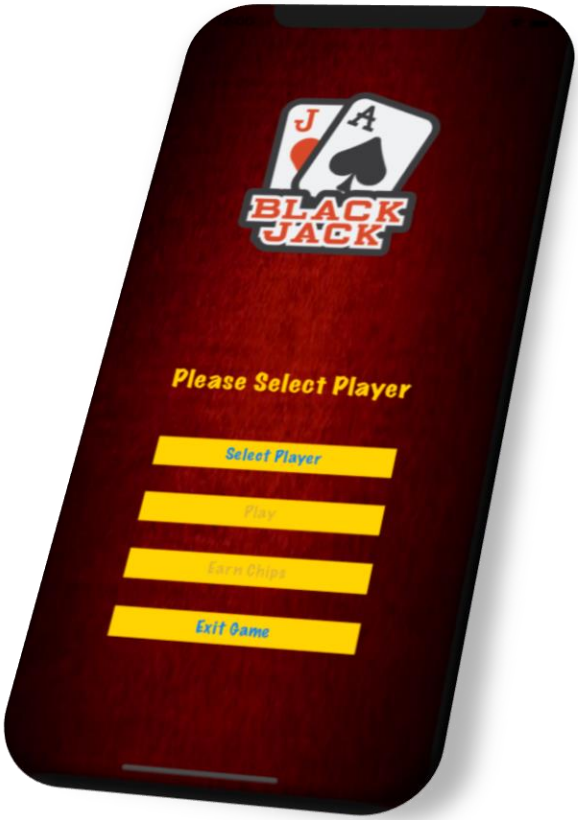


BLACKJACK GAME

Course Final Project

ABSTRACT

My final project is a fully functional blackjack game that incorporates various technologies and techniques to provide - persistent user storage, dynamic animations, and excellent user feedback.

Kyle's PC

COSC 316 iOS Development

Table of Contents

Game Rules

Game

- **Objective** – score higher than the dealer without going over 21!
  - Standard rule notes - Face cards = 10 & Aces = 1 OR 11
- **Place Bet** – Player can select bet (denominations: 5 – 100, by 5's). Once bet is placed, the initial card deal begins.
- **Hit** – Player can add a new card
  - If player count exceeds 21 – game ends with "BUST"
- **Stand** – Dealer card is revealed and playthrough starts
  - Dealer draws new cards until they win or bust – game ends accordingly
  - Player wins if player count is greater than dealer count without exceeding 21

Mini Game

- **Rules** - Players can earn an additional prize by selecting a chip (Prizes: 0, 5, 10)
- **triggers**:
  - When the current game ends
  - When player selects "Earn Chips" button from main menu

Software Architecture (Code)

Notable Implementations

This app contains many notable features such as (and more!):

- **Dynamic card flip animations** – done with recursively generated nested closures
- **UI Feedback** – buttons disable while animations are progressing
- Full **CRUD** Database Operations
- Custom Game Over **Dialogs**
- **Custom TableView** – customized table cells and table actions

Database Layer

- Player Class – Model for player database object mapping
- DatabaseHelper class – reusable helper class to run queries on the database
- PlayersDBHelper class – concrete helper class to run specific functions on players table
- QueryParams & DatabaseVariables – additional objects to support DB helpers

TableView Layer

- Custom table view classes for dynamically generating several specific custom cells at runtime
  - AddButtonTableViewCell
  - SelectButtonTableViewCell
  - CustomPlayerTableViewCell

Controller Layer

- Contains 5 view controllers of varying UI functionality:
  - MainMenu, SelectPlayer, AddPlayer, Game, GameOver
- Navigation between controllers is done via NavigationContoller
- Segues - A selectedPlayer variable is passed between views to keep player information saved between views

Game Layer

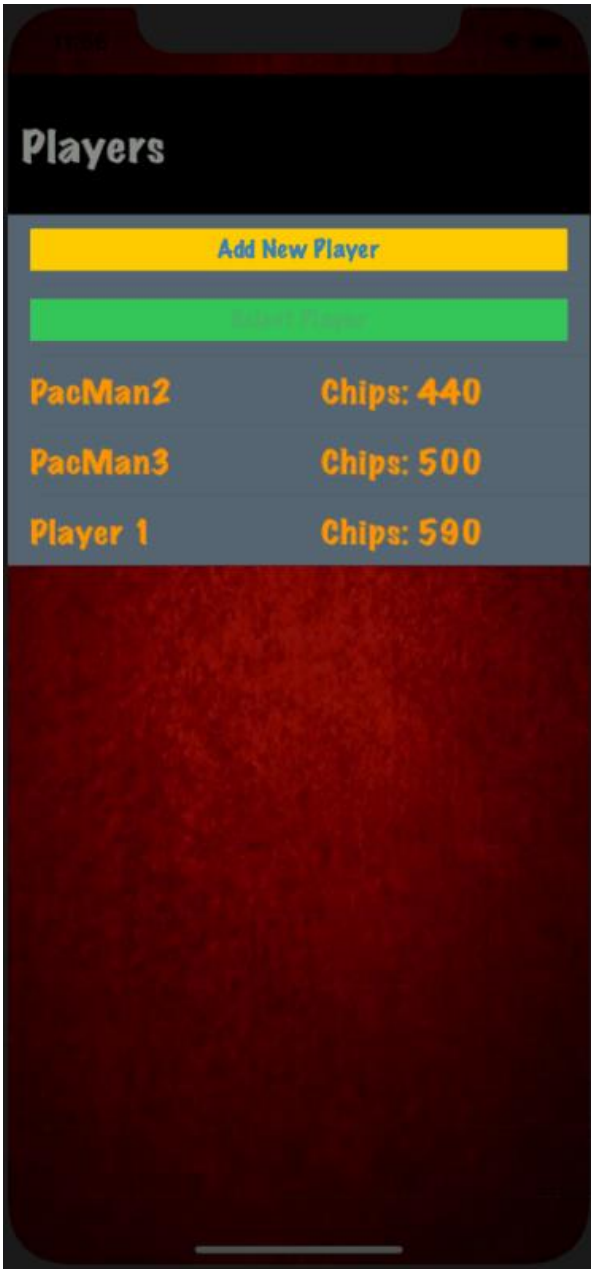
- Card Class – Stores vital information about each card drawn:
  - Metrics – suit, number, value etc
  - Image – image paths of back and front and ref to ImageView
- Game Class – contains most of the abstracted game logic and functionality (the Game View Controller doesn't need to worry about much!)

App Feature Screen Captures (Recordings)

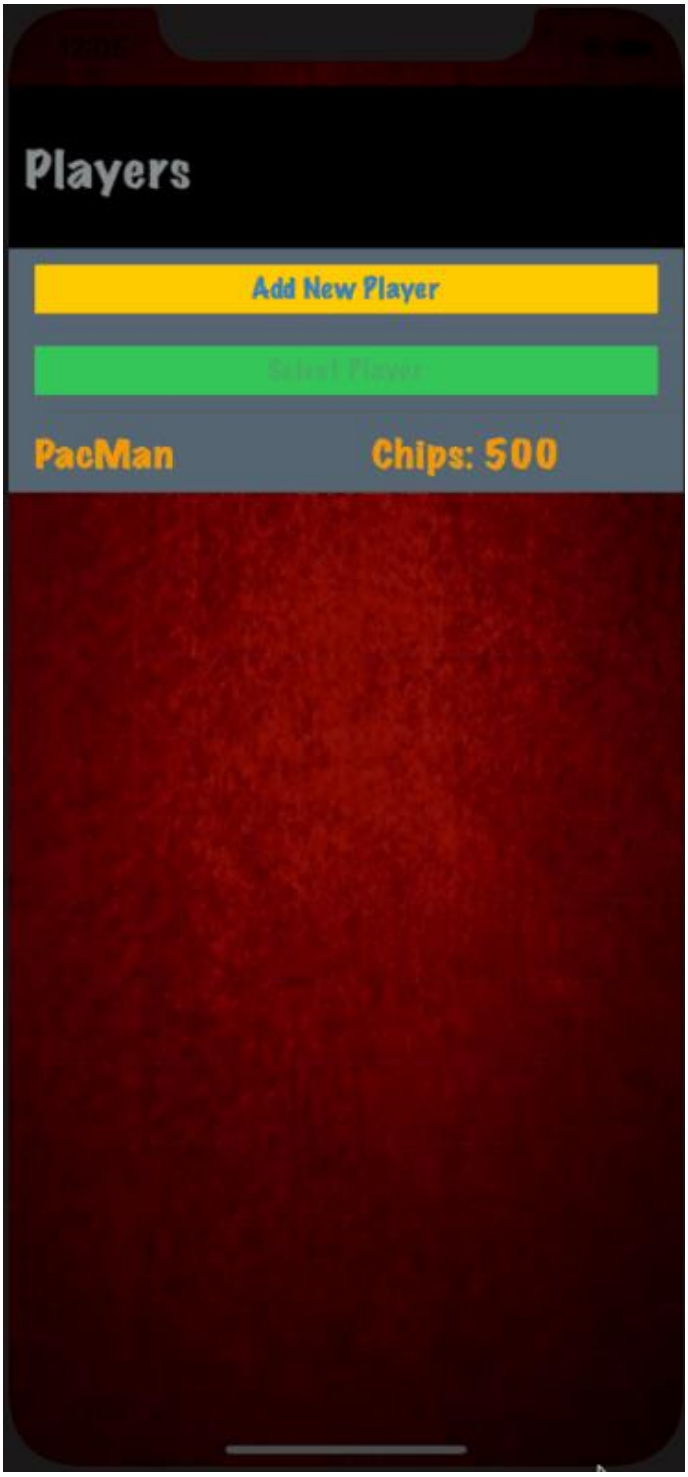
(Enable Editing and click to play)

Add New Player

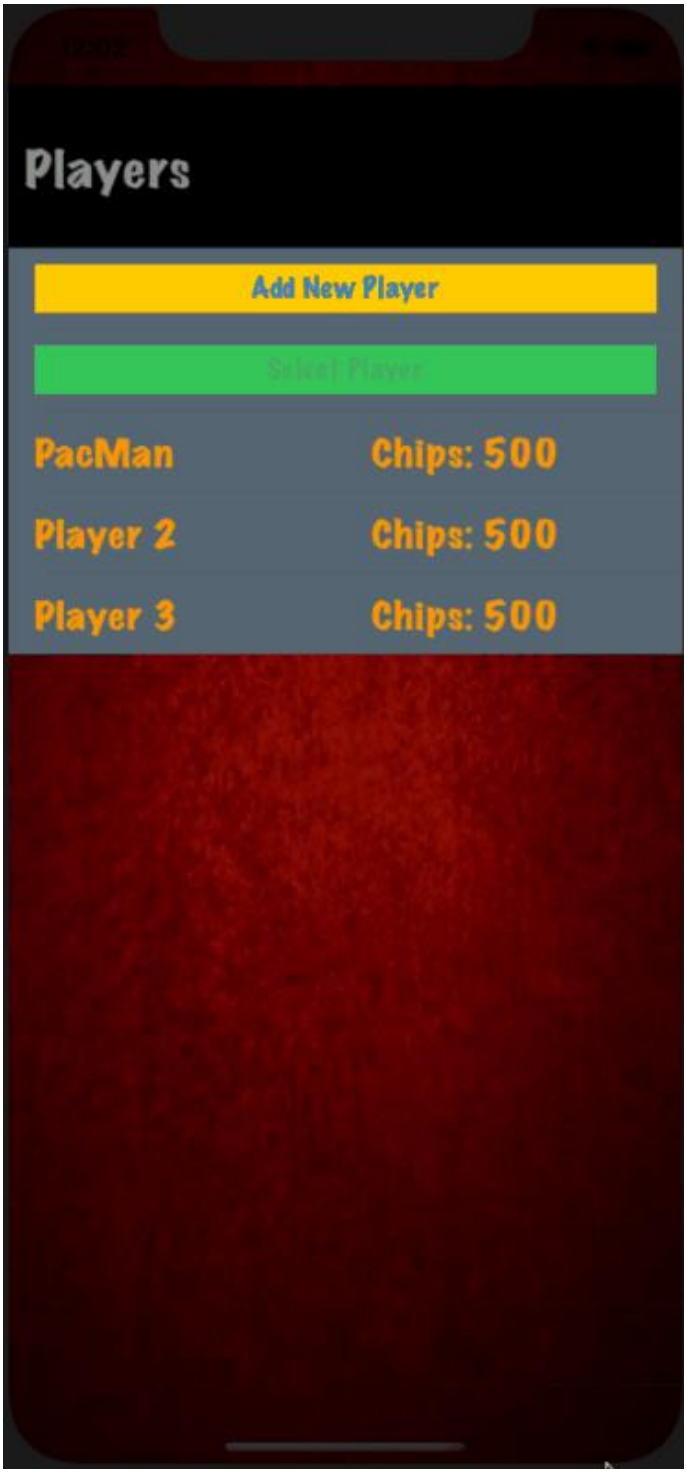
Update Player



Select Player



Delete Player



Game Start

Place Bet And Deal

Hit Player (Recursive Completion Closures)

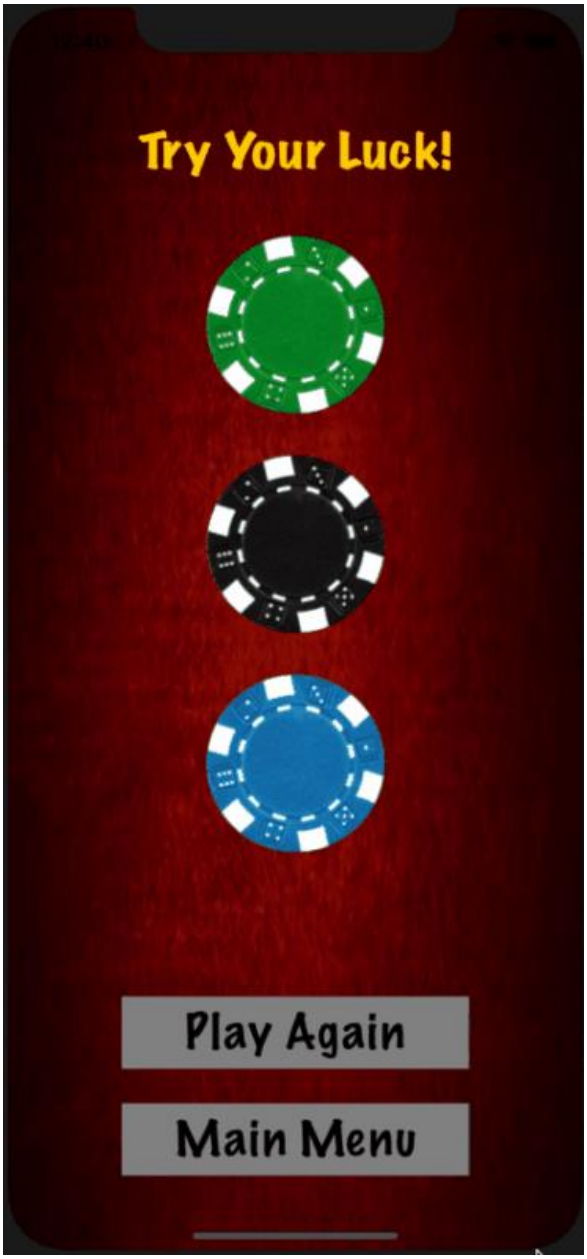
Stand (Recursive Completion Closures)

Game Over Dialog - Bust

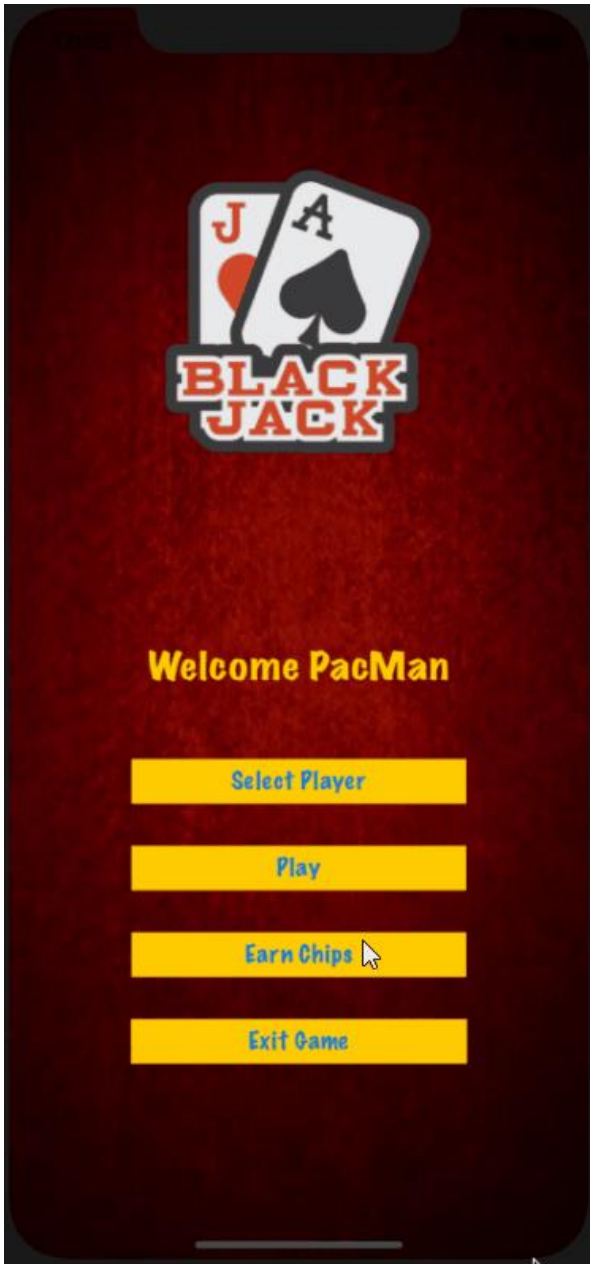
Game Over Dialog - Win

Game Over Dialog - Loss

MiniGame



MiniGame From Menu



Quit Game

Exit App


BLACK
JACK
Welcome PacMan
Select Player
Play
Earn Chips
Exit Game