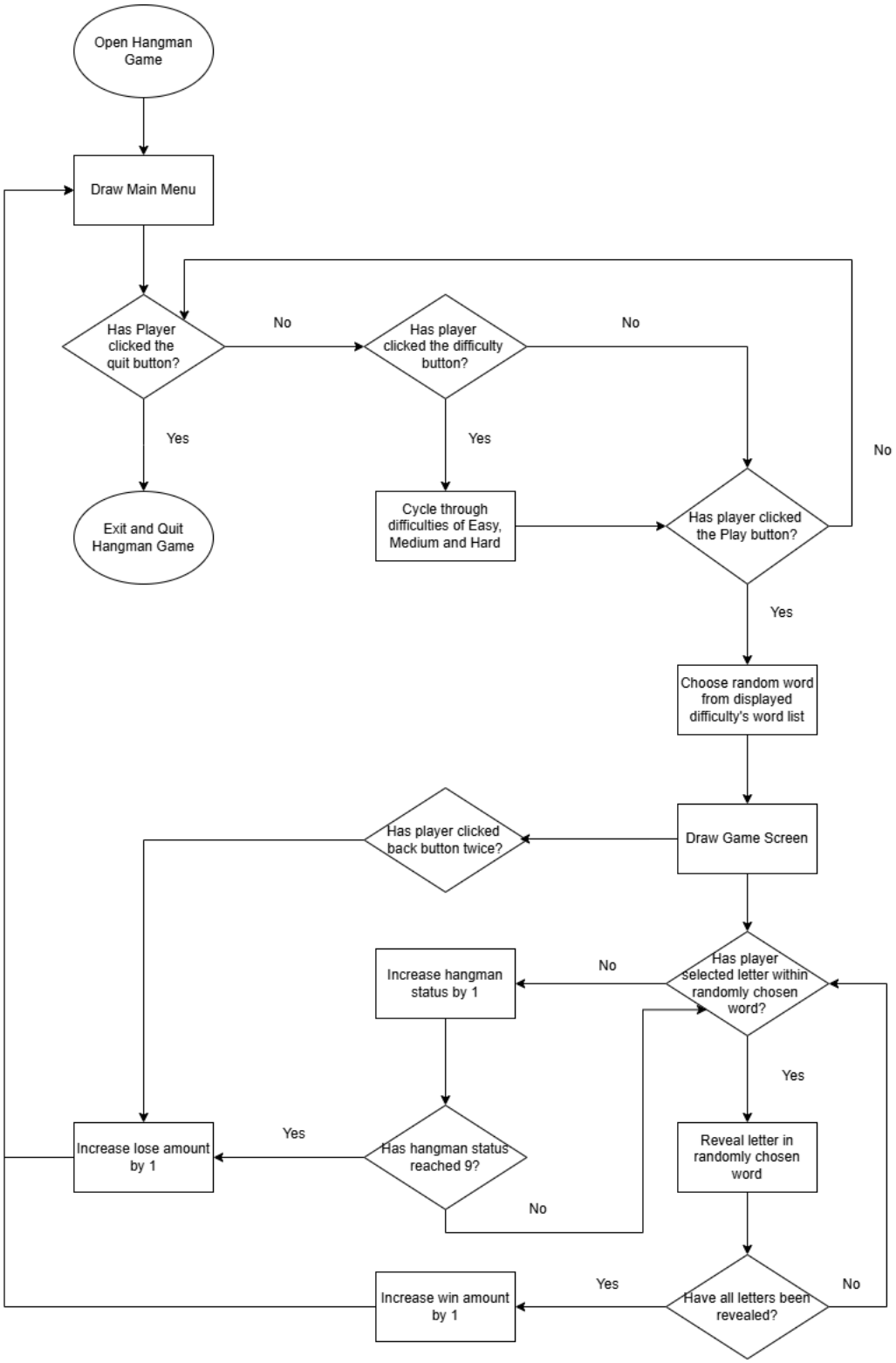
Assessment Criteria #2:

Class/Sequence diagram, Pseudocode, IPO context diagrams, Data dictionary

Class/Sequence Diagram:



Pseudocode:
<Completed on Separate PDF>

IPO Context Diagrams:

Play Button Functionality

Input	Process	Output
• Position of Mouse Click • Bounding Box of Play button • "Current_state" variable	1. Detect if current_state matches "menu" 2. Compare if the position of the mouse press is inside the bounding box of the rectangle	1. Change "current_state" variable to "game" 2. Run word selection process 3. Fill screen with white 4. Redraw title 5. Draw W/L statistics 6. Display image of first level of hangman status. 7. Draw chosen word on screen with "_" representing unguessed letters

Difficulty Button Functionality

Input	Process	Output
• Position of Mouse Click • Bounding Box of Difficulty button • "Difficulty" variable value • "Current_state" variable	1. Detect if current_state matches "menu" 2. Compare if the position of mouse press is inside the bounding box of the rectangle 3. Detect the value of the Difficulty variable	1. Increase Difficulty Variable by 1 (If 2>, reset back to 0) 2. Draw name of difficulty depending on value of Difficulty variable (0 = Easy, 1 = Medium, 2 = Hard)

Quit Button Functionality

Input	Process	Output
• Position of Mouse Click • Bounding Box of Quit button • "Current_state" variable	1. Detect if current_state matches "menu" 2. Compare if the position of the mouse press is inside the bounding box of the rectangle	1. Set run variable to false, thus stopping the game loop 2. Close application

Word Selection Process

Input	Process	Output
• Value of Difficulty variable • Word string variable	1. Detect if Difficulty = 0, fill words list variable with words from easy words list file 2. Else if Difficulty = 1, fill words list variable with words from medium words list file 3. Else if Difficulty = 3, fill words list variable with words from hard words list file 4. Choose a random word from all words within words list variable	1. Word string will be populated with a single randomly chosen word

Letter Button Functionality

Input	Process	Output
• Position of Mouse Click • Radius of circle drawn over letter button	1. Detect if mouse click position is within radius of circle drawn 2. Detect if letter chosen is in randomly chosen displayed word	1. If letter is in displayed word, unhide all of that specific letter within that word 2. Else if letter is not in displayed word, increase hangman status by 1

Back Button Functionality

Input	Process	Output
• Position of Mouse Click • Bounding Box of Back button • Value of “shown_warning” variable • “Current_state” variable	1. Detect if current_state matches “game” 2. Compare if the position of the mouse press is inside the bounding box of the rectangle 3. Detect value of “shown_warning” variable	1. If shown_warning variable = 0, draw warning message on screen 2. If shown_warning variable = 1, increase player lose_amounts by 1 3. If shown_warning variable = 1, set “current_state” to “menu” 4. If shown_warning variable = 1, redraw menu

Data dictionary:

Field Name	Data Type	Description
Hangman Status	Integer	Used to determine what image of hangman to load and how close a player is to losing a game. If hangman status = 9, the player loses the game.
Words	List	Used to store all words from specific word list chosen by difficulty variable
Word	String	Used to randomly select a word from all words in the words list variable. This word is then used as the word that needs to be guessed.
Guessed	List	Used to store all letters in the game that have been guessed. If all letters in the guessed list are also in the word trying to be guessed, the player wins the game.
Difficulty	Integer	Used to determine which word list the words list variable will populate from. Also used to determine what difficulty will be displayed next to the difficulty button and in-game title.
Current State	String	Used to track whether the player is playing a game or looking at the main menu. Buttons activate and deactivate depending on this string.
Win Amount	Integer	Used to store the total amount of wins the player has accumulated after playing. Displayed as the green number on the bottom of the screen.
Lose Amount	Integer	Used to store the total amount of losses the player has accumulated after playing. Displayed as the red number at the bottom of the screen.