**Project 2**

**BlackJack Expanded**

Michael Echaluse

CSC-17A: 43396

Professor Lehr

June 8, 2021

**Intro**

The game is Blackjack.

The rules of this game involves the player being dealt initially two cards, the same is done for the dealer.

In order to win, the player must have a card total higher than the dealer while also being below or equal to 21.

The player is given the option to either “hit” or “stand” for a card.

If the player hits, they are given another card that adds to their total in order to be higher than the dealer. If the player is dealt a card that puts them over 21 then they automatically lose.

If the player stands, the dealer will deal themselves cards as long as their total is still below 17. If they deal themselves a total over 21 then the player wins. Once the dealer has dealt enough cards to be over 17 then they will compare their cards to the player.

If the player’s cards are higher than the dealer’s then the player wins.

If the dealer’s cards are higher than the player’s then the dealer wins.

If both the player’s cards and the dealer’s cards are equal then it is a draw.

**Summary**

Project Size: 569 lines

Number of variables: 21

This second project was an improved version of BlackJack from the first project, adding new concepts such as classes, templates and exceptions.

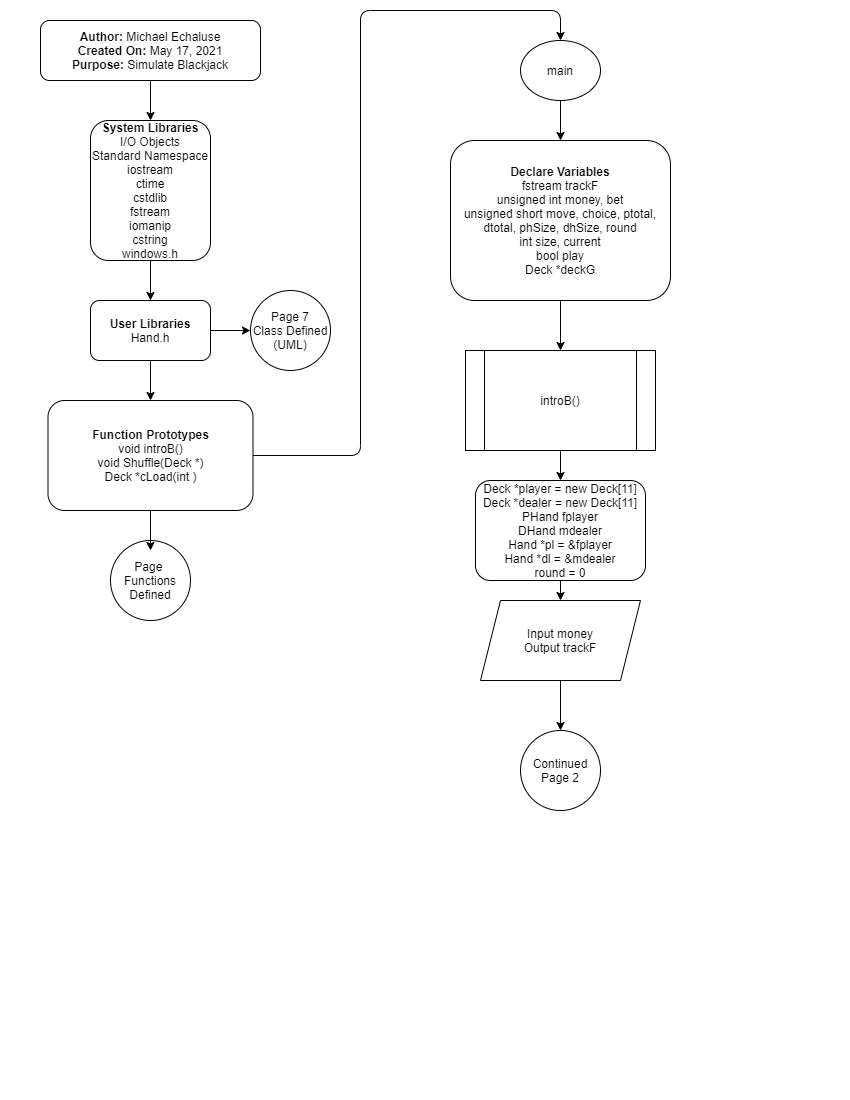
There wasn’t any difficulty with this project as all I needed to do was add the new concepts to certain points of the code such as adding variables that utilized the Hand class which works separately for the dealer and the player through polymorphism. I also added functions utilizing the template system as well as adding an exception near the end of the code.

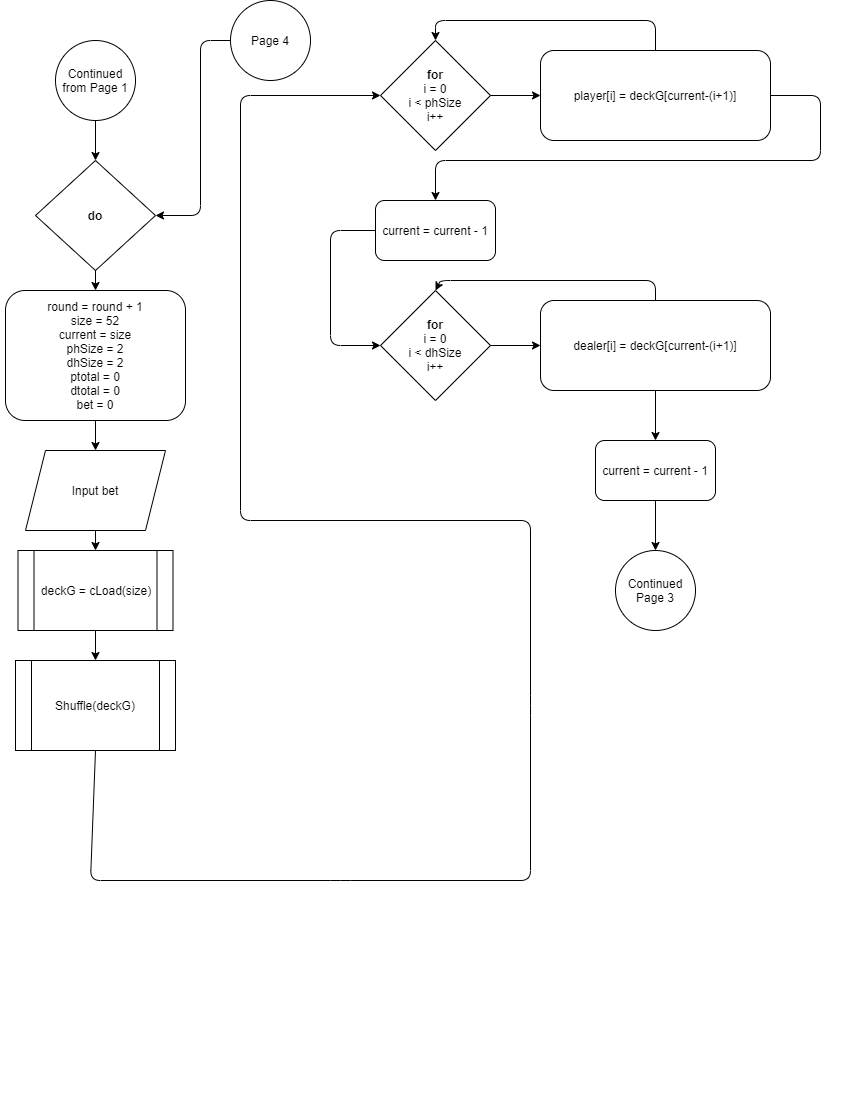
There is not much else to improve upon this code as it functions accordingly as a game of BlackJack. So for this project all that was needed was improving on it using classes to differentiate the player and the dealer.

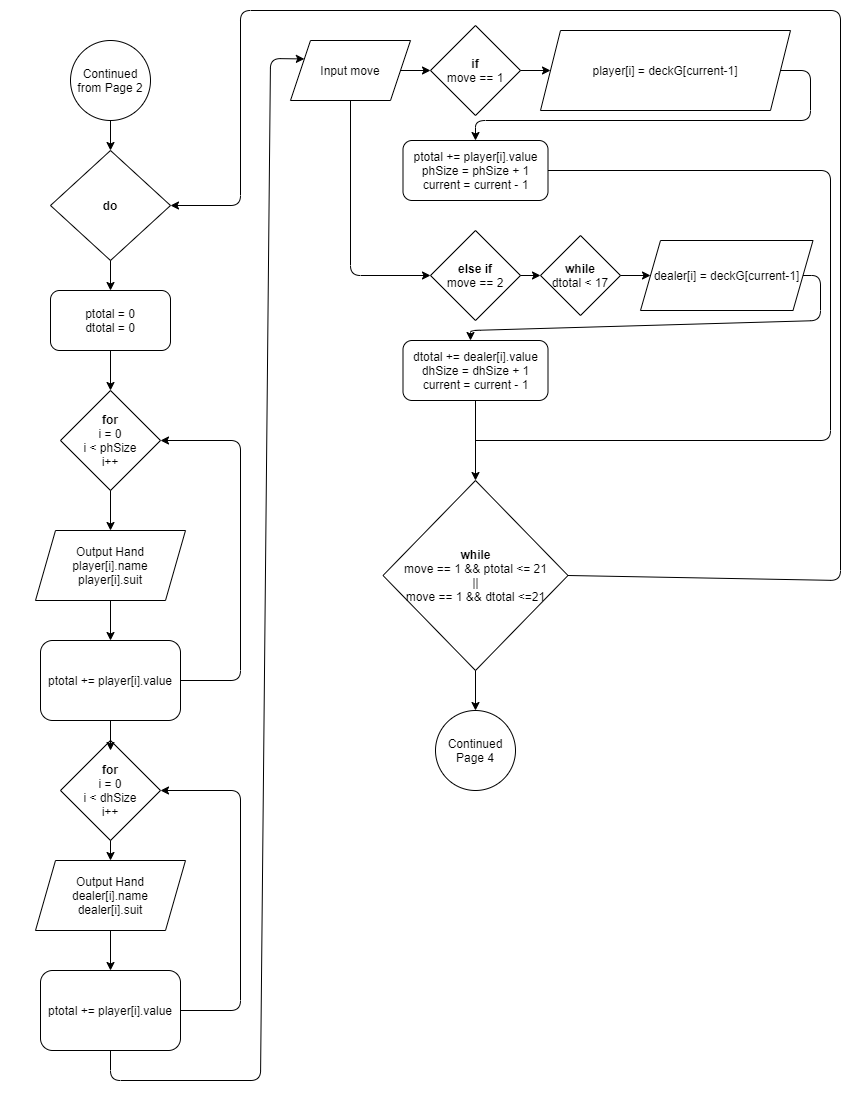
**Description**

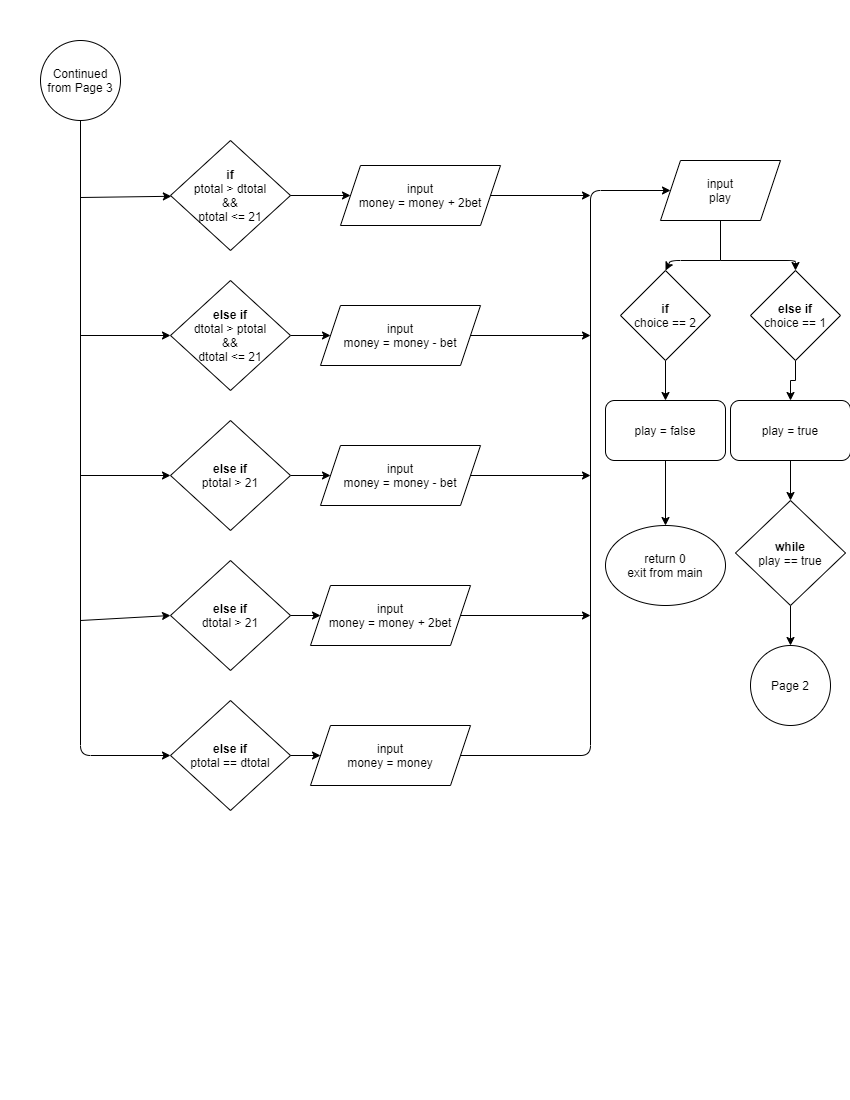
BlackJack Improved On.

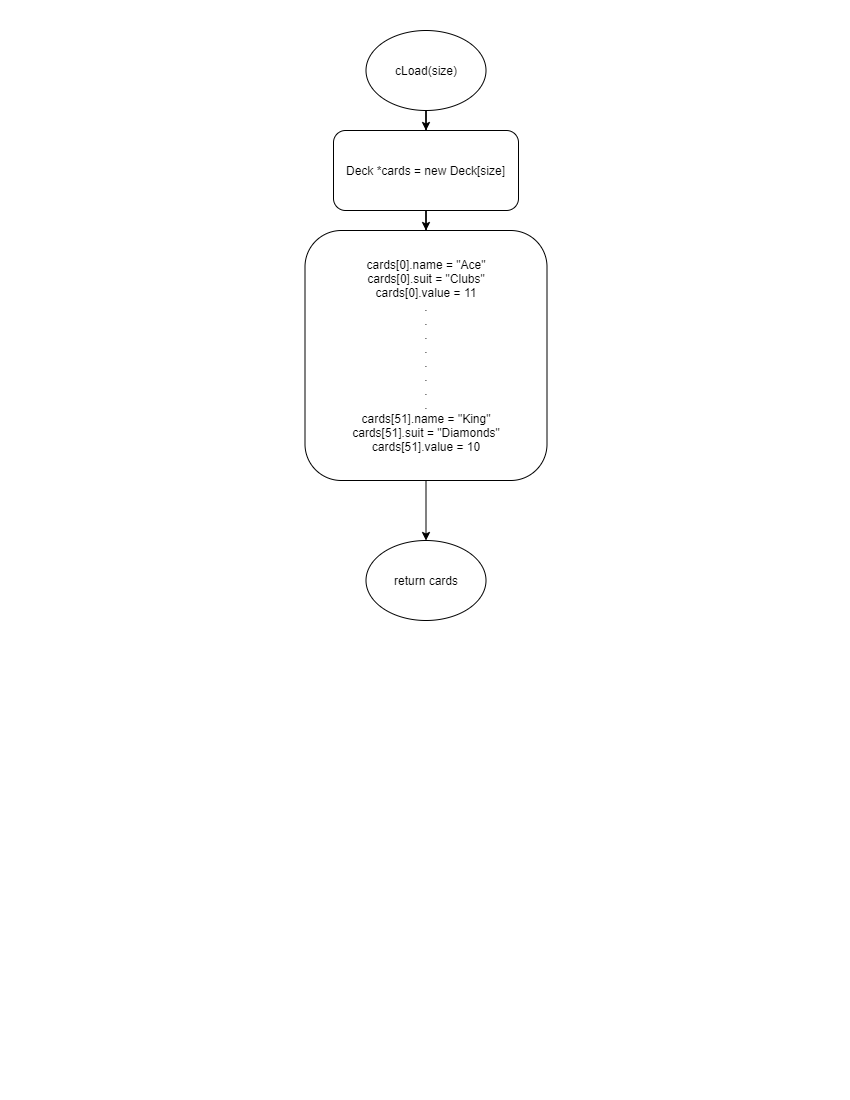
**Flowchart**

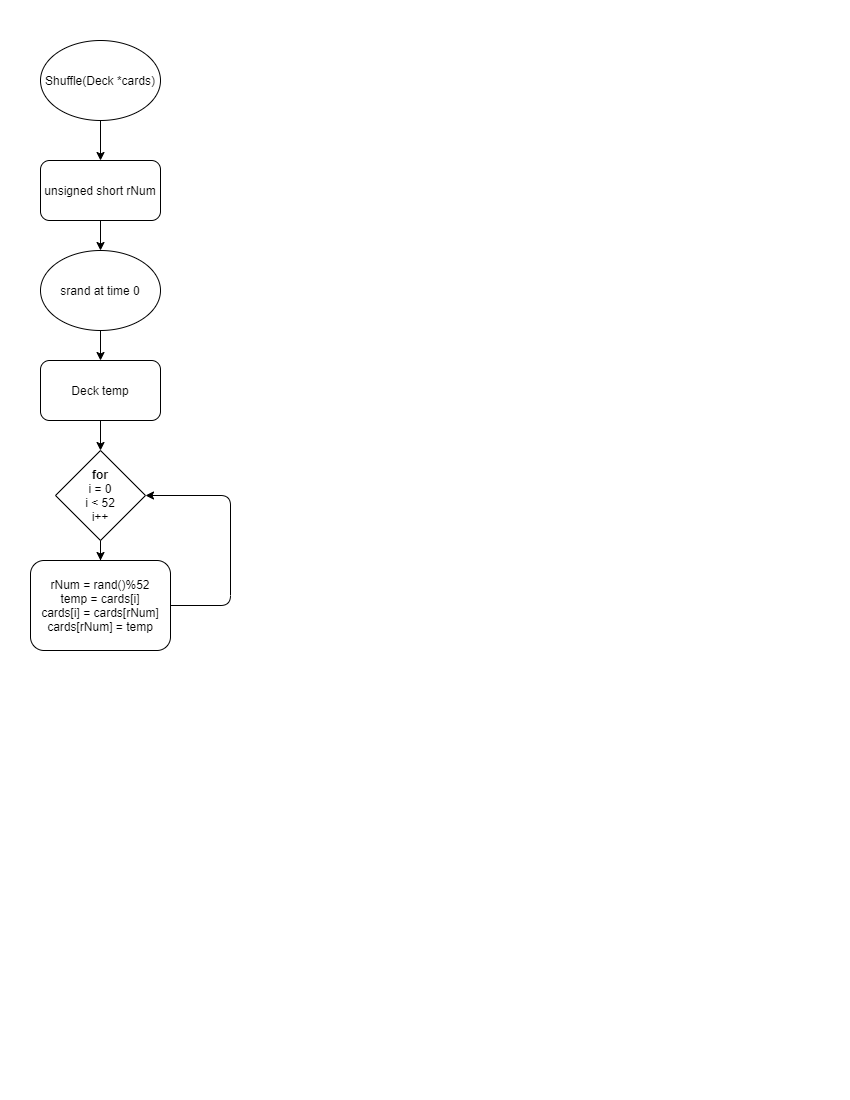




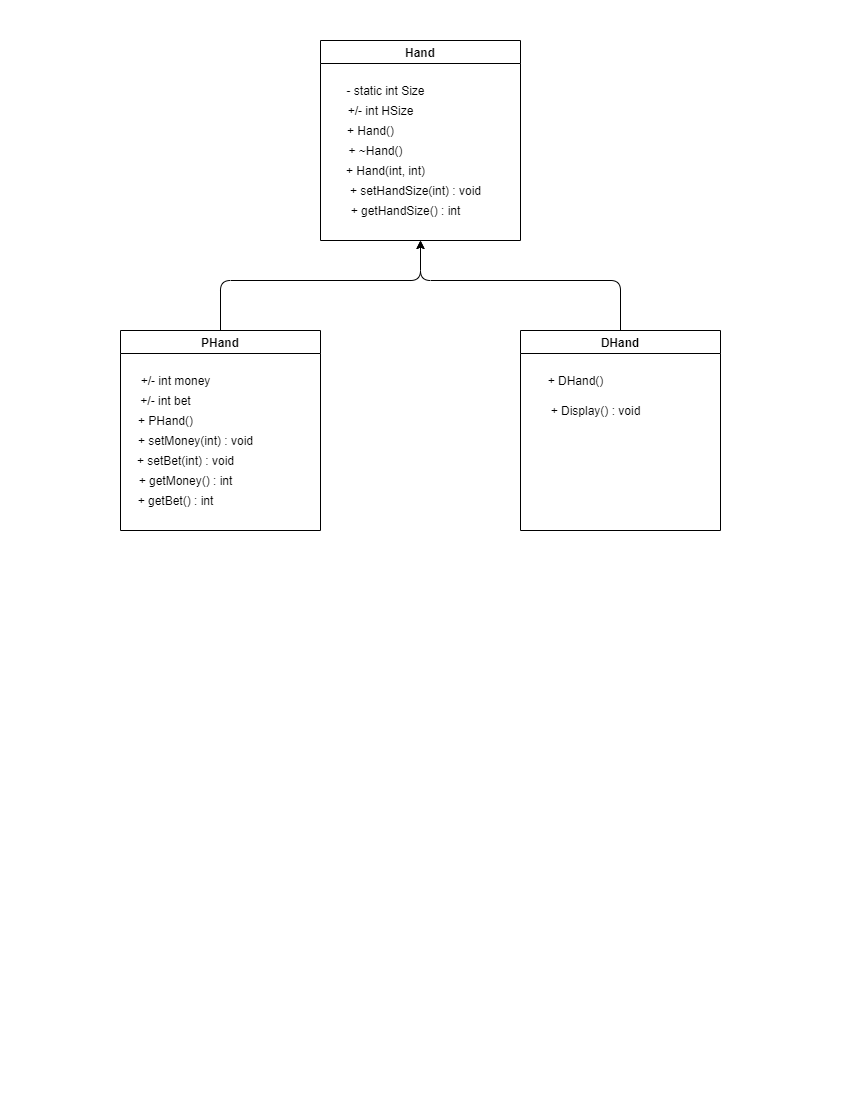








**UML**



**Pseudo Code**

Initialize

Enter Initial Money

do

Enter bet

Call function to load deck

Call function to shuffle deck

Deal 2 cards to player

Deal 2 cards to dealer

Do loop per round

Show player cards

Show dealer cards

Ask to hit or stand

If player total or dealer total is over 21

Break loop

If hit

Deal one card to player

If player total value is over 21

Break loop

Else if stand

Dealer’s full hand is shown

Dealer is dealt cards if total is under 17

While move is hit and player total is below 21 or move is hit and dealer total is below 21

Show hand of player

Show hand of dealer

If player total is higher than dealer total and is below 21

Return twice bet

Else if dealer total is higher than player total and is below 21

Return loss of bet

Else if player total is over 21

Return loss of bet

Else if dealer total is over 21

Return twice bet

Else if player total is equal to dealer total

Return bet

Ask to play again

While play is yes

**Major Variables**

**Integer:**

money: Allows the player to initialize total money

bet: Allows the player to bet an amount of money

size: Initializes the deck size

current: Keeps track of the current size of the deck

**Short**

move: Gives the player the option to hit or stand

choice: Gives the player the option to play again

ptotal: Keeps track of the player’s card total

dtotal: Keeps track of the dealer’s card total

phSize: Keeps track of the player’s current hand size

dhSize: Keeps track of the dealer’s current hand size

round: Used to keep track of how many rounds that player has played

**Bool**

play: Tracks whether the play wants to replay

**Deck: Structure holding parts of a card**

deckG: Holds an entire deck of cards

player: Holds the player’s hand of cards

dealer: Holds the dealer’s hand of cards

**Class Variables:**

fplayer: Derived Hand object that holds hand size of player, money, and bet.

mdealer: Derived Hand object for the dealer’s hand size.

pl: Hand class pointer for fplayer in order to change base class variables through polymorphism.

dl: Hand class pointer for mdealer in order to change base class variables through polymorphism.