Project 2 Log

Meetings-

Meeting 1

September 27
All members present
In class

- Talked about different ideas to implement
    - Scoreboard
    - GUI
    - Sounds
    - Animations

Meeting 2

September 30
All members present
In class

- What can we implement
    - Sounds for sure
    - Scoreboard for sure
    - Animations probably
    - GUI is questionable
    - maybe special shots
- Talked about how we can implement AI
    - Inherits from player class
- Need to draw diagram for classes
- Talked about dates
    - People would be gone
    - When to have product done
- Verbal confirmation from Gibbons for our implementations

Meeting 3

October 2
Justin, Hunter, Zach, Brandon
In class

- Draw up class diagram so we can all follow the code

Meeting 4

October 7
All members Present
In class

- Talked about changes each person is implementing and how or why they are changing things

-

Meeting 5
October 8
In Lab

- Split up work for the day into different jobs working on Ai and Scoreboard.

Known Issues:-

- **C++ SOUND IN LINUX IS HARD** --yup
- Normal AI segfaults when ships bigger than one are placed on border of map

Changelog:-

- 9/27/19 - Hunter C.
    - **Created:** Board::randPosGen() for generating random legal position values forAi.
    - **Created:** main menu to choose between 1-player and 2-players.
    - **IN PROGRESS(Finished):** Board::setupBoard_AI() to setup board for AI.

- 10/4/19 - Zack K.
    - **Added:** Calls to setupBoard_AI() in Executive.cpp
    - **Created:** AIgame() function in Executive.h
    - **IN PROGRESS:** Adding different difficulty AI functionality to aiGame()

- 10/7/19 - Hunter C.
    - **Fixed:** error with ai setup being private in Board.h
    - **Fixed:** error with ai number of ships not being initialized in Executive.cpp
    - **Created/Finished:** AI setup works for all number of ships
    - **IN PROGRESS(FINISHED):** creating game loop(s) for easy ai game
- 10/7/19 - Justin K.
    - **Added:**
        - Values inside of Executive.cpp that will display Player 2's sunk ships and player 1's sunk ships.
    - **IN PROGRESS:**
        - **DONE:** Need to display proper values inside of executive.cpp showing Player 2's sunk ships and player 1's sunk ships at the end of the game -**Seems to work properly? ZK 10/10/19**

    - **DONE**Adding private member *shipsLeft* to display ships that Player 1/ Player 2 sunk before game over
    - **DONE**Going to try and implement game over screen with both boards displaying where all ships are, which ships were hit, and at which part.
    - **DONE**Planning to add cin player names so that I can write to a file their number of wins for a scoreboard page
  - **ISSUES:**
    - Getting a stacking issue, but only when program ends. Everything else works fine.

- 10/8/19 - Hunter C.
  - **IN PROGRESS(FINISHED):** Working on Hard Ai in Executive.cpp
  - **IN PROGRESS/CONFUSED:** Researching how to include sound in linux
-
- 10/8/19 - Brandon W.
  - **IN PROGRESS**: Working on Orthogonal Shot function for normal AI in Board.cpp
- 10/10/19 - Zack K.
  - **IN PROGRESS:** SetShipsLeft function decrements numberOfShips by 1 when a ship is shown to be sunk in executive.cpp
  - **Completed:** Shipsleft function appears to be operational, but I think it would be better to print it in a different location as it's hard to see. current cout is in the executive *shoot()* function
- 10/16/19 - Brandon W.
  - **Completed**: Normal AI Still need to push it onto github
- 10/17/19 - Brandon W.
  - **Completed**: Medium AI completed and pushed to the master branch

Description on how work was split between teammates

- Work was split by everyone choosing which features they felt most comfortable adding to the game
- Error checking was a whole team effort
- Most work was error checking and bug fixing rather than feature implementation

Challenges and how they were overcome or dealt with

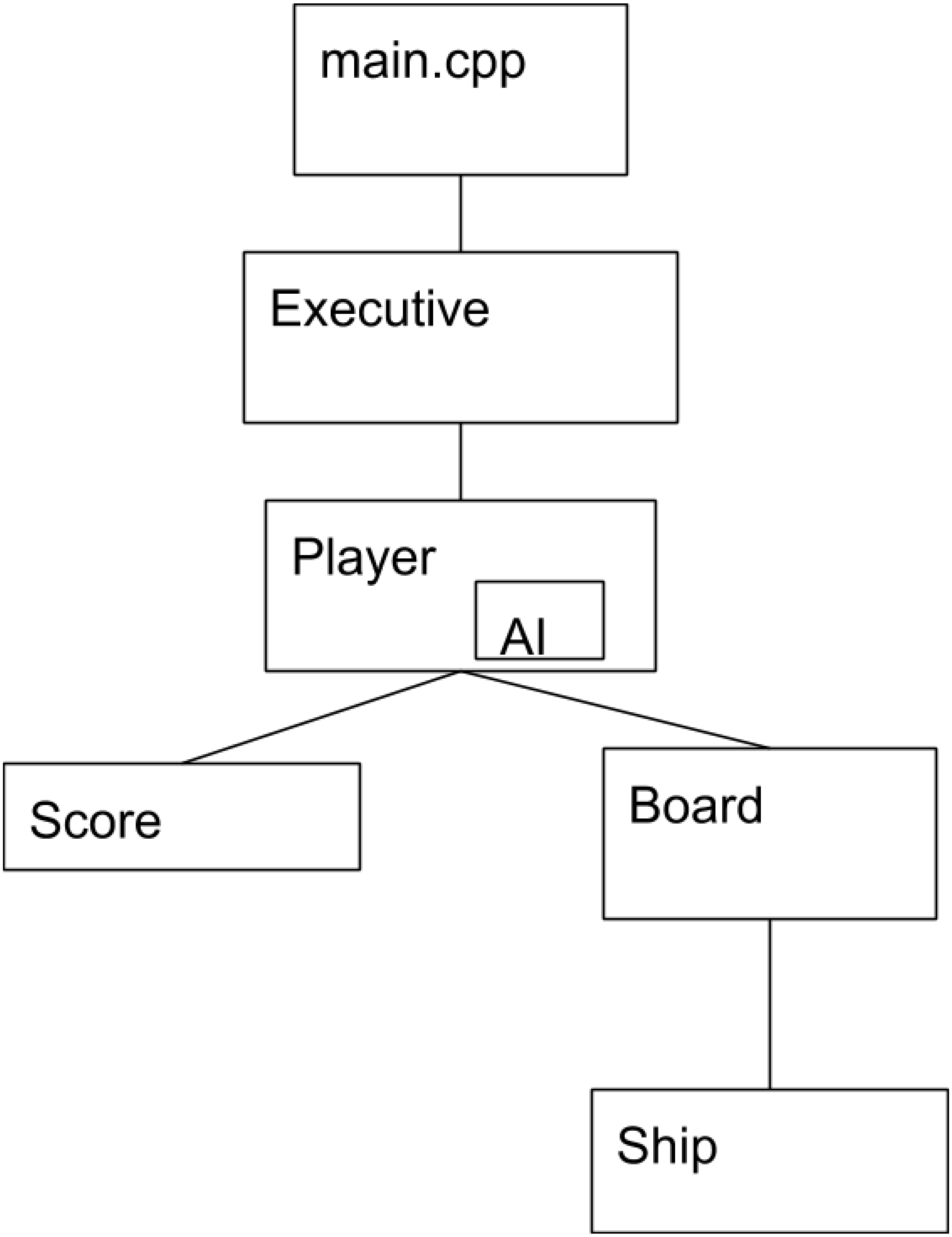
- Segfaults
  - Error checking
  - Keeping track of errors on a known errors log which was shared with everyone in the group
- Understanding and using another groups code implementation

Any features that did not make the demo version

- Sound effects - Too much trouble to get library installed

Retrospective on what the team would have done different

- More Planned team meetings and more direct/centralized communication on errors and layout of project.
    - Planned meetings on paper would be best to fix this.
    - Using software to keep track of changes would help overall group editing of project.
- Better planning of overall project and who will make what class/feature
- Better documentation of all group meetings. A few meetings between not entire team were forgotten to be added to meeting doc.
- Learn to merge more efficiently, so multiple people could work on a single file at the same time
- Work out timeline of work before work is started
- Better organized meetup times, so we can meet up and work on the project together more often


main.cpp
Executive
Player
AI
Score
Board
Ship