M02 Introduction to WPF Apps

Author: Yi Chen
Date: 2021.02.04

Table of Contents

1. M02 Introduction to WPF Apps 1
2. Project Requirements 1
2.1. Derived Requirements 1
3. Design Plans 2
3.1. Player vs computer | Computer vs Computer 2
3.1.1. Player vs computer 2
3.1.2. Computer vs Computer 3
3.1.3. Labels && End Game 3
4. Implementation 4
4.1. Player & Computer Implementation 4
4.2. Computer & Computer Implementation 6
4.3. Label & End Game Implementation 8

List of Tables

Table 1: Labels and End Game 4
Table 2: Computer & Computer 6
Table 3: Lables and End Game 8

List of Figures

Figure 1: Player & computer 4

1. M02 Introduction to WPF Apps

Using the Windows Presentation Foundation .Net Framework create an app which mimics a rock-paper-scissor game.

2. Project Requirements

- *Use WPF with .Net Framework 4.7.2 to create a two-player game in the same style as rock-paper-scissor.*
- *Each player has at a minimum 3 cards.*
- *Use, at a minimum, the following WPF controls:*
- *Button, Image, Label, Textbox.*
- *User interface requirements:*
- *Divide the game screen in two so both players are represented.*
- *Keep track of each players score.*
- *The game cards should be represented as images, not just a number.*
- *A 'Play Game' button.*
- *Will randomly select a card from each player's deck.*
- *Will compare the cards and declare, if possible, a winner.*
- *Add a score to the winner's pot.*
- *A 'End Game' button.*
- *Will declare a winner of the game.*
- *Allows user to exit application.*

2.1. Derived Requirements

1. *prepared PDF document with your commented code.*
   - *Document should have relevant meta information and be well formatted.*
2. *Application executable in zipped format (do not submit a .exe file to Brightspace).*

3. Design Plans

Using WPF .NET framework - WPF Controls which including Button, Image, Label, Textbox.

3.1. Player vs computer | Computer vs Computer

There are two play model : Player vs computer and computer vs computer.

3.1.1. Player vs computer

Before game started, player show a cat image and computer show a robot image.

Rock Button Click event:

- *when a button is clicked the Image show rock.jpg, paper.jpg, scissor.jpg are same.*

Computer play Button Click event:

- *computer choose a card randomly.*

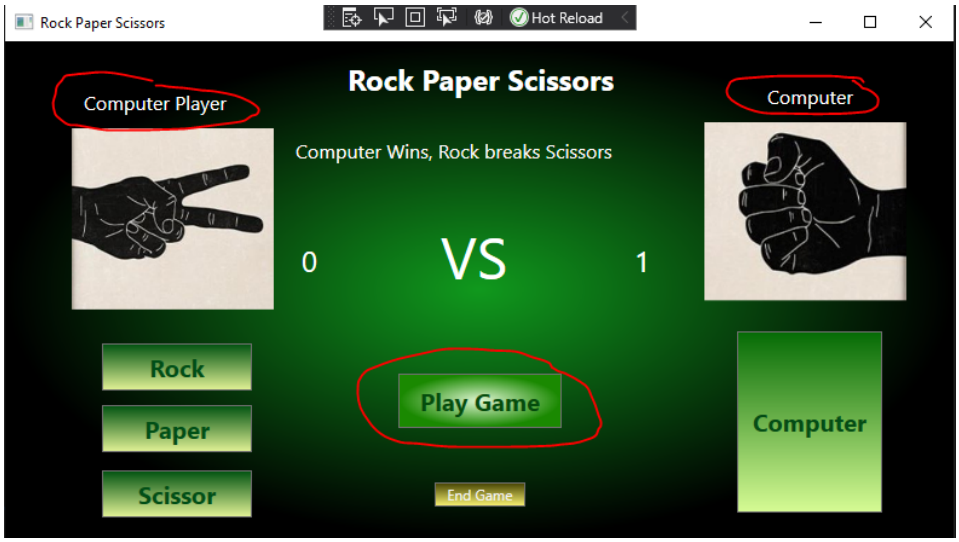
3.1.2. Computer vs Computer

Play Game Button:

- *Computer will choose one card for each computer player and the result will be sent to CheckGame().*
- *Change the player label to computer player.*

CheckGame()

- *Calculate the result then put into the win textbox .*



3.1.3. Labels && End Game

Winner labels:

- *Show the result after CheckGame() calculated.*
- *Show final result after End Game Button be pressed.*

Score Labels:

- *Show the plyer and computer scores*

End Game:

- *Initialize the game score.*
- *Initialize the image.*
- *Put the final result of the game score and show the winner.*

Table 1: Labels and End Game

4. Implementation

MainWindow.xaml.cs:

- *Implement all design function*

MainWindow.xaml:

- *Implement all design layout*

4.1. Player & Computer Implementation

Play:

- *R_Btn_Click*
- *P_Btn_Click*
- *S_Btn_Click*

Computer:

- *Computer_play_Btn_Click*

Figure 1: Player & computer

Player:

```
/**
         * @R_Btn_Click
         * set rock button click event: when click button the image change to
rock.
         */
        private void R_Btn_Click(object sender, RoutedEventArgs e)
        {
```

```
            User.Source = new BitmapImage(new Uri(@"/Images/rock.jpg",
UriKind.Relative));
            Player_LB.Content = "Player";
            playerChoice = "rock";
        }

        /**
         * @R_Btn_Click
         * set paper button click event: when click button the image change to
paper.
         */
        private void P_Btn_Click(object sender, RoutedEventArgs e)
        {
            User.Source = new BitmapImage(new Uri(@"/Images/paper.jpg",
UriKind.Relative));
            Player_LB.Content = "Player";
            playerChoice = "paper";
        }

        /**
         * @R_Btn_Click
         * set scissor button click event: when click button the image change to
scissor.
         */
        private void S_Btn_Click(object sender, RoutedEventArgs e)
        {
            User.Source = new BitmapImage(new Uri(@"/Images/scissors.jpg",
UriKind.Relative));
            Player_LB.Content = "Player";
            playerChoice = "scissors";
        }
```

Computer:

```
  /**
         * Computer will choose a card randomly
         *
         */

        private void Computer_play_Btn_Click(object sender, RoutedEventArgs e)
        {
            randomNumber = rnd.Next(0, ComputerchoiseList.Length);

            ComputerChoice = ComputerchoiseList[randomNumber];


            switch (ComputerChoice)
            {
                case "rock_r":
                    Computer.Source = new BitmapImage(new
Uri(@"/Images/rock_r.jpg", UriKind.Relative));
                    break;

                case "paper_r":
                    Computer.Source = new BitmapImage(new
Uri(@"/Images/paper_r.jpg", UriKind.Relative));
                    break;
```

```
                case "scissor_r":
                    Computer.Source = new BitmapImage(new
Uri(@"/Images/scissors_r.jpg", UriKind.Relative));
                    break;

            }

            CheckGame();


        }
```

4.2. Computer & Computer Implementation

PlayGame_LB_Click:

- *Switch*
- *ChackGame()*

Table 2: Computer & Computer

Switch:

```
private void PlayGame_LB_Click(object sender, RoutedEventArgs e)
        {
            Player_LB.Content = "Computer Player";
            randomNumber = rnd.Next(0, ComputerchoiseList.Length);
            randomNumberp = rnd.Next(0, PlayerchoiseList.Length);
            ComputerChoice = ComputerchoiseList[randomNumber];
            playerChoice = PlayerchoiseList[randomNumberp];


            switch (playerChoice)
            {
                case "rock":
                    User.Source = new BitmapImage(new
Uri(@"/Images/rock.jpg", UriKind.Relative));
                    break;

                case "paper":
                    User.Source = new BitmapImage(new
Uri(@"/Images/paper.jpg", UriKind.Relative));
                    break;

                case "scissor":
                    User.Source = new BitmapImage(new
Uri(@"/Images/scissors.jpg", UriKind.Relative));
                    break;

            }


            switch (ComputerChoice)
            {
                case "rock_r":
```

```
                    Computer.Source = new BitmapImage(new
Uri(@"/Images/rock_r.jpg", UriKind.Relative));
                    break;

                case "paper_r":
                    Computer.Source = new BitmapImage(new
Uri(@"/Images/paper_r.jpg", UriKind.Relative));
                    break;

                case "scissor_r":
                    Computer.Source = new BitmapImage(new
Uri(@"/Images/scissors_r.jpg", UriKind.Relative));
                    break;

            }

            CheckGame();

        }
```

ChackGame():

```
        /**
         * @CheckGame()
         * Do the calculation and out put score to the textbox
         *
         */
        private void CheckGame()
        {
            //computer
            if (playerChoice == "rock" && ComputerChoice == "paper_r")
            {
                computerScore += 1;
                ComputerScore_LB.Content = computerScore;
                Winner_LB.Content = "Computer Wins, Paper Covers Rock";
            }
            else if (playerChoice == "scissor" && ComputerChoice == "rock_r")
            {
                computerScore += 1;
                ComputerScore_LB.Content = computerScore;
                Winner_LB.Content = "Computer Wins, Rock breaks Scissors";

            }
            else if (playerChoice == "paper" && ComputerChoice ==
"scissor_r")
            {
                computerScore += 1;
                ComputerScore_LB.Content = computerScore;
                Winner_LB.Content = "Computer Wins, Scissor cuts Paper";

            }

            //player
            else if (playerChoice == "paper" && ComputerChoice == "rock_r")
            {
                playerScore += 1;
                PlayerScore_LB.Content = playerScore;
                Winner_LB.Content = "Player Wins, Paper Covers Rock";
```

```
            }

            else if (playerChoice == "scissor" && ComputerChoice ==
"paper_r")
            {
                playerScore += 1;
                PlayerScore_LB.Content = playerScore;
                Winner_LB.Content = "Player Wins, Scissor cuts Paper";
            }

            else if (playerChoice == "rock" && ComputerChoice == "scissor_r")
            {
                playerScore += 1;
                PlayerScore_LB.Content = playerScore;
                Winner_LB.Content = "Player Wins, Rock breaks Scissors";
            }
            else if (playerChoice == "none")
            {
                Winner_LB.Content = "Make a choice";
            }
            else
            {
                Winner_LB.Content = "Draw!!!";
            }


        }
```

4.3. Label & End Game Implementation

- *Winner_LB*
- *ComputerScore_LB*
- *PlayerScore_LB*

- *End Game*

Table 3: Lables and End Game

```
/**
         * @St_Btn_Click
         * when stop button be pressed the final result will be out put to win
textbox
         * initalize the scores.
         */

        private void St_Btn_Click(object sender, RoutedEventArgs e)
        {
            User.Source = new BitmapImage(new Uri(@"/Images/User.jpg",
UriKind.Relative));
            Computer.Source = new BitmapImage(new Uri(@"/Images/Robot.jpg",
UriKind.Relative));
            Winner_LB.Content = "Player: " + playerScore + " - " + "CPU: " +
computerScore;
```

```
            if (playerScore > computerScore)
            {
                Winner_LB.Content ="Player Won!!!  " + "Player: " + playerScore +
" - " + "Computer: " + computerScore;
            }
            else if(computerScore > playerScore)
            {
                Winner_LB.Content = "Computer Won!!!  " + "Player: " + playerScore
+ " - " + "Computer: " + computerScore;
            }
            else
            {
                Winner_LB.Content = "Tied!!!  " + "Player: " + playerScore + " - "
+ "Computer: " + computerScore;
            }

            computerScore = 0;
            playerScore = 0;
            PlayerScore_LB.Content = 0;
            ComputerScore_LB.Content = 0;

        }
```