**PLEASE SCROLL TO THE BOTTOM FOR PART 2 DOCUMENTATION**

**A paper with a diagram

Description automatically generated**

**What are we trying to do? Diagram above ^**

Create a basic slide catch game using the imports given and previous pygame skills.

**Break it down: what are the variables?**

Time

Images

Movement of player

Movement of coins

Position in space

Borders of box

Collison of coins and player

**Set up the “coin” class, it will be the football:**

Class Coin:

Initialize with scene:

- Call parent initialization

- Set image to "ball.png"

- Set size to (40, 40)

- Initialize speed range (minSpeed, maxSpeed)

- Call reset method

**Set up how the reset operates:**

Method reset:

- Set y to 20 (top of the screen)

- Randomly set x within screen width

- Randomly set falling speed (dy) within speed range

**Set up the “player” class, it will be Dj Moore:**

Class DjMoore:

Initialize with scene:

- Call parent initialization

- Set image to "Djmoore.png"

- Set size to (100, 100)

- Set initial position

- Set movement speed

**Movement method process (to handle input):**

- If LEFT key is pressed, move left

- If RIGHT key is pressed, move right

Class Game:

Initialize:

- Call parent initialization

- Set background image to "field.png"

- Initialize score and number of coins

- Create DjMoore instance

- Create a list of Coin instances

- (Optional) Initialize a timer for game logic

**Updating position, and check if the coin hit Dj Moore:**

Method process:

- For each coin in the list:

- Update coin position based on dy

- Check collision with DjMoore:

- If collides, reset coin and update score

- Call checkBounds on coin to reset if needed

- Handle DjMoore movement based on input

Function main:

- Create a Game instance

- Start the game

Call main function

**New Documentation:**

**Define class Introduction Screen:**

Define constructor with parameters self and score:

Call parent class constructor

Select scene image ()

Initialize status attribute to be able to quit game

Initialize score attribute with score parameter

Create a Label for instructions:

Set text to be instructions; In my case instructions look like this:

"How to play: Catch as many footballs as you can",

"with D.J. Moore before the time runs out. Good luck!"

Set center position ()

Set size ()

**Create a Label for score:**

Set center position of label ()

Set size ()

Set text to "Previous Score: " followed by the score attribute value

**Create a Button for playing:**

Set center position of button ()

Set text to "Play"

**Create a Button for quitting:**

Set center position of button ()

Set text to "Quit"

**Put score label, instructions label, play button, and quit button into sprites list**

**Define method process to handle button clicks**:

If play button is clicked:

Set status to "play"

Stop the scene – move to screen 2

**If quit button is clicked:**

Set status to "quit"

Stop the scene

Sprites list:

Self.mainscreen

* 1st Image of DJ moore catching a football

self.lblScore

* No image required for score
* Standard sprite
* Its born at the start when its at 0, does not die but updates the to the previous game’s score
* Does not move

self.lblInstructions,

* Text box
* No image

self.btnPlay,

* No image
* Standard sprite
* Born at beginning, when clicked brings you into the game

self.btnQuit

* Same as play button, but instead of bringing you into the game, it exits the program
* Standard

self.DjMoore,

* Image of DJ moore headshot
* User controlled sprite
* Moves along a fixed y axis
* Can only move horiztonally
* When colliding with the footballs, footballs disappear and score goes up

self.coins,

* Image of a football
* Computer controlled sprite
* Dies when touched by user controlled Dj moore sprite

self.lblTime]

* No image
* Controls how long gameplay is

**PHOTO CREDITS**

**Image Credits**

1. DJ Moore headshot photo: Property of NFLPA
2. DJ Moore catching football: Getty Images
3. Football: Getty Images
4. Football field: PNGTree