Early on, the inherited codebase was very overwhelming to read and attempt to understand, but taking notes over the code and talking with teammates really helped to ease the learning curve of the inherited codebase.

We also struggled a bit with who was going to work on what new features to implement as the variety of things we could have added was pretty large, but we selected a few features and ended up implementing just two of those features.

We also had some trouble figuring out how to implement the different AI difficulties, but it became fairly easy once we settled on using new game states to differentiate between the AI’s. The easy AI implementation wasn’t too difficult and that was the first AI to be completed. The hard AI implementation was the next to be implemented as it already knows where to attack each and every turn. The hardest implementation was the medium AI as the fire adjacent function was a little tricky to create, but working together, we were able to create it and implement it shortly after the easy and hard AI’s.

Ultimately, working together on the different additions we had to make for this project was really helpful and definitely made the project come together within the last week of working on it.

1. Any features that did not make the demo version:

There were features which we discussed including the ability to allow a player to replace a boat if they sunk an opponent’s ship. We also discussed adding a “special shot” which would allow the player to attack their opponent with a 3x3 attack. The player would earn this special shot if they went on a “hot streak” by hitting their opponent 3 times in a row. We ended up scrapping these ideas as we wanted to incorporate our other features instead, a scoreboard and sound effects. Along with the above-mentioned features, we also thought about making a feature to where the player can repair their own ships. This would be done if they sink an opponent’s ship and then would get X number of repairs, each repair point is only able to repair one spot on the grid. Players would earn repair points depending on the size of the ship they sunk. For example, if I sunk my opponents 3x1 ship, I would then be able to earn 3 repair points. We scrapped this because we realized we would also have to implement it for the AI and that seems like a daunting task with the amount of time we had left. A sub-feature we had to scrap was having a victory sound once a specific player won. The reason for not going forward with this feature was because the previous team had the game states on a constant loop and the sound would keep playing. We thought about fixing this when implementing AI, but it seemed like a larger and unnecessary task; however, this is one of the reasons it runs slower on specific computers.

1. Retrospective on what the team would have done different:

Our team got all the important work done but we did not work on it as soon as we could. We waited a long while before implementing changes in the code and we could have had everything done way sooner. Although it did not affect our final product, we could have made optimization changes in the backend, allowing us to make the program run better on laptops which could not handle as many operations at once, like the game required. Aside from that, if we could go back, we would have conducted more meetings throughout the duration of our project time. A reason for both of these previously mentioned things was the fact that fall break was in the middle of our time duration and many members of the team were unable to meet during this time. This all could have been solved if we, as a group, set deadlines for ourselves which were clearer so that we knew who was doing what and when they would be done with it. This would allow us to plan more accordingly and allow members who were struggling with their section of code to get help from the other members.