COMP 340 – HW 3

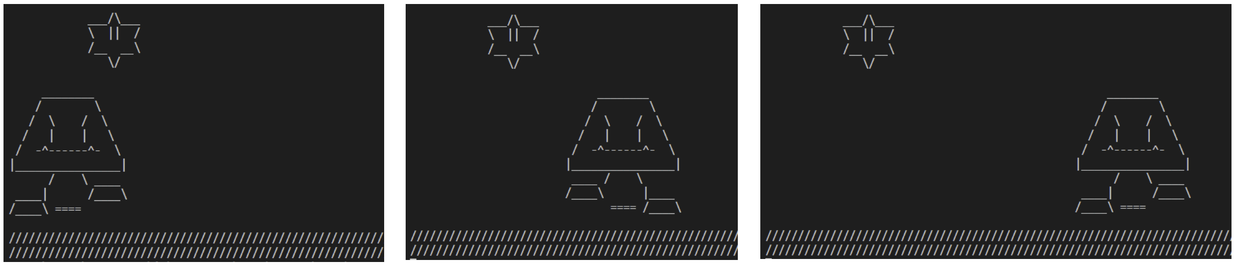
Due date: 03/16/2020 11:59 pm

Advanced Goomba Walking Project

***This homework is a programming project.**
***You need to submit your C# file on Blackboard.**
***While working on the project, read this document THOROUGHLY.**

Project Description

In this project, you will implement an advanced Goomba walking project. This project will give an animation result where Goomba's feet are raised while walking. Several frames of the animation are given below.



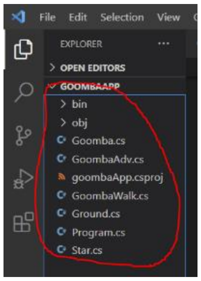
You will add work on Goomba project that we have done so far during the class. This document assumes that you work on the *regular* Goomba project, but it is also OK to work on the Paragoomba project.

Project Guideline

Follow the below guideline to write an advanced Goomba project.

Part 0. Setup Programming Environment

1. Download the required file (`GoombaAdv.cs`) from Blackboard and save it under a folder where you have a Goomba project.
2. In Visual Studio Code, when you open the folder where you save the `GoombaAdv.cs`, you should see a group of classes like the below. (If you work based on the Paragoomba project, you will also have `ParaGoomba.cs` in the group.)



3. Open `GoombaAdv.cs`. You will see that there is a basic structure of GoombaAdv class and GoombaAdv class extends Goomba class. You will complete `GoombaAdv.cs` by following the below guideline to have the advanced Goomba animation.

Part 1. `SetSprite()` of GoombaAdv class

1. Inside `SetSprite()` of GoombaAdv, you will write a code that stores Goomba images to `goombaSpriteLeftFoot` and `goombaSpriteRightFoot`.
2. You can use Goomba images stored in `GoombaSprites.txt` on Blackboard.
3. There are two Goomba images in `GoombaSprites.txt`. Goomba image that raises left foot will be stored to `goombaSpriteLeftFoot`, and Goomba image that raises right foot will be stored to `goombaSpriteRightFoot`.
4. Remember how we stored star image to `starSprite` in Star class and goomba image to `goombaSprite` in Goomba class.

Part 2. DrawSprite() overriding

1. Because Goomba's `DrawSprite()` displays a single goomba image on a console window, you will override the `DrawSprite()` in a way that it displays two goomba images one by one (ex. left foot → right foot → left foot → …).
2. Open `Goomba.cs`. Add `virtual` keyword to `DrawSprite()` (i.e. `public virtual void DrawSprite()`…).
3. In `GoombaAdv.cs`, add `override` keyword to `DrawSprite()` (i.e. `public override void DrawSprite()`…).
4. In `Goomba.cs`, change **`private`** `int posX` to **`protected`** `int posX` so that GoombaAdv can use `posX` freely. (You may need `posX` in GoombaAdv's `DrawSprite()`.)
5. Complete GoombaAdv's `DrawSprite()`. In `DrawSprite()`, you need to write a code that displays two different Goomba images (left-foot image & right-foot image) in order whenever `DrawSprite()` is called in an animation.
   (*Don't forget that your Goomba also should move to right then to left.*)
6. Feel free to add variables and methods in GoombaAdv class. However, you CANNOT add variables and methods in other classes.
   ***Hint:** You may add a variable in GoombaAdv that increases the number whenever `DrawSprite()` is called (ex. 1 → 2 → 3 → 4 …). Then, when the variable has an odd number and even number, you may display `goombaSpriteLeftFoot` and `goombaSpriteRightFoot`, respectively.

Part 3. Advanced Goomba animation

1. Open `Program.cs`. Create an object of GoombAdv class (you can set up the speed value as you want).
2. Put the GoombaAdv object into `gWalk.StartAnimation()`.
3. Remember how we put Goomba object (or ParaGoomba object) into `gWalk.StartAnimation()`.
4. Run the advanced Goomba project using `dotnet run` and enjoy the advanced Goomba animation!

Submission Guideline

1. Upload the completed C# file `GoombaAdv.cs` on Blackboard.
2. After you submit it, DOUBLE-CHECK whether you've submitted the correct file on Blackboard.