De La Salle University- Manila
Gokongwei College of Engineering

LBYCPA1
Programming Logic and Design Laboratory

Project Proposal

"Blackjack"

Percival Q. Macaranas III
Shawn Karlsten A. Narvaja
Franco Miguel C. Ong

Project Description

The project aims to create a blackjack game using the Python programming language. However, the blackjack game will be non-traditional in the sense that the limit can be adjusted to determine its difficulty. For instance, instead of the traditional limit of 21 in blackjack, the user can input a limit of 12 to make it more challenging or 30 to make it easier. The problem to be solved is the development of a program that can accurately simulate a game of blackjack, including dealing cards, calculating scores, handling bets, and determining the winner.

Technical Objectives of the Project:

- Implement a card deck that can shuffle and deal cards.
- Develop a system to calculate scores based on the values of the cards dealt.
- Develop a system to calculate the total money the player has across multiple games.
- Develop a system to determine the winner of the game based on the scores of both player and dealer.
- Develop a system to determine what difficulty the player wants to set the blackjack to be.

The following project will be implemented by breaking down each technical objective into sub-functions and writing code to complete each task. This is done to also balance out the workload for the members in the group. Throughout the project, testing will be conducted to ensure that each component is functioning correctly and integrated properly into the larger program.

IPO

INPUT	**PROCESS**	**OUTPUT**
● Player's hand ● Player's bet ● Player's chosen difficulty ● Dealer's face-up card	***Setting the Difficulty*** - The difficulty will be determined based on what the player wants to set the limit to be. For instance, instead of the traditional limit of 21 in blackjack, the user can input it as 12 to make it harder or 30 to make it easier. ***Betting on the Game*** - If the user wins in	● The player's final hand value ● Whether the player won, lost, or tied with the dealer ● The player's total money

	blackjack, the corresponding bet they placed will be doubled. - If the user loses in blackjack, the corresponding bet they placed will be lost. ***Playing the Game*** - The player must decide whether to "hit" (request another card) or "stand" (keep their current hand). - The player must evaluate the likelihood of busting (exceeding 21) if they hit. - The player must evaluate the likelihood of beating the dealer's hand if they stand**.** - If the user inputs “Hit”, a card would be added to the corresponding card on the table. - If the user inputs “Stand”, the game stops and the AI opponent will reveal their cards. - If the user’s card total is greater than the AI’s card total and less than or equal to 21, the user wins.	

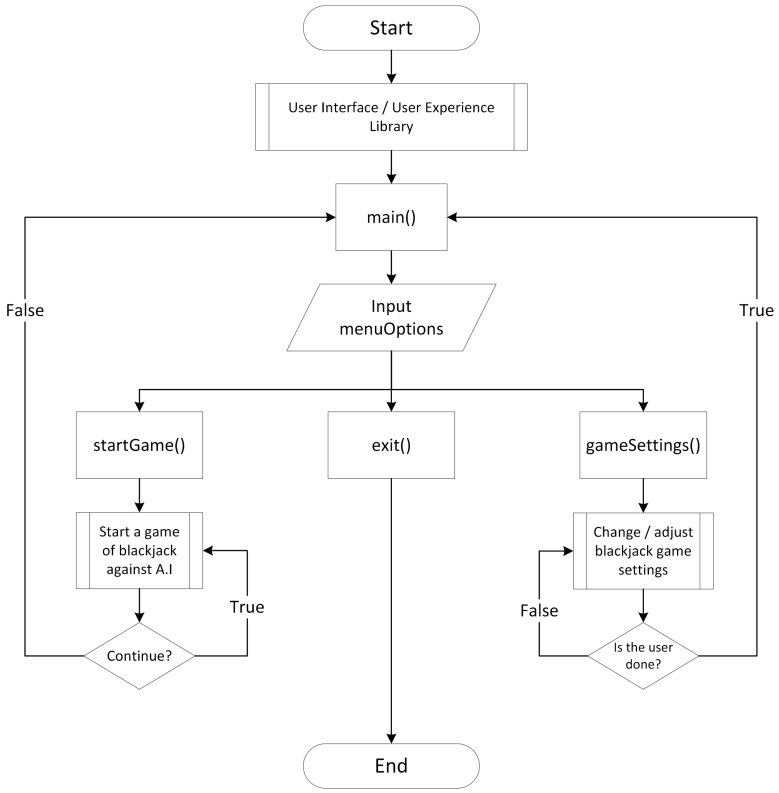
	- If the user's card total is less than the AI's card total and the AI's card total is less than or equal to 21, then user losses.	

Methodology

The project will be written in the Python programming language, this will be the primary foundation and basis of this project.. For its User Interface & User Experience (UI/UX) for presentation and accessibility overall, we will use the PyQt library for creating those elements. For the 52 playing cards art assets, we will use the "Open Source Vector Playing Cards" for the blackjack game overall.

The Python concepts that will be used to develop this project specifically are:

- conditional statements or switch statements for user decision,
- array lists for code optimization in looping,
- declaring multiple functions for the game system such as calculation, etc,.



Schedule of Activities

Task Description	**Task Owner**	**Start Date**	**End Date**
Requirements Gathering	All Members	Mar 17, 2023	Mar 20, 2023
Design UX	Percival	Mar 21, 2023	Mar 25, 2023
Design Game Logic	Shawn	Mar 22, 2023	Mar 27, 2023
Implement Game Logic	Franco	Mar 28, 2023	Apr 2, 2023
Implement UX	Franco	Mar 28, 2023	Apr 2, 2023
Testing and Debugging	Shawn	Apr 3, 2023	Apr 7, 2023
Finalize Deliverables	Percival	Apr 8, 2023	Apr 10, 2023

References

Open Source Vector Playing Cards. Total Nonsense. (n.d.). Retrieved March 17, 2023, from https://totalnonsense.com/open-source-vector-playing-cards/

PyQt download. Riverbank Computing | Download. (n.d.). Retrieved March 17, 2023, from https://riverbankcomputing.com/software/pyqt/download