Kuwait University
College of Computing Sciences and Engineering
Department of Information Science

COMP 2139
Web Application Development with C# .NET

Lab 8
The PIG (Dice) Game Application

Table of Contents

Laboratory #8 – The PIG (Dice) Game Application

1. Laboratory Objective

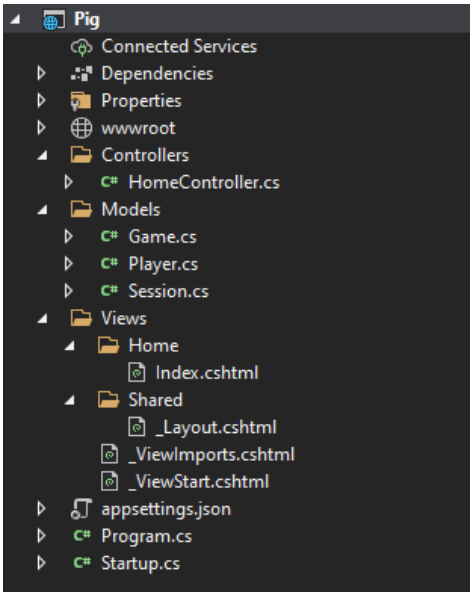
The goal of this lab is to build a single page application (SPA) that uses session state to persist data between requests.

2. **Laboratory Learning Outcomes:** After conducting this laboratory students will be able to:
    a. Create and maintain a Git repository project
    b. Create the PIG (Dice) Game Application using C#.NET
    c. Build a single-page, using session-state to persist data across requests.

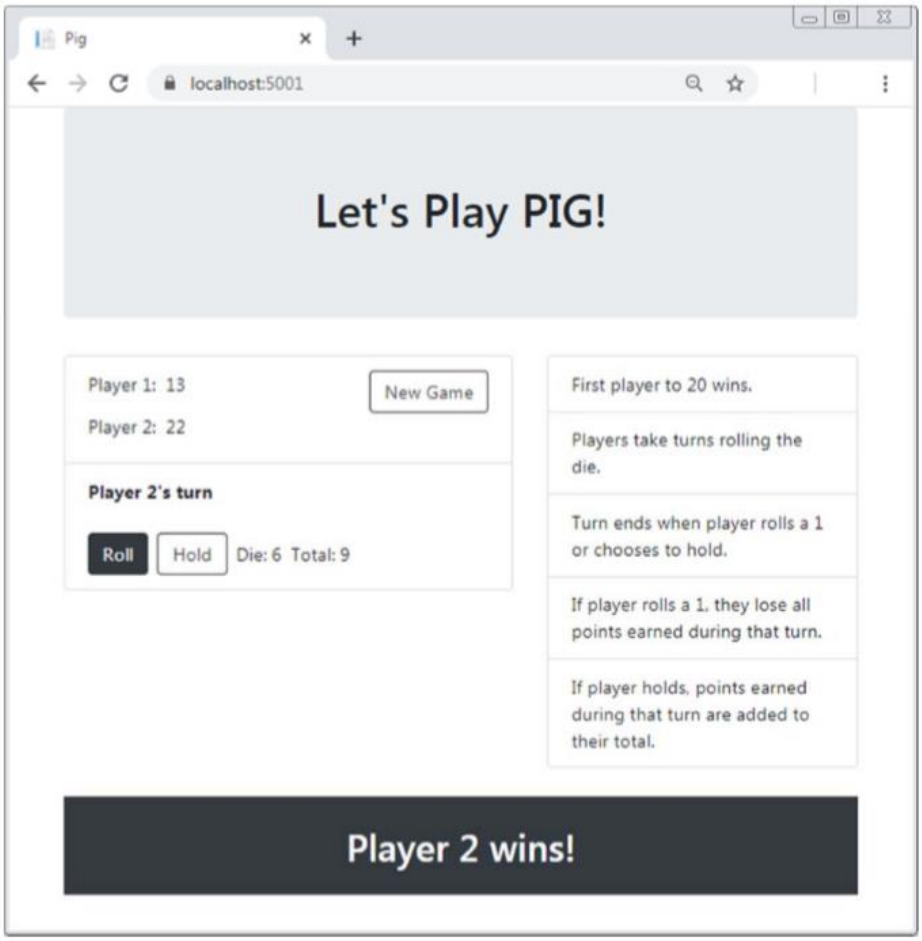
3. Laboratory Instructions

For this lab, we will build a multi-page, web application that will have a structure similar to the one presented below:

The PIG (Dice) Game Application



The PIG Game Application



Specifications

- When the application starts, or when the user clicks New Game, the scores for both players, the value for the die, and the total for the turn should all be **zero**.

- When it's the users turn, a player can click Roll to roll the die or **Hold** to end their turn and add the point total for that turn to their score.

- When a player clicks **Roll**, the application should generate and display a random number **between 1 and 6**. If the number is **1**, the point total for the turn should be set to **zero**, and that player's turn should end. Otherwise, the number rolled should be added to the point total for the turn, and the player's turn should **continue**.

- When a player clicks **Hold**, the app should add their points for that turn to their score and then end the turn. If the score is greater than the number of points needed to win, the app should end the game and display the name of the winner.

- Clicking the browser's Refresh button should only redisplay the page, not post the previous button click again.

- Use the Random class in the System namespace to generate a random number like this:

```
Random rand = new Random(); int roll = rand.Next(1, 7);
```

- Use session state to store game data between requests.

Creating Git Repository (Optional / Review)

1. Refer to past lab instructions on creating new repositories and clone them into your local workspace.

Creating the Web Application (Review)

1. Within Visual Studio select **File→New→Project**, select **ASP.NET Core Web Application**.
    a. Fill in your desired project details for Project Name, and Location, then select create
    b. From the resultant page, select **Web-Application (Model-View-Controller),** then **Create**
    c. You should then be presented with your workspace to start creating your project.
    d. Clean-up (delete/remove) all auto-generated **controllers**, **views**, **models** from the project

Create the PIG Game Model

1. Based on the screenshot provided for the PIG game create application model classes, based on the following design:

Player

Attribute	**Description**
Name	String holding players name (player1 or player2)
Score	Integer holding players score

Game

Attribute	**Description**
WinningNumber	Integer set to the winning score total (constant) set to 20.
Player1	Player object
Player2	Player object
LastRoll	Integer holding the value of the last roll (1-6)
CurrentPlayerName	String that holds value of current players name
CurrentTurnScore	Integer that holds value of the turn/round score.
IsGameOver	Boolean set to true (game is over) or false (game is not over)
Method	
Constructor (default)	Creates a new Game
Roll()	Simulates rolling of dice (value 1-6) IF the value rolled == 1 reset current players score to 0 change player ELSE update current score
Hold()	If invoked, the current player switches (player1 ⇔ player2)_ update current score IF the currently updated score == WinningNumber the game is over ELSE current turn = lastroll = 0 change player
ChangePlayer()	Switches currently active player (player1 ⇔ player2)_

SessionExtensions

Methods	Description
SetObject	Sets a session attribute in the current session
GetObject	Get a session attribute in the current session

GameSession

Attribute	Description
GameKey	Store the current Game in session
Session	Current active session
Methods	
Constructor	Sets current session
GetGame	Gets the Game stored in session
SetGame	Sets the Game stored in session

Create the Context

1. No database for this lab, so this is not required.

Update appsettings.json

1. No database for this lab, so no changes required.

Update Startup.cs

1. Configure the Startup.cs for Session (**AddSession()**)
2. Configure the Startup.cs for JSON manipulation for Session objects (**AddNewtonsoftJson()**)
3. Default routing is sufficient

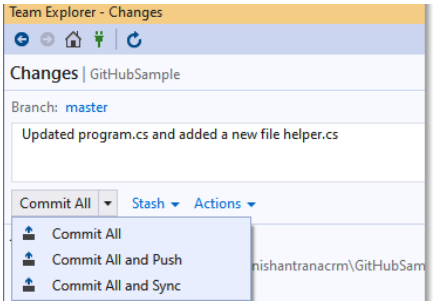
Create the HomeController

1. The HomeController for the The PIG application require the following action methods:
    i. IActionResult **Index()**
    ii. IActionResult **NewGame()**
    iii. RedirectToActionResult **Roll()**
    iv. RedirectToActionResult **Hold()**

Run and test the Application, then Commit

1. Run the application and test to make sure it functions as expected (correct any errors).

2. If all is well, commit the code to your code GitHub repository. To commit the code, navigate to the git changes tab, and select **commit all and push**



Home Work

1. Based on the wireframe/screenshots provided, create the following views for the project namely:
    - Index.cshtml
    - Layout.cshtml