CBL Project Poker:

Initial Plan:

1. Project Setup and Git Workflow:
    - Learn about Git and how it works (https://www.w3schools.com/git/)
    - Initialize git repository
    - Create project structure
    - Write a basic README once finished to show the repo and what was done
    - Set up Git branching strategy
2. Basic game and logic and card handling:
    - Card and Deck class: Create the classes for Cards and Deck
    - Procedures for shuffling and dealing with the cards
    - Hand evaluation: Implement logic to evaluate poker hand. Create the methods that are going to help in identifying the power of the hand
    - Player class that has all the attributes of the player
    - Bot class that inherits player and has also extra methods for decision making (raising, calling, bluffing, etc.)
    - Wallet class to store information about the players amount of money and how they can be handled
    - Pot class to store information about the money flow during the game
3. User Interface Setup (Swing):
    - Labels for displaying cards and use the images for the UI (can use the assets from: https://code.google.com/archive/p/vector-playing-cards/downloads)
    - Buttons for player (raise, check, fold, all in)
    - Text areas and labels for cards and buttons
    - Game menu class to set up the game or read the rules of the game
    - Game set up class to set up wallet amount and blinds
    - Poker mat class to actually show the game on the screen
4. Game theory for the bot:
    - Implement decision making system based on hand strength
    - Implement statistics to help the bot make choices based on cards remaining in the deck
    - Implement simple bluffing (random chance for bot to bluff)
    - Used: https://upswingpoker.com/pot-odds-step-by-step/
5. Gameplay loop:
    - Betting rounds: Create the full game mode with the betting rounds the player and the bot can play.
    - Round class for each round played. Every time declaring new deck, shuffling the cards and creating the betting rounds. The class works correctly if the player initializes everything after clicking set up game.
6. Testing basic game flow
    - Test if all the functions of the program work
7. Playtest – Full game test
    - Play the full game to see if it works
8. Fix bugs
    - Identify any bugs and fix them
9. Final updates and UI polishing
    - Finish with last details that can make the game better

Important topics of choice of the CBL (also mentioned before):

1. Learn about Git and all its features and how to use them so that we have good implementation of our program with versioning and all its parts
2. Develop the advanced game theory of poker bot

How Git is shown:

1. We initialized a repository on the computer and synced it through GitHub
2. Created different branches when it came up to versioning
3. Committed all the changes we were doing to the files
4. Finally pushed the changes to the repo so that the teammate could get them by pulling

Extra resources:

- https://stackoverflow.com
- https://git-scm.com/

How game theory for bot is developed:

1. Research on the internet for different formulas and randomization procedures so that it is unpredictable
2. Website for explanations: https://upswingpoker.com/pot-odds-step-by-step/

How game features can be demoed:

- Raise: when the player raises a certain amount and it is added to the pot and bot either calls, fold or calls and reraise then it works.
- Call: when you call the bet of the pot (e.g. when the bot raises) and the money is added to the pot and betting round is finished.
- Check: When you press the button check for basically not adding anything to the pot. You can see that it can only be done at the beginning of each betting stage.
- Fold: when you can fold and the round is over with the bot being the winner
- Blinds: blinds are correctly added to the pot and subtracted from players wallet.

- New round button: When a round is finished new round button becomes available to start a new round. Correctly starts a new round with a new deck and current wallet amounts.
- Bot decision: test different scenarios and you can see when the bot calls or raises or even bluffs at the end of the round when the cards appear. It can also be seen that the bot does not fold easily.

Christos Theofanous (Student ID: 2244411)

Leo Fischer (Student ID: 2275821)