Project Title: Joker

Team name: Jokers

Members:

Chen Huang: chen_huang@ucsb.edu ← The alpha
Zeyu(Harry) He: zeyuhe@ucsb.edu
Zhijun Yan: zhijunyan@ucsb.edu
Nan Wang: nwang01@ucsb.edu
Shida Sheng: shidasheng@ucsb.edu

What the project is about?

-> What problem the project is solving?

Everyone needs entertainment and gambling is a classic way to reduce stress, but Las Vegas is way too far. Why don’t we gambling on mobile devices. So, we decided to make a single casino game that can be played on PC. Since we don’t involved in real world money, we create an easier, faster, safer way to gambling at home, at work, at toilet, etc..

-> Why the problem is important?

Since nowadays the mental health is of crucial importance and being bored and stressed out can lead to unwanted effects, this kind of online games can make people cheerful and in good mental health condition.

-> The outcome of the project

A recently very popular casino game, BlackJack. It is a 2D game that can be played on mobile devices with vivid vision and sound. Every player is automatically assigned an account with 5000 dollars and can play the game. The money he or she gains or loses can automatically be calculated by the system and appears on his or her account. In this way, players can enjoy this game with fictitious currency at home at any time.
Milestone

stage 1: implement Black Jack game → python (2nd, 3rd week)
   a. the logic of the game
   b. bet and win money calculator
   c. player's account building
stage 2: visualize the game → pygame (4th, 5th week)
   a. build the user interface that can get input from the user by graph instead of
      command line
   b. visualize different kinds of cards
   c. visualize what cards are at hand and the fictitious money
   d. show the player profile (money remains, profile picture)
stage 3: improve the visualization(adding sound and motion) → pygame(6th week)
   a. add the motion of the card (drawing the cards and open the cards)
   b. add the sound of the game (background sound, hit sound, open a card sound,
      winning sound, losing sound)
   c. s
   d. c
   e. c

possible stage 4: try to achieve multi-players and networking
possible stage 5: try to use the classes for Black Jack to implement other card games.

Process

stage 1: Implement Black Jack game → python(3rd, 4th week)
   a. The whole logic of the game is built
b. The game can run successfully

stage 2: working on new features according to user stories(5th,6th week)

a. Test for each new feature
b. Working on card visualization
c. Revise code and give more useful information while player playing games

Use case

1. As a player, I want to login in this game so I can start to play the game on my mobile devices.
2. As a player, I already logged in to my account, I go to the main menu directly where I can choose to check my balance to see how much money I remain or start a new game.
3. As a player, I can check my balance by input in the terminal and then input "start game" instruction and the game can get started.
4. As a player, as I start a new game, I can decide how much money to put in the next game when I already know my balance so that the next game can get started.
   a. Task 1: Create a basic program that accepts user input. Prints start the game when user inputs 'start'.
      i. Estimated time: 1 day
      ii. Who is assigned to this: alex
      iii. Go right test: the game will ask: how much you want to bet
           the player type in a number from 1 to the money he has
           the game will say: ok you bet ___ and game started
      iv. Exceptional test: the player type a number out of range or other words
           the game will say: please type a number from 1 to ___
5. As a player, when the game starts, I want to see what card I have and what card the dealer have so that I can think about if I need to draw another card.
   i. Estimated time: 1 day
   ii. Who is assigned to this:
   iii. Go right test:
      the dealer draws a card [***,***]
      you draw a card [diamond, 5]
      the dealer draws a card [diamond, 8]
      you draw a card [heart, 6]
      now you have point 14 in total
   iv. Exceptional test:
6. As a player, after the game starts, I want to decide whether or not to draw another card so that I can let the sum of my cards close to 21
   i. Estimated time: 1 day
   ii. Who is assigned to this: Harry
   iii. Go right test: the terminal will output an prompt: “Do you want to hit or stand?” then the user will input: “hit” or “stand”. If “hit” is the input then we print the new random card on the terminal: “The new card you draw is: ___”
   iv. Exceptional test: If the user input something else other than the two input, the terminal will print out "please specify do you want to hit or stand"
7. As a player, I want to see what new cards I draw and what is the total number I got so that I can decide if I need to draw another card.
   i. Estimated time: one day
   ii. Who is assigned to this:
iii. Go right test: After you type "Hit"(or choose to draw a new card), the game will show the information of new card. It will automatically show the total points of cards you have. If points are over 21, game over. And go to step 8. If points are close to 21, the game tells you whether you want to draw another card or not (go back to step 6).

iv. Exceptional test:

8. As a player, after a round of a game, I want to see if I win or lose and how much money I win so that I know how much money I have right now.
   i. Estimated time: 1 day
   ii. Who is assigned to this: Kenneth
   iii. Go right test: Dealer show card, if total number is smaller than 17, dealer automatically draw another card. And show all the cards on table(dealer and player); show result and then the game shows win or lose and followed by the statement that shows the money you win from the game and also the total money you have currently.
   iv. Exceptional test: If total is 0, kick out or “Do you want more token(chip)?”
   v. As a player, I can input in terminal “RESTART” and return to step 4 to decide how much money to bet.
   vi. Estimated time: 1 day
   vii. Who is assigned to this:
   viii. Go right test: a player can whenever input “RESTART” on the terminal, then the current game will terminate and the computer will ask “how much you want to bet”, which means that the game progress restart from the beginning(back to step 4)
   ix. Exceptional test: the player type things other than “RESTART”, the game will not restart. If the word the user typed can engine other command, then other command will

9. As a player, I can input in terminal “RESTART” and return to step 4 to decide how much money to bet.
   i. Estimated time: 1 day
   ii. Who is assigned to this: Nan
   iii. Go right test: a player can whenever input “RESTART” on the terminal, then the current game will terminate and the computer will ask “how much you want to bet”, which means that the game progress restart from the beginning(back to step 4)
   iv. Exceptional test: the player type things other than “RESTART”, the game will not restart.

10. As a player, if in step 9 I choose don’t make any further bet or enter "MAIN MENU", then we will return to the main menu, where you can choose to check your total remaining balance.

11. As a player, I want to stop the game anytime I want because sometimes I may have other things that prior to the game to do.

12. As a player, I want to resume the game so that I can pick up where I left last time.

13. As a player, I want to see what card I draw because this can offer me a real game feeling.

14. As a player, I want all my game data in one account can be reserved online because in this way anytime when I log in this account on any devices, I can continue my gaming process.

Necessary link:

github:
https://github.com/Alexxx411/cs48_jocker
slack: https://joker-nfm5441.slack.com/
trello: https://trello.com/b/ie5HjcwH/jocker
High-level Architecture Diagram.

USER

terminal I/O

Vision

Call

Game

Search Information

database

Fetch Information

Player

Continue

Team: Joker.
Member: Alex Huang
Harry He
Kenneth Yan
Nan Wang
Shida Shi
User
 request input

Login
 login info
 successful or not
 save info after
 quit game

check_account()

give account info (if password
 works and it exists)

store the account
 balance to the
 Data Base