CS246
Project RAIINET

Project Breakdown

Wednesday, November 23rd

- [ ] UML (Everyone)
- [ ] Plan of Action (Everyone)

Thursday, November 24th

- [ ] Create a shared repo for the project (Pratham)

Friday, November 25th

- [ ] Create basic command interpreter for interaction (no setup) (Milan)

Saturday, November 26th

- [ ] Create header file for BOARD CLASS (Brashan)
- [ ] Create header file for addText OBSERVER (Pratham)

Sunday, November 27th

- [ ] Implement .cc file for BOARD CLASS (Brashan)
- [ ] Implement .cc file for addText OBSERVER (Pratham)
- [ ] Checkpoint: Be able to display blank 8x8 board (using text-based observer class)
- [ ] Create header file for PLAYER CLASS (Milan)
- [ ] Create header file for LINK CLASS (Pratham)

Monday, November 28th

- [ ] Implement .cc file for PLAYER CLASS (Milan)
- [ ] Implement .cc file for LINK CLASS (Pratham)
- [ ] Checkpoint: Be able to display board with links on it for each player

Tuesday, November 29th

- [ ] Implement mechanics for game movement (Brashan)
- [ ] Implement reading moves from external file (Brashan)
- [ ] Implement game quit (Brashan)

Wednesday, November 30th

- [ ] Implement Setup through command line arguments (Brashan)
- [ ] Implement Server port interaction and moving links off board (Milan)
- [ ] Checkpoint: Run test here, players should be able to move and download data and viruses now

Thursday, December 1st

- [ ] Implement Battle action between links (Pratham)

CS246
Project RAIINET

- [ ] Create header file for the ABILITIES CLASS (5 given + 3 extra) (Brashan & Milan)

Friday, December 2nd

- [ ] Implement .cc file for ABILITIES CLASS (Everyone)

Saturday, December 3rd

- [ ] Incorporate and test ABILITIES (Brashan)
- [ ] Create header file for addGraphics OBSERVER (Milan & Pratham)
- [ ] Implement .cc file for addGraphics OBSERVER (Milan & Pratham)

Sunday, December 4th

- [ ] Perform bulk final test cases (Everyone)
- [ ] Make finishing code changes and debugging (Everyone)
- [ ] Begin final design document (Everyone)

Monday, December 5th

- [ ] Complete majority of final design document (Everyone)

Tuesday, December 6th

- [ ] Finish up final design document (Everyone)
- [ ] Update UML diagram (Pratham & Brashan)

CS246
Project RAIINET

Questions

Question 1:
In this project, we ask you to implement a single display that flips between player 1 and player 2 as turns are switched. How would you change your code to instead have two displays, one of which is player 1's view, and the other of which is player 2's?

Answer 1:
The way we implemented the display's currently, is that we have an observer, either text or graphics based which takes in a player object as its subject. Then using the subject's information on the board's current situation (link locations, abilities, etc) it displays the board for that player, whether this be player one or player two. In this scenario where we are asked to have two displays, we would have two observers open at the same time, either text or graphics based. And each observer would be fed one of the two players into it. Then, everytime one of the players makes a move, the notify method would be called and both observers would be updated with the relative information.

Question 2:
How can you design a general framework that makes adding abilities easy? To demonstrate your thought process for this question, you are required to add in three new abilities to your final game. One of these may be similar to one of the already present abilities, and the other two should introduce some novel, challenging mechanic.

Answer:
The three new abilities will be included in the final project.

Question 3:
One could conceivably extend the game of RAIInet to be a four player game by making the board a plus shape (formed by the union of two 10x8 rectangles) and allowing links to escape off the edge belonging to the opponent directly adjacent to them. Upon being eliminated by downloading four viruses, each links and firewall controlled by that player would be removed, and their server ports would become normal squares. What changes could you make in your code to handle this change to the game? Would it be possible to have this mode in addition to the two-player mode with minimal additional work?

Answer 3:
It is definitely possible to add a four player mode in addition to the two player mode with minimal additional work. In order to implement a four player game mode to the code, we'd first have to manipulate our player class to add a boolean variable, alive, and a method, isAlive(), to determine if said player is alive or not and if their abilities/pieces/serverports should be on the board. In addition, we'd have to alter the vector of vector's size to accommodate the extra board space (increasing the vector size). Finally, we'd have to keep track of the player that is directly adjacent to them in the player class. The reason we have to do this is because a player can only allow links to escape off the edge belonging to the opponent directly adjacent to them. Meaning, if a player were to go off the edge of an opponent, that is not adjacent to them, it would cause an undefined behavior.