CSIS 1175

Due date: Jun 29 (Tuesday), 2021, 17:00

Submission

You need to submit **ONE** files *DigitalInvader.cs* to blackboard by the due date. **NO LATE SUBMISSION** will be allowed. Zip* this file in a single zip file for submission.

You may submit your work multiple times, but only the last submission will be graded.

*Tutorial for zipping a folder:
https://www.youtube.com/watch?v=Ipn-T5Um3d4 for Windows
https://www.youtube.com/watch?v=V0wkG6zOpjA for Mac

File Needed

`DigitalInvader.cs`, which is an incomplete program, is posted in blackboard for you to start.

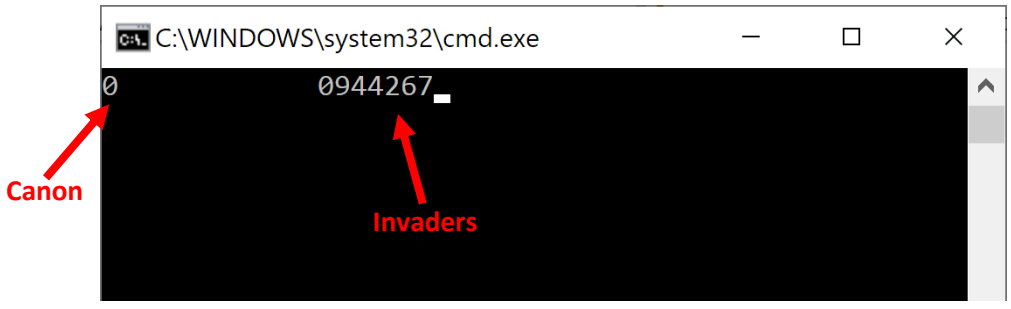
Attention

You need to read and study the program **`DigitalInvader.cs`** carefully before you start the implementation.

Description

You are asked to write a C# program to implement a game **`DigitalInvader`** revised from the classic Casio Calculator Digital Invader Game. How to play the original Casio Calculator Digital Invader Game can be found in the video https://www.youtube.com/watch?v=q_y_DD1m8fw.

The following diagram is a snapshot of the **`DigitalInvader`** game in this assignment:



The game runs as follows:

- When the player presses the key "**`j`**" in the keyboard, the canon will be updated from 0, 1, 2, …, 7, 8, 9, 0, 1, 2, 3… … repeatedly.
- An invader will be generated every one second.

- When the player presses the key "**k**" in the keyboard, the leftmost matched invader, if any, will be killed and removed.
- The game will run continuously until there are 10 invaders not killed by the player.

In this assignment, an incomplete C# program is given as shown at the end of this document. Your tasks are as follows:

- Implement the **`UpdateCanon()`** method:
- Implement the **`Shoot()`** method;
- Implement the **`Play()`** method;

The purpose of the above methods is stated in the comment of the program.

Grading

Correctness of the program: 80%
Programming style/comment/clarity: 20%

Additional Requirement, Assumptions and Hints

1) You must not change any existing methods or variables already exist in the program. You can only complete the methods UpdateCanon, Shoot and Play, or add any methods/variables when necessary.
2) The output of the program should be **EXACTLY** the same in response to the user's input as shown in the sample runs above.
3) You may assume that the user must input relevant and valid input when using the system.
4) The following code will save the second of the current time to the variable s. For example, if the current time is 17:34:28. The value of s is 28.

```
DateTime now = DateTime.Now;
int s = now.Second;
```

Sample Run

A video showing the sample run of the program can be found in Blackboard.

```
using System;
using System.Threading.Tasks;
using static System.Console;

namespace CSIS1175Asgn2
{
    class DigitalInvader
    {
        static int canon = 0;                    // an integer, 0-9, to represent a canon
        static string invader = "";              // a string to presents invaders
        static Random random = new Random(42);   // a Random object generating an invader

        public DigitalInvader()
        {
            // Constructor to initialize the game
            Setup();
        }

        private void PrintCanon()
        {
            // An instance method to print the canon
            Console.SetCursorPosition(0, 0);
            Console.Write(canon);
        }

        private void PrintInvader()
        {
            // An instance method to print the invader
            Console.SetCursorPosition(10, 0);
            Console.Write("{0,10}", invader);
        }


        private void SetKeyListener()
        {
            // An instance method to create the listener
            // When the key "j" is pressed, UpdateCanon() will be called automatically
            // When the key "k" is pressed, Shoot() will be called automatically
            ConsoleKey key;
            Task.Factory.StartNew(() =>
            {
                while (true)
                {
                    key = Console.ReadKey(true).Key;
                    if (key == ConsoleKey.J)
                    {
                        UpdateCanon();
                    }
                    else if (key == ConsoleKey.K)
                    {
                        Shoot();
                    }
                }
            });
        }

        private void Setup()
        {
```

```
            // An instance method to setup the game
            Console.CursorVisible = false;
            Console.Clear();
            SetKeyListener();
            PrintCanon();
        }

        private string GenerateNextInvader()
        {
            // An instance method to generate the next invader
            return (random.Next() % 10).ToString();
        }

        private void UpdateCanon()
        {
            // An instance method to update the Canon
            // After calling SetKeyListener, this method will be automatically called
            // when "j" is pressed
            // The canon will be updated through 0-9 and then back to 0

        }

        private void Shoot()
        {
            // An instance method to shoot an invader, if any
            // After calling SetKeyListener, this method will be automatically called
            // when "k" is pressed
            // The leftmost matched invader will be removed.


        }

        public void Play()
        {
            // This is an instance method to start the game
            // This method will run continuously until 10 invaders not killed
            // Before termination, an invader will be generated every second.



        }

        static void Main(string[] args)
        {
            DigitalInvader invader = new DigitalInvader();
            invader.Play();
            WriteLine("\nThank you for playing Digital Invader!");
        }
    }
}
```