BSC – HGP- Assignment 02
Pictionary Game
UI Design Document

Student Name: Xiang Yu Oon
Student Number: 3159517

Gameplay

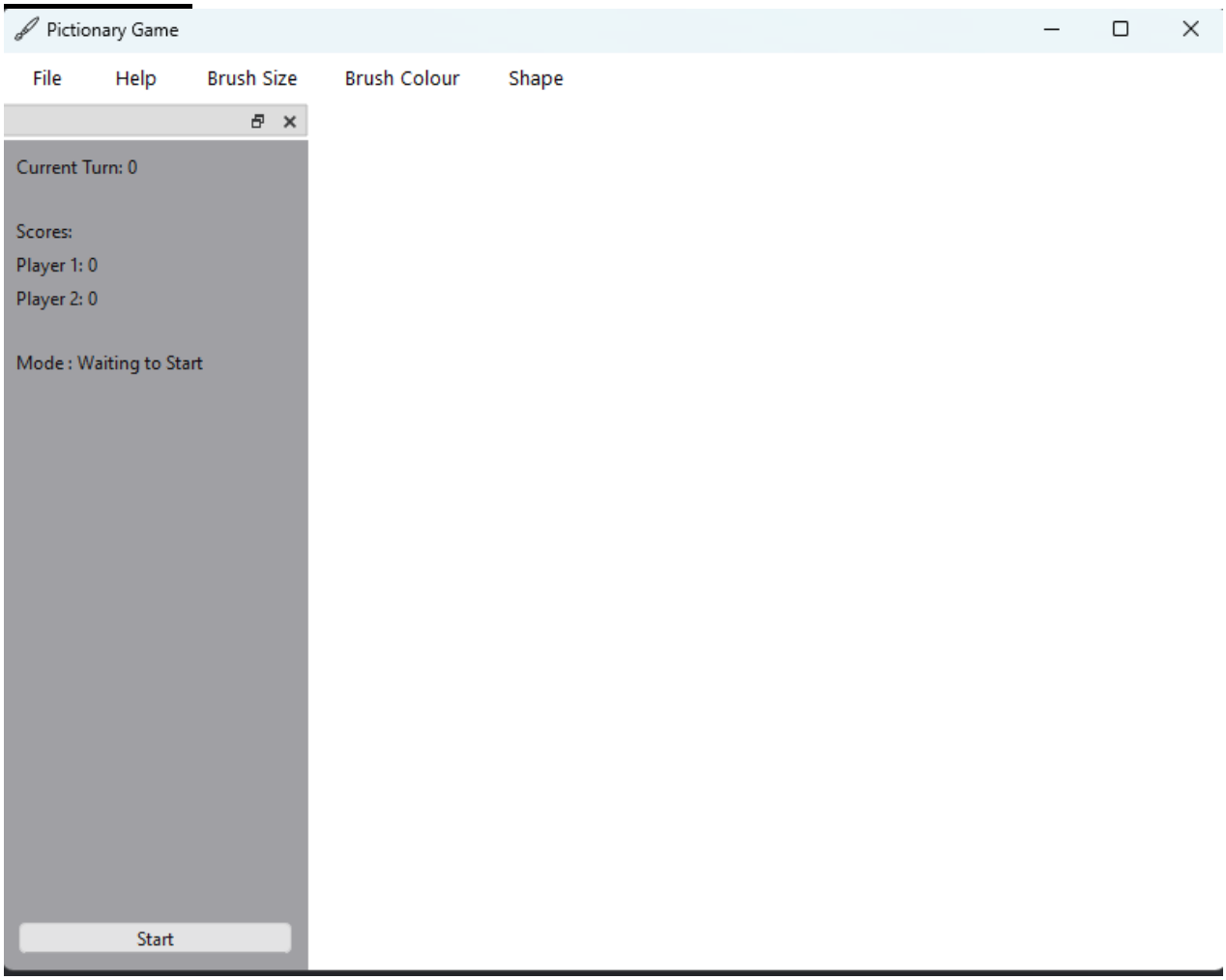
- The first player to reach 5 points wins the game.
- If the guesser answers correctly, the guesser receives 2 points, and the drawer receives 1 point.
- The drawer can skip their turn if the word is too difficult to draw.
- The guesser has up to 4 attempts to answer each question. If they answer incorrectly on the fourth attempt, the turn will end.
- The game result will be either a win or a loss.

WHY:
Five points for each game will not take too long. Additionally, I set the points so that the drawer is encouraged to add more details, as they will earn a point if the guesser answers correctly.

Instead of using time for each round, I'm using chances to move the game forward because time would stress the drawer more than the guesser. The guesser has the potential to earn more points, so I want to place more pressure on the guesser rather than the drawer.

The result is either a win or a loss, with no option for a draw. This ensures that players are motivated to give their best effort throughout the game, as a draw would eliminate the sense of urgency to win.

Main UI

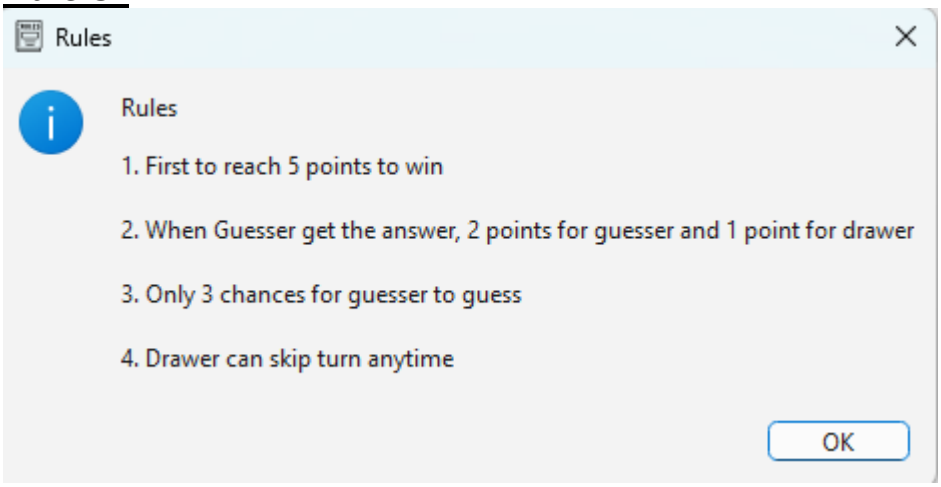


WHY:
This is the main UI of my Pictionary game. As you can see, it is similar to the template. However, there are some differences: the button on the bottom left and the menu bar have one more option. The left side of the UI is a dock window used to show the game stats. The upper section contains functions for file options and brush adjustments.

In this assignment, I'm using multiple windows for different operations and a global instance to store the game stats. This is to simulate players using different PCs to play.

HOW:
The upper section uses QMenuBar, and the left side uses QDockWidget. The layout and design are very clean and simple.

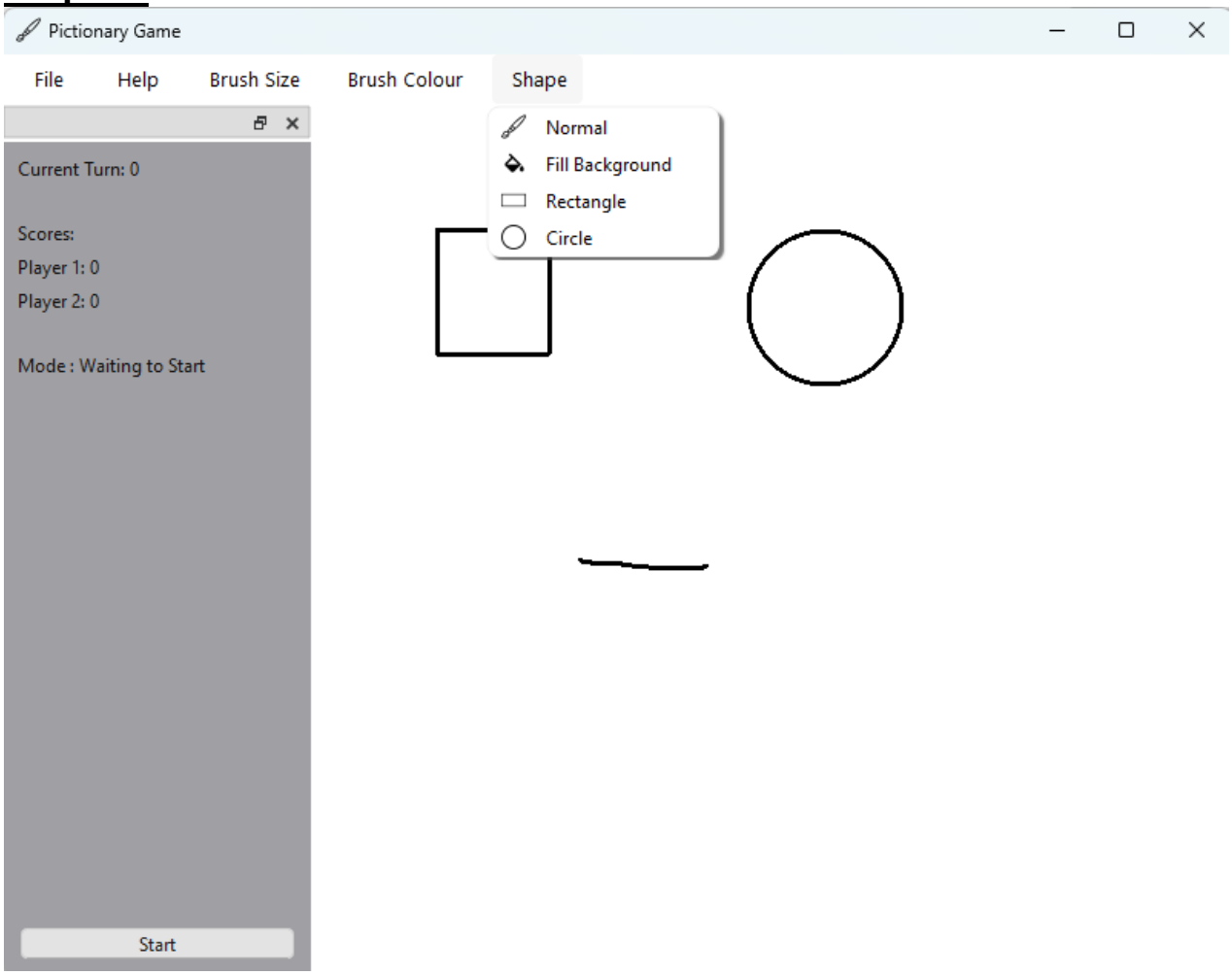
Rule UI



WHY:
A new window clearly displays the rules, ensuring that all players can easily understand them. This window can be opened on top of the main UI via the help bar or by using the shortcut key Ctrl + H.

HOW:
This is a QMessageBox, which is commonly used to display information or warnings.

Shape UI



WHY:

There are four different drawing methods: free drawing, fill background, rectangle, and circle. These methods make it easy for the player to draw what they want.

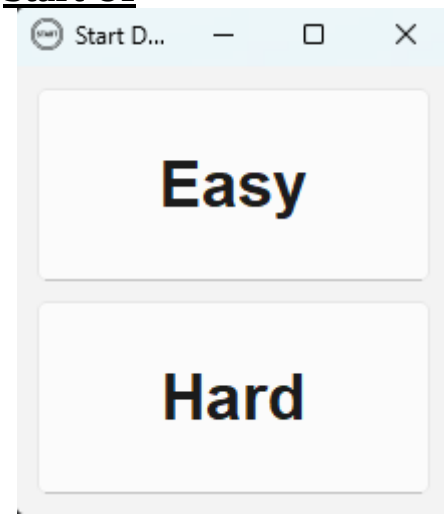
- Free drawing: This is a regular pen that allows you to draw freely without any restrictions.
- Fill background: This can fill the whole background with a colour.
- Rectangle: This can draw a rectangle with the size and colour of user selected.
- Circle: This can draw a rectangle with the size and colour of user selected.

HOW:

For the rectangle, it is drawn using QRect, which essentially saves the starting point and the endpoint to form a rectangle. The circle is more complex; it uses the QPainter function drawEllipse and some calculations to determine the center and radius.

GamePlayUI

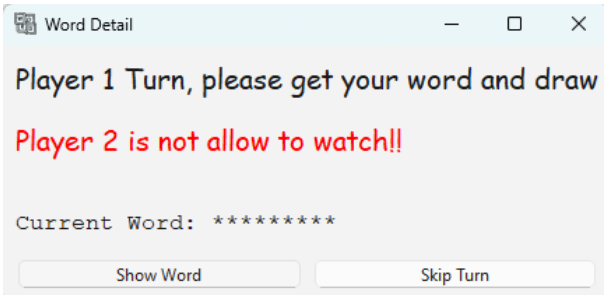
Start UI



WHY:
When you click Start, a pop-up window appears, allowing you to choose the difficulty. I decided to use a separate window to make it clear to the player that the game had started.

HOW:
It is very easy and basic; it is just using another QMainWindow to do this.

Word Display UI



WHY:
After you select the difficulty, this window will pop up. It follows the same logic as the Start window. I used different fonts to distinguish between the word and the rules for the player. Red is used for warnings, as people naturally associate red with avoidance.

The Skip button allows the drawer to skip their turn if the word is too difficult to draw.

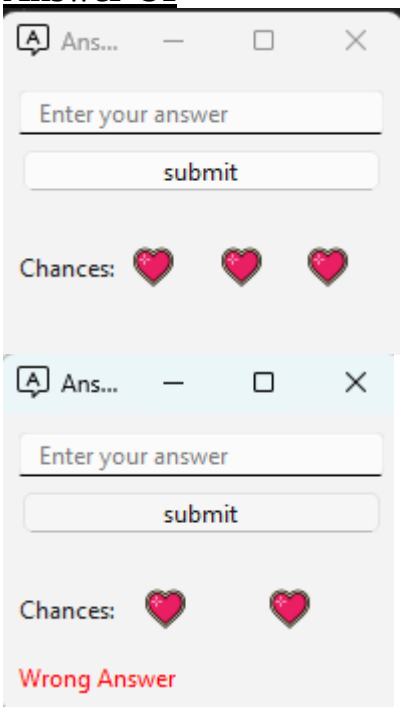
When the user presses the 'Show Word' button, the word will be displayed. When the user releases the button, the word will revert to a symbol. This can ensure the guesser can't see the word.

HOW:
It is very easy and basic; it is just using another QMainWindow to do this.

The skip function essentially increments the turn global variable by 1 and updates the UI.

The 'Show Word' button uses the pressed and released functions of QPushButton.

Answer UI



WHY:
After you select the difficulty, this window will pop up. This is a different window from the word display UI. I'm using two separate windows to simulate two players playing on two desktops.

I've assigned three chances for the guesser to answer, aiming to encourage proper guesses. When the chances are used up, the game moves to the next turn.

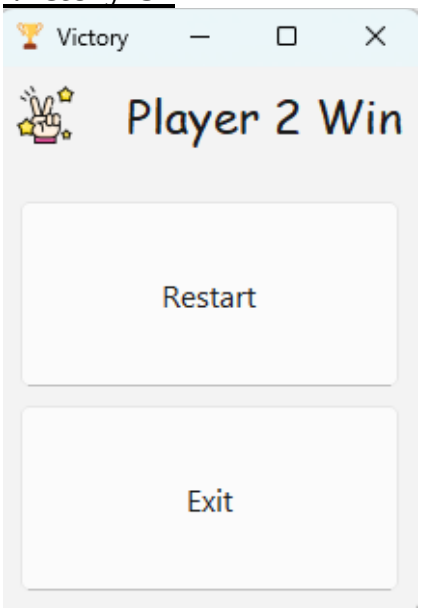
When the user answers incorrectly, the heart representing the guesser's chances will decrease by 1, indicating that one chance has been deducted.

Then, a label will appear below the chances to indicate whether the last answer was correct or incorrect. A red label will mean the answer was wrong, and a green label will mean it was correct, similar to a traffic light.

HOW:
It is very easy and basic; it is just using another QMainWindow to do this.

For the heart icon, a for loop is used to add the icons into a horizontal layout based on the chances left.

Victory UI



WHY:
When one of the players reaches 5 points, they are declared the winner, and this window will pop up. There are two buttons: click Restart to continue playing, or Exit to close the Pictionary game.

HOW:
A vertical layout is used to keep all widgets organized. The Restart button is linked to the start method, and Exit uses the QApplication function to close all windows.