Mi Gao

**CPS 280 - Lab Assignment #8**

**Creating a Tabbed Application**

Generally, I designed and created a C# Console or GUI application to play 5 different simple games in an OO manner (each game is in a separate class file). It includes:

1. **Coin Flip –** have the user call the toss by selecting from radio buttons, then a button to flip the coin.  Show the face of the coin as two images you “flip.”  A correct toss gives 1 winner point.
2. **Roll Dice (not entirely done) –** utilize a method which displays two die as colored boxes with large text or images for the six sides.  Have a button “roll” the die, and determine a winner if they get 7 or 11, in which they get 1 winner point.
3. **Pick a Card –** keep an array of 52 cards and generate a different one for a button press.  A face card is a winner, giving 1 winner point.
4. **Play Roulette (not entirely done) –** have the player choose a color with a radio button (black or red), and a number (between 1 and 16) in a textbox.  Then have a button spin the wheel (the wheel needn’t be shown), and return a color (red or black) and a number.  The correct color gives a winner point; the correct number gives 5 winner points.
5. **Rock, Paper, Scissors –** is a [zero-sum](https://en.wikipedia.org/wiki/Zero-sum_game) [hand game](https://en.wikipedia.org/wiki/Hand_game) usually played between two people, in which each player simultaneously forms one of three shapes with an outstretched hand. These shapes are "rock", "paper", and "scissors". The game has only three possible outcomes other than a tie: a player who decides to play rock will beat another player who has chosen scissors ("rock crushes scissors") but will lose to one who has played paper ("paper covers rock"); a play of paper will lose to a play of scissors ("scissors cut paper"). Winner get 1 point.

UML diagram:

The code:

Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace TabbedProgram

{

public partial class Form1 : Form

{

public int image = 1;

public int score = 0;

Slots slotGame = new Slots();

CardGame poker = new CardGame();

Guess guess = new Guess();

Toss toss = new Toss();

public Form1()

{

InitializeComponent();

}

private void btnChangeFace\_Click(object sender, EventArgs e)

{

switch (image)

{

case 1: picFace.Load("../../images/evilface.png");

break;

case 2: picFace.Load("../../images/sadface.png");

break;

case 3: picFace.Load("../../images/happyface.png");

break;

}

++image;

if (image == 4)

image = 1;

}

private void tabPage1\_Click(object sender, EventArgs e)

{

}

private void Form1\_Load(object sender, EventArgs e)

{

showDeck();

}

private void btnSpin\_Click(object sender, EventArgs e)

{

slotGame.play();

int n;

n = slotGame.getRandom();

switch (n)

{

case 0: picFace.Load("../../images/evilface.png");

break;

case 1: picFace.Load("../../images/sadface.png");

break;

case 2: picFace.Load("../../images/happyface.png");

break;

case 3: picFace.Load("../../images/texting.jpg");

break;

case 4: picFace.Load("../../images/tongue.jpg");

break;

}

txtSpinNum.Text = n.ToString();

}

private void showDeck()

{

txtDeckBefore.Text = poker.getDeck();

}

private void btnShuffle\_Click(object sender, EventArgs e)

{

poker.shuffleDeck();

txtDeckAfter.Text = poker.getDeck();

score += 1;

txtScore.Text = score.ToString();

}

private void tabPage2\_Click(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void btnScissors\_Click(object sender, EventArgs e)

{

// Show what player's choice on the label

lblPlayer.Text = btnScissors.Text;

// Show what computer's choice on the label

string strPC = string.Empty;

switch (guess.PC())

{

case 1:

strPC = "Scissors";

break;

case 2:

strPC = "Rock";

break;

case 3:

strPC = "Paper";

break;

default:

throw new Exception("Unknown: Cannot decide what the Player show!");

}

lblPC.Text = strPC; // show what the computer's hand out

// Obtain the value in label and converse to int

int vPlayer = guess.ChangeStrToInt(lblPlayer.Text);

// Call the way of Judge. Compare the values and show the result in the label

int pc = guess.PC();

lblResult.Text = guess.Judge(vPlayer, pc);

if (lblResult.Text == "Win"){

score += 1;

txtScore.Text = score.ToString();

}

}

private void btnRock\_Click(object sender, EventArgs e)

{

lblPlayer.Text = btnRock.Text;

string strPC = string.Empty;

switch (guess.PC())

{

case 1:

strPC = "Scissors";

break;

case 2:

strPC = "Rock";

break;

case 3:

strPC = "Paper";

break;

default:

throw new Exception("Unknown: Cannot decide what the Player show!");

}

lblPC.Text = strPC;

int vPlayer = guess.ChangeStrToInt(btnRock.Text);

int pc = guess.PC();

lblResult.Text = guess.Judge(vPlayer, pc);

if (lblResult.Text == "Win")

{

score += 1;

txtScore.Text = score.ToString();

}

}

private void btnPaper\_Click(object sender, EventArgs e)

{

lblPlayer.Text = btnPaper.Text;

string strPC = string.Empty;

switch (guess.PC())

{

case 1:

strPC = "Scissors";

break;

case 2:

strPC = "Rock";

break;

case 3:

strPC = "Paper";

break;

default:

throw new Exception("Unknown: Cannot decide what the Player show!");

}

lblPC.Text = strPC;

int vPlayer = guess.ChangeStrToInt(btnPaper.Text);

int pc = guess.PC();

lblResult.Text = guess.Judge(vPlayer, pc);

if (lblResult.Text == "Win")

{

score += 1;

txtScore.Text = score.ToString();

}

}

// Toss part

private void btnFlip\_Click(object sender, EventArgs e)

{

if (!radioButton1.Checked && !radioButton2.Checked)

{

MessageBox.Show("Please select a choice!", "Error!");

}

else

{

// user Inputs

if (radioButton1.Checked)

toss.userGuess = "Heads";

if (radioButton2.Checked)

toss.userGuess = "Tails";

toss.HeadsorTails();

switch (toss.HeadsorTails())

{

case "Heads":

picCoin.Load("../../images/heads.jpg");

break;

case "Tails":

picCoin.Load("../../images/tails.jpg");

break;

}

if (!toss.userGuess.Equals(toss.headOrTail))

{

MessageBox.Show("You lost!");

//picCoin.Image = null;

}

else

{

MessageBox.Show("You won!");

score += 1;

txtScore.Text = score.ToString();

//picCoin.Image = null;

}

}

}

private void label3\_Click\_1(object sender, EventArgs e)

{

}

}

}

IGame.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TabbedProgram

{

interface IGame

{

void play();

int getRandom();

}

}

Toss.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TabbedProgram

{

class Toss

{

public string headOrTail = "", userGuess = "";

public string HeadsorTails()

{

Random headsOrTails = new Random();

int headsTails = headsOrTails.Next(1, 3);

if (headsTails.Equals(1))

{

headOrTail = "Heads";

}

else

{

// pic = Properties.Resources.Tails;

headOrTail = "Tails";

}

return headOrTail;

}

}

}

Guess.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TabbedProgram

{

class Guess

{

/// Setting mouse click from string to int

public int ChangeStrToInt(string str)

{

int n = 0;

//determine str or not

switch (str)

{

case "Scissors":

n = 1;

break;

case "Rock":

n = 2;

break;

case "Paper":

n = 3;

break;

default:

throw new Exception("Unknown");

}

return n;

}

// the way of how computer

public int PC()

{

Random ran = new Random();

//random.Next: take minimum, but not maximum, so take random value from 1 to 3

int vPC = ran.Next(1, 4);

return vPC;

}

/// Comparing the value between player's and computer's

public string Judge(int Player, int PC)

{

// Scissors: 1 Rock: 2 Ppaer: 3

// 2 way for win: one way is Player's value larger 1 than computer's; other one is small 2

int tmp = Player - PC;

string judgedResult = string.Empty;

if (tmp == 1 || tmp == -2)

judgedResult = "Win";

else if (tmp == 0)

judgedResult = "Draw";

else

judgedResult = "Lost";

return judgedResult;

}

}

}

Slots.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace TabbedProgram

{

class Slots : IGame

{

public void play()

{

}

public int getRandom()

{

Random r = new Random();

return r.Next(5);

}

}

}

Output samples:

