

Poker game

(Sprint3 Planning Document)

<https://github.com/srhee91/PokerGame>

Team 3

So Mi Choi, Bo Heon Jeong, Hanchen Li,
Sang Rhee, Yixin Wang, Yuying Wang

Sprint3 Planning Document

1. User stories to be implemented

1. As a user, I would like the GUI elements for *OngoingMode* to update in response to my actions as well as those of the other players.
2. As a player, I would like a time limit of 30 seconds per turn imposed on all players before forcing them to fold to limit the amount of time players have to wait between turns. I would like the time remaining to be displayed onscreen during my turn.
3. As a player in an ongoing game, I would like to be able to quit the game through an onscreen button and return to the “startup” screen. I would like the host to be able to handle players quitting or their connection to the game ending.
4. As a player, I would like the game to be void of major game-breaking bugs.

2. Description of tasks

1. Integrate the already-existing *OngoingMode* animations to correctly respond to received *GameState* from the host to show each player’s actions. (User story 1)

(Yixin Wang, 20 hrs) (Sang Rhee, 10 hrs)
2. Expand the *GameState* class to include each player’s most recent action so that the *OngoingMode* GUI can accurately display status labels for each player. (User story 1)

(Sang Rhee, 20 hrs) (Bo Heon Jeong, 10 hrs)
3. Add a timer in *OngoingMode* that starts when a player’s turn starts so the GUI can show how much time that player has remaining to make a decision. If the timer expires for the controlling player’s turn, the GUI will execute the fold action automatically. (User story 2)

(Yixin Wang, 10 hrs)
4. Add a Quit button to the *OngoingMode* GUI that takes the player back to *StartMode*. (User story 3)

(Yixin Wang, 2 hrs)
5. Improve the host to be able to handle disconnecting clients (either from willingly quitting or from network errors) and to update the *GameSystem* accordingly, removing that player from the game. (User story 3)

(Yuying Wang, 10 hrs) (Hanchen Li, 30 hrs)

6. Debug the Rank class through adding test cases that handle all possibilities of poker-hands in the *TestRank* class. (User story 4)

(SoMi Choi, 20 hrs) (Bo Heon Jeong, 10 hrs)

7. Debug the GUI portion of the game by play-testing the game with all possible user inputs. (User story 4)

(Yixin Wang, 6 hrs) (Sang Rhee, 6 hrs) (Yuying Wang, 6 hrs) (SoMi Choi, 6 hrs)
(Bo Heon Jeong, 6 hrs)

8. Debug the *GameSystem* portion of the game by play-testing the game with all possible user inputs. (User story 4)

(Yixin Wang, 6 hrs) (Sang Rhee, 6 hrs) (Yuying Wang, 6 hrs) (SoMi Choi, 6 hrs)
(Bo Heon Jeong, 6 hrs)

9. Debug the Network (*ClientMessageHandler*, *HostMessageHandler*, *HostBroadcaster*, *HostSearcher*) portion of the game by simulating network issues and disconnects and host crashing. (User story 4)

(Yuying Wang, 20 hrs) (Hanchen Li, 10 hrs)