Dice Rolling Game

This is an exercise on real time applications in UI using async await and task.

This code is supposed to run continually and keep rolling dices, until it lands on two 6's.

The UI will show two dice that "roll" → by switching through images until either the Users interrupts or the two 6's are rolled and counts the number of rolls needed.

However, it currently doesn't work! Study the code and implement the tasks below in the **DiceGame.cs** and **MainWindowViewModel.cs** file.

We have provided code for you in the following classes (The files you will need to change are bold):

- Models/
    - ImageHelper.cs
    - Dice.cs
    - **DiceGame.cs**
- ViewModels/
    - **MainWindowViewModel.cs**
- Views/
    - MainWindow.axaml

Task 1 - Setup Images

Set up the UI to show the two dice in the MainWindowViewModel by:

1. In the Constructor, fill the List of images with the dice images from Assets/ by using the ImageHelper class from /Models/
2. Set both Dice to show a One

Task 2 - Dice Game

Implement the following in the DiceGame class

1. Fill in the Start method that should check if the game is not yet running, then set the _isRunning flag and finally start the DiceRolling Task.
2. Implement the Stop function using the _isRunning flag.
3. Implement the Dice Rolling Task that continuously rolls two Dice (from the Dice.cs file) until either the User stops (by checking the _isRunning flag) or both dice show a six (by using the EqualsMax method from the Dice class). Then call the Update method from the ViewModel.

Task 3 - Update the View

Finally finish the ViewModel by

1. Implementing the Start and Stop commands by calling the corresponding methods in DiceGame
2. Update both Dice Images and the Result TextBox on the UIThread by using:

```
Dispatcher.UIThread.Post( () =>

{



}
```

See https://docs.avaloniaui.net/docs/guides/development-guides/accessing-the-ui-thread