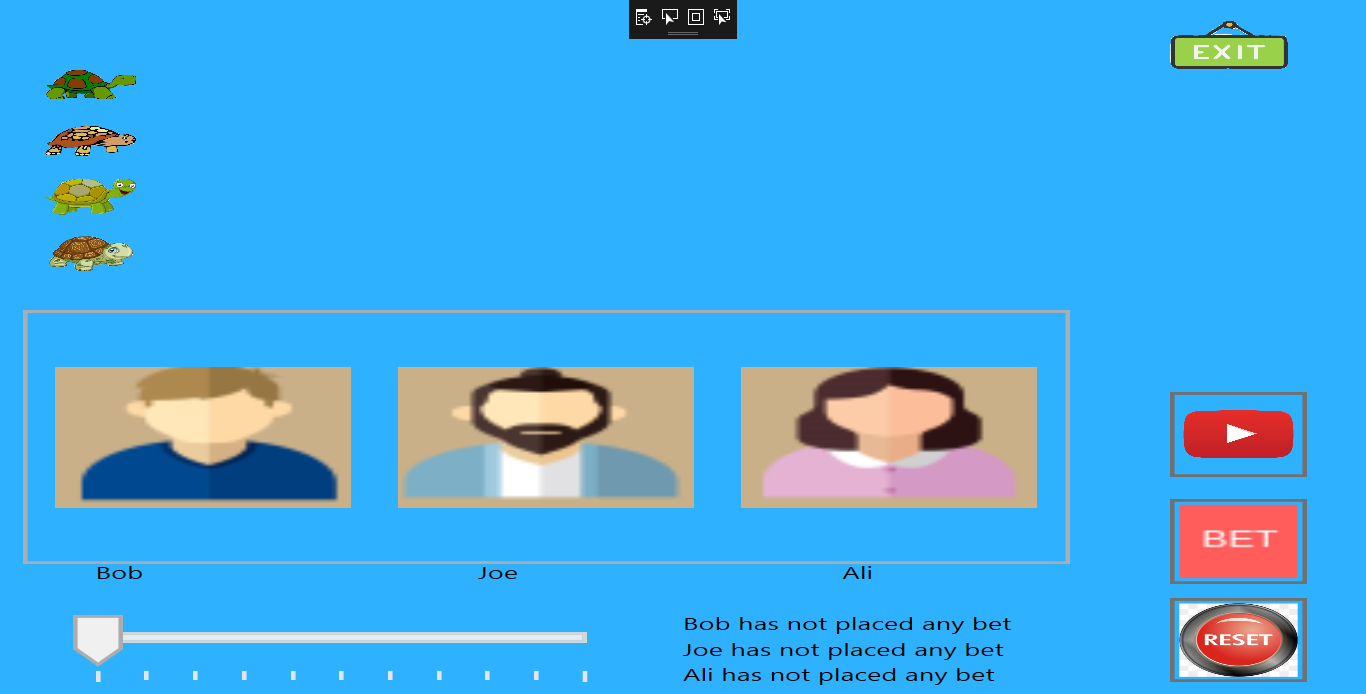
Race Game



Tutorials on making a Greyhound Race Game

Greyhound Races is a betting game where players place bets on one or more out of four tortoise that are racing at the track. This game has three better name Bob, Joe and Ali and four tortoise racer.

This game is based upon following specifications:-

* 1. **Context:** There are two parts to this program, the first is to create a 4 race using random numbers and picture boxes for each racer. The second part of the program is to create a betting system that is based on the outcome of the race.
  2. Each bettor is given some money to bet with, the program has to add and subtract from that original bet until the money has gone. The Max Bet label says how much money the bettor has to spend, and the Up/Down control only goes to that limit, so you can’t spend more than you have.
  3. The game ends when everyone has lost their money.
  4. **The bettors must show**
* The maximum amount that can be bet for each bettor in a label
* The Up/down box can only go to that maximum number for each bettor. (i.e.: Al’s max bet is $45)
  1. When the Bet is laid the Name,Amount and Dog appear on the right.
  2. When the person is out of money, they cannot bet again
  3. When all the betters loose the game is over.

**Pre-requisite**

This project developed in WPF using Visual Studio 2017 and CSharp as a code Behind language. UI is developed in XAML Language.

1. XAML: - *XAML is a declarative markup language that can be used to define user interfaces. The user interface is defined in an XML file using the XAML syntax, while runtime behavior is defined in a separate code-behind file.*  It is available under Microsoft's Open Specification Promise. The acronym originally stood for Extensible Avalon Markup Language, Avalon being the code-name for Windows Presentation Foundation (WPF).

Basic Syntax of XAML :-

<TextBox>This is a Text Box</TextBox>

Advantages of XAML:-

* Easy designing of a UI(User Interface).
* Shorter code then the previous designing techniques.
* The UIs are easier to transfer and present in other environments. E.g: A UI can be presented. On the web or a Windows Client with ease.
* Designing a dynamic UI is absolutely easier with XAML.
* XAML allows creating visible UI elements and separate the UI definition from the programming logic.

2. WPF:- WPF, which stands for Windows Presentation Foundation, is Microsoft's latest approach to a GUI framework, used with the .NET framework. In WPF, UI elements are designed in XAML while behaviors can be implemented in procedural languages such C# and VB.Net. So it very easy to separate behavior from the designer code. There are a lot of GUI frameworks out there, but for .NET developers, the most interesting ones are currently WinForms and WPF. WPF is the newest, but Microsoft is still maintaining and supporting WinForms.

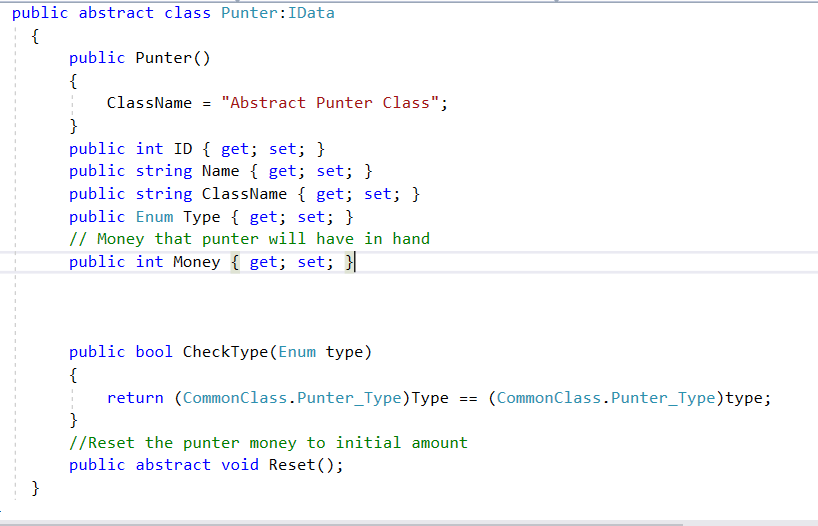
3.C# :- C# (pronounced C sharp) is a general-purpose, multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed around 2000 by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006). C# is one of the programming languages designed for the Common Language Infrastructure.

4. Visual Studio :- Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code. Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

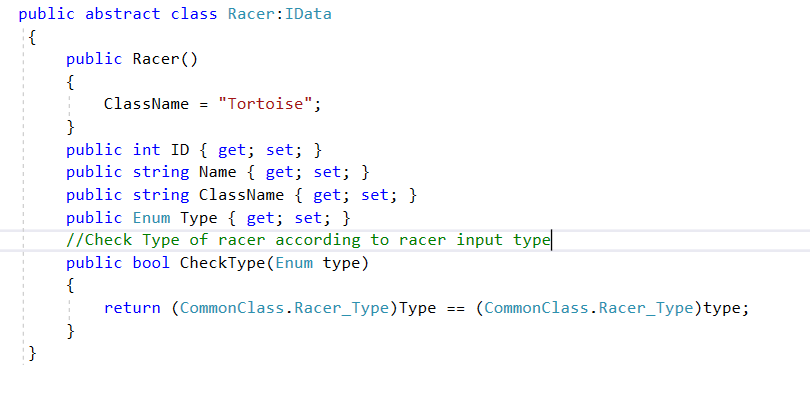
**Coding**

There are two abstract classes Punter and Racer which provides blueprint for Betters and Tortoise Racers which provides basic template for Punter and Racer functionality. Below is the screenshot of Punter and Racer abstract class.

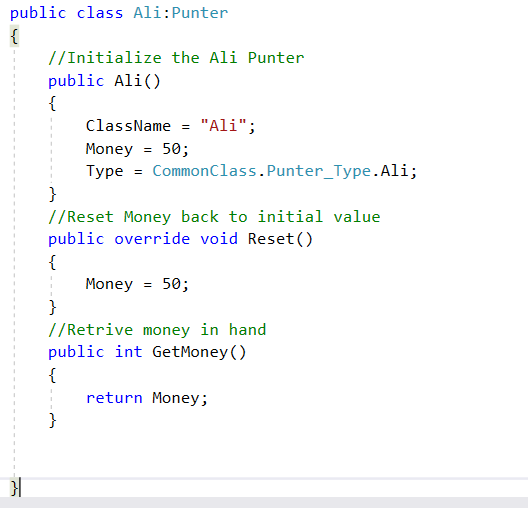
**Punter Class**



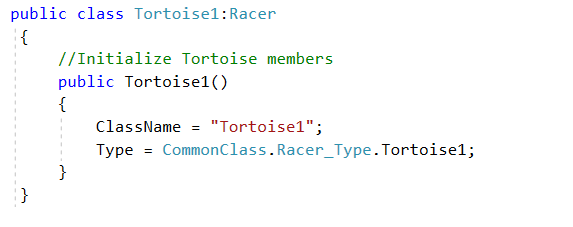
**Racer**



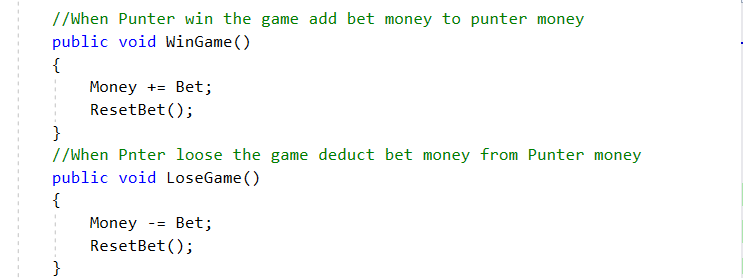
Snapshot of one Punter Ali



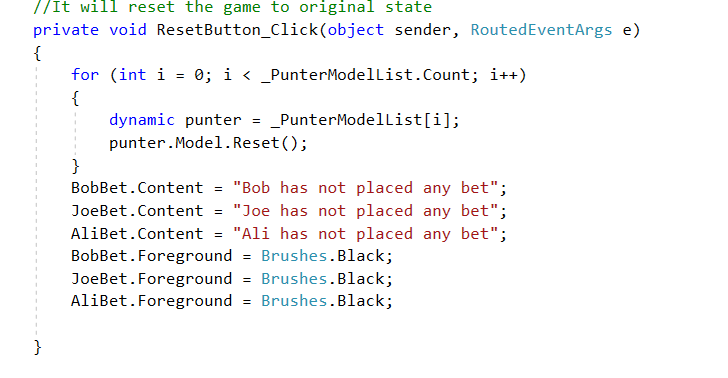
Tortoise Class:-



Code for Adding and deducting bet money from Punter:-

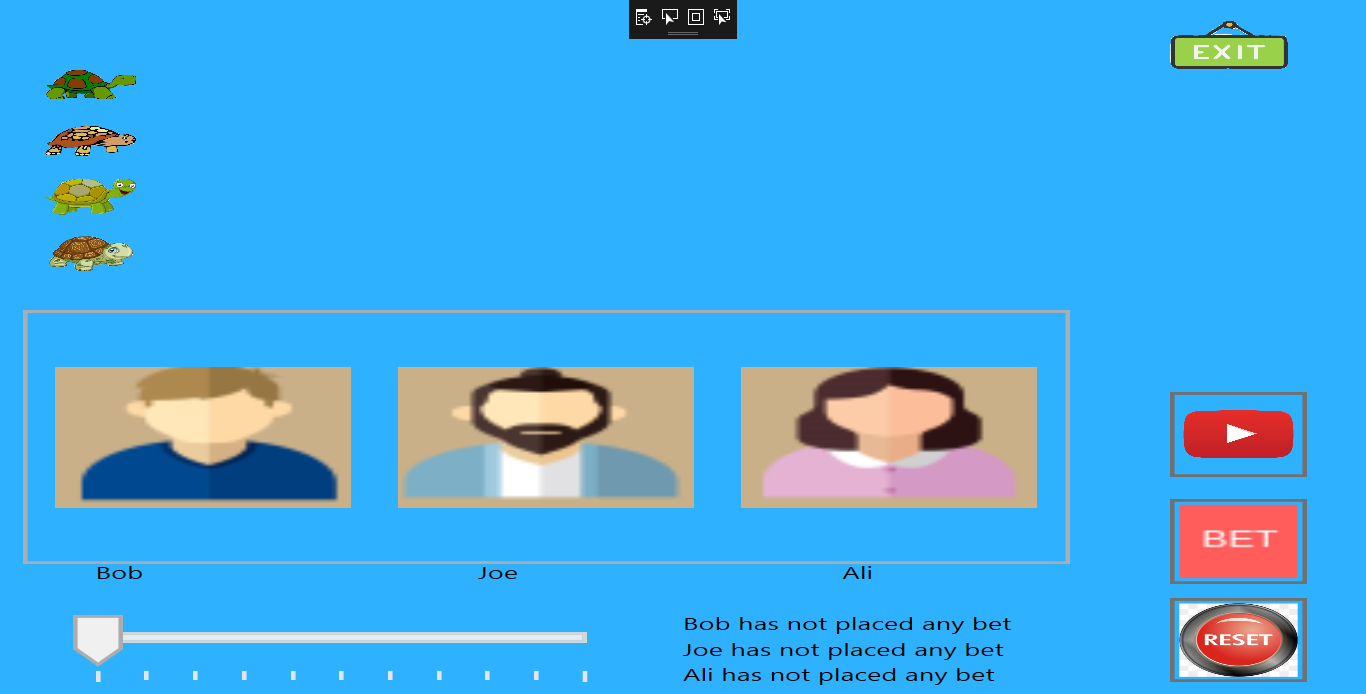


Code for Reset the Game

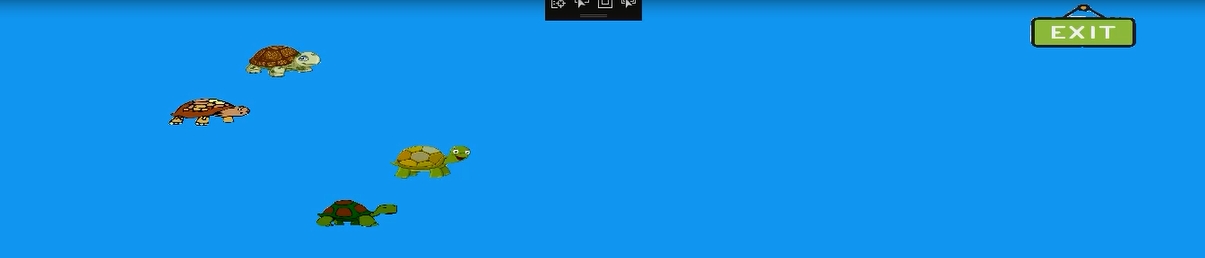


User Interface Snapshot

Main Screen:- Home Page of Game



Racing :-



Bet and Busted :-

