Background pattern

Description automatically generated**RMIT University**

**COSC2659| iOS Development**

A hammer and pencil on a blue paper

Description automatically generated

Assignment 1

Student name: Tran Xuan Hoang Dat

SID: s3651550

App Name: Guess the picture game App

Professor: Tom Huynh

Date: 01/09/2024

Contents

[Introduction 3](#_Toc176348223)

[Motivation and Inspiration for Guess the Draw: 3](#_Toc176348224)

[Implementation Details 3](#_Toc176348225)

[2.1 Drawing and Guessing Interface 4](#_Toc176348226)

[2.2 Game Progress and Score Tracking 4](#_Toc176348227)

[2.3 Game Center Integration 4](#_Toc176348228)

[2.4 Background Music and Sound Effects 5](#_Toc176348229)

[2.5 Game Over and Menu Navigation 5](#_Toc176348230)

[2.6 Drawing Tools and Customization 6](#_Toc176348231)

[Known Bugs/Problems 6](#_Toc176348232)

[Known Bugs/Problems 6](#_Toc176348233)

[Design Elements and User Experience 6](#_Toc176348234)

[Conclusion 6](#_Toc176348235)

[Reference: 7](#_Toc176348236)

# Introduction

**Overview and Context:** Guess the Draw is a multiplayer game where the rules are simple: one player receives a keyword representing an object, and they must draw a picture for the other player to guess. The goal is to correctly identify the drawing within the time limit.

**Goals:**

* **Educational Resource:** To fulfill all the requirements of the app assignment and learn how to create a game with Xcode from scratch.
* **Play Guide:** One player will receive a keyword and must draw a picture for the other player to guess that word. If the player guesses the correct word, they will become the drawer. Both players have 100 seconds to play the game.

**App Name and Logo Explanation:**

* **App Name:** Guess the Draw the name is straightforward and descriptive, making it clear how the game works.
* **Logo:** I love Shiba dogs, so I chose it as the logo. It also makes users feel happy and relaxed when they open the app.

## **Motivation and Inspiration for Guess the Draw:**

* **Topic Choice:** The inspiration for "Guess the Draw" stems from a love for creativity and the joy of connecting with others through art and imagination. The idea of combining these elements into an engaging and interactive experience motivated the creation of this app.
* **Motivation:** The driving force behind developing "Guess the Draw" was a passion for bringing people together in a fun and creative way, and recognizing the potential for an app that allows users to express themselves through drawing while also challenging their guessing skills.

By offering a platform where users can draw, guess, and interact with each other, "Guess the Draw" aims to spark creativity, foster social connections, and provide a delightful experience that encourages players to engage with art in a playful and meaningful way.

# Implementation Details

Technical Features

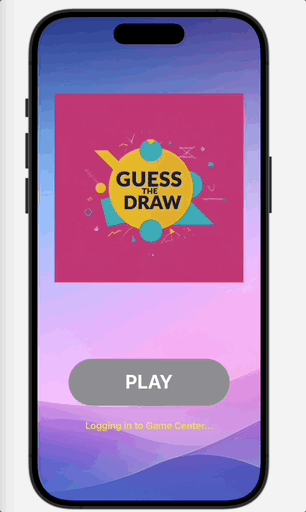
The "Guess the Draw" application is developed using Swift and SwiftUI, optimized for iOS 15. The app not only meets the project requirements but also provides a smooth and engaging user experience.

## 2.1 Drawing and Guessing Interface

**Feature Overview:** The main interface of "Guess the Draw" presents a canvas for drawing and a text input for guessing. Users can switch between drawing and guessing modes, depending on their role in the game.

**Implementation:**

* **UI Components:** Utilizes SwiftUI’s Canvas, TextField, and Button for creating an interactive drawing and guessing experience.
* **State Management:** Manages UI states with @State properties for the current drawing, guesses, and eraser mode.
* **Interactive Logic:** Incorporates
* real-time updates for drawing strokes and processes user guesses instantly.



### 2.2 Game Progress and Score Tracking

**Feature Overview:**  
The Game Progress and Score Tracking feature displays the current score and remaining time, providing players with real-time feedback on their performance.

**Implementation Details:**

* **UI Components:** Uses SwiftUI Text, Label, and ProgressView to show game progress and scores.
* **State Management:** Employs @ObservedObject for managing game state and tracking the player's score across rounds.
* **Timing Logic:** Implements a countdown timer to manage the duration of each round, updating the UI as time progresses.

### 

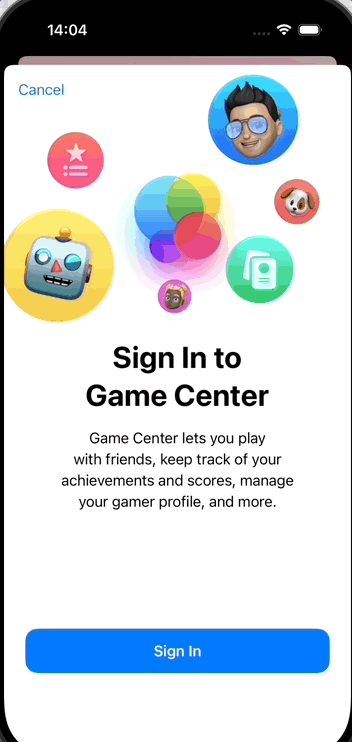
### 2.3 Game Center Integration

**Feature Overview:** The app connects to Game Center to track players' scores on the leaderboard and allow for multiplayer interactions.

**Implementation:**

* **Game Center Integration:** Connecting to Game Center via SwiftUI and GameKit.
* **Leaderboard Tracking:** Displaying players' scores and rankings within the app.

**Known Issue:**

* **Account Access:** The school account used for development does not have access to Game Center features. Testing and verification were performed using a personal account, and the integration was successful.
* 

### 2.4 Background Music and Sound Effects

**Feature Overview:**  
Background music and sound effects enhance the gaming experience by adding audio feedback when users make guesses or complete a round.

**Implementation Details:**

* **UI Components:** Integrates AVAudioPlayer for handling background music and sound effects.
* **Audio Management:** Configures audio loops for background music and triggers sound effects upon specific user actions like submitting a guess or finishing a round.

### 

### 2.5 Game Over and Menu Navigation

**Feature Overview:**  
The Game Over screen provides a summary of the player’s performance, including their final score, and allows them to return to the main menu to start a new game.

**Implementation Details:**

* **UI Components:** Combines VStack, Text, Button, and Image in SwiftUI to create an engaging Game Over screen.
* **State Management:** Resets the game state when the user navigates back to the menu, readying the app for a new game session.

### 2.6 Drawing Tools and Customization

**Feature Overview:**  
The app includes various drawing tools, such as different brush sizes, colors, and an eraser, allowing users to create detailed and creative drawings.

**Implementation Details:**

* **UI Components:** Employs SwiftUI’s Button, ColorPicker, and Slider for tool selection and customization.
* **State Management:** Utilizes @State and @Binding to manage tool settings and ensure the user's choices are reflected in their drawings.

### Known Bugs/Problems

* **Duplicate Navigation Buttons on Drawing Detail View:**
  + **Issue:** The Drawing Detail View currently displays two back buttons, potentially confusing users and cluttering the interface.
  + **Cause:** This issue is likely due to overlapping navigation settings or redundant navigation views within the code.

# Conclusion

Building the "Guess the Draw" app was a highly rewarding experience that significantly enhanced my skills in SwiftUI and iOS development. Throughout the process, I gained valuable insights into state management, data processing, and crafting a user interface that is engaging and intuitive. For future improvements, I plan to optimize the app's performance, introduce new features like additional drawing tools and game modes, and address any existing issues to make the app even more powerful and user-friendly.

Reference

<https://www.youtube.com/watch?v=jujyUk_eyUA&t=170s>