MSIM 408/508 & ECE 407/507  
Introduction to Game Development

Programming Assignment Three

**Assigned March 3, Due 12:00 PM March 31**

### 1. Overview

Electronic games including computer games and video games started humbly with 2D games, i.e., games that don’t involve any 3D objects and complex lighting computations. Two-dimensional (2D) games mainly use sprites to represent various game objects; history has witnessed a wide range of extremely popular 2D games, such as Tetris and Nintendo Super Mario games. Unity initially only provided 3D game development tools and introduced 2D game development capabilities in Unity 4. To some extent, 2D games are treated as a special case of 3D games with z coordinates of 0 in Unity.

In this programming assignment, you will first study the 2D Platformer game (shown in Figure 1) developed by Unity and then change it to a 2D game that is similar to the Nintendo game Super Mario Bros, which is one of the most popular video games in history. You will obtain the game graphical assets from various sources, design the game scene, and develop the game mechanics.

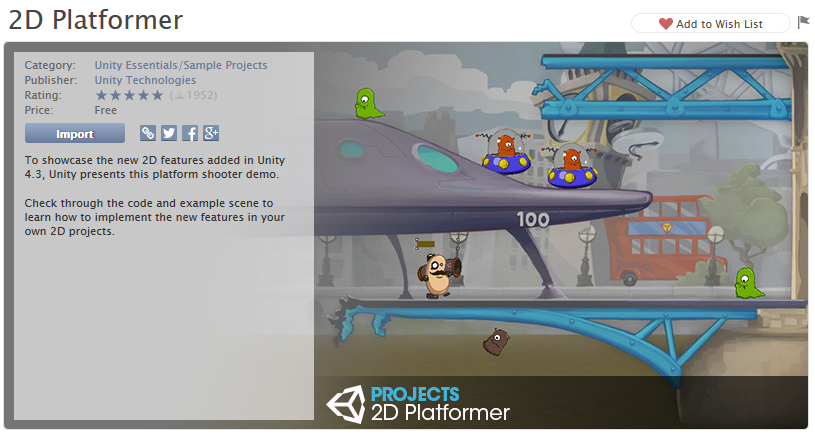


Figure 1. 2D Platformer by Unity

### 2. Objectives

Upon successful completion of this programming assignment, you shall possess the following new skills in addition to the ones you have learned so far in this course.

* Understand 2D sprites and animations.
* Use Unity Sprite Editor.
* Use image editing tools.
* Understand 2D game physics.
* Understand layers and sorting layers in Unity.
* 2D character movement and animation techniques.

### 3. Tasks

The following tasks are required.

1. Run the 2D Platformer game and understand the overall game architecture and techniques used in the game.
   1. The background contains a number of different sprites and a technique called background parallax scrolling is used by the game. Explain the concepts of parallax and layers (and Order in Layer) and why they are useful for generating more realistic graphics.
   2. The game uses two types of techniques for sprite animations. Explain these two types of sprite animations with examples from the game.
   3. For each script, list its references and explain its main functionalities.
   4. Discuss different types of colliders used in the foreground (foregrounds) and different types of physics materials.
2. Create your new game Super Mario Bros based on the game 2D Platformer.
   1. Generate a new folder named “Super Mario Bros” and save all your new assets in this folder and its sub-folders.
   2. If you have not played Super Mario Bros before (that I can hardly believe), read some background information of Super Mario Bros from the following resources.
      1. Nintendo Mario: <https://mario.nintendo.com/>
      2. Wikipedia Super Mario [1]: <https://en.wikipedia.org/wiki/Super_Mario>
      3. YouTube: Super Mario Bros (1985) Full Walkthrough: <https://www.youtube.com/watch?v=rLl9XBg7wSs>
      4. Wikipedia Super Mario Bros. [2]: <https://en.wikipedia.org/wiki/Super_Mario_Bros>.
   3. Re-create a 2D side-scrolling platformer game that is a much simplified version of Super Mario Bros (1985) [2], as shown in Figure 2. It is a single player game with Mario as the protagonist. The objective is to race through the Mushroom Kingdom, survive the main antagonist Bowser's forces, and save Princess Toadstool [2]. The player moves from left to right in order to reach the flag pole at the end of each level. The game consists of eight worlds with four sub-levels called "stages" in each world [2]. Students enrolled in MSIM 408/ECE 407 must develop a minimum of one level (stage), while students in MSIM 508/ECE 507 must develop at least two levels (stages). Each level (stage) has single-exit objective, which must be reached within a time limit and lead to the next sequential level (stage).  
        
      

Figure 2. Super Mario Bros.

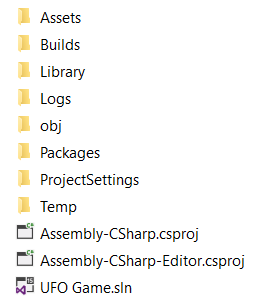
* 1. Original game elements. The player controls Mario, which is the protagonist of the game. Mario’s enemies (antagonists) include Koopa Troopa (also known as Bowser) and Goomba. Mario can pick up various power-ups, such as Super Mushroom, coins, Fire Flower, invincibility star, and 1-Up mushroom. Background includes bricks, trees, bushes, clouds, flat pole, and castle or fortress. You can obtain necessary sprites from the Internet for this project (and you need to state the source of your assets) or you can create them from scratch using digital content software such as Photoshop, Gimp, and Adobe Illustrator. The following are some possible resources.
     1. <https://www.spriters-resource.com/snes/smarioworld/>
     2. <http://www.mariouniverse.com/sprites/nes/>
  2. Minimum Requirements. Your game must consist of the following elements at minimum.
     1. World 1-1: The opening section of Super Mario Bros. was specifically designed in such a way that players would be forced to explore the mechanics of the game in order to be able to advance. Rather than confront the newly oriented player with obstacles, the first level of Super Mario Bros. lays down the variety of in-game hazards by means of repetition, iteration, and escalation [2]. The first level of your game must bear strong similarities to World 1-1 in the original Super Mario Bros. Students in enrolled in MSIM 508/ECE 507 must develop one additional level with similar complexity, but sufficient difference.
     2. Mario, who kills a Goomba by jumping on top of it but is killed by the Goomba if otherwise the contact between them is in a different way, e.g., horizontally.
     3. Goomba, which is a type of enemy that does not have any weapons and runs back and forth between obstacles. Goomba must be generated randomly at different locations and times.
     4. Two types of coins. One type of coins is hidden in the bricks, while the other floating in the sky (still). Collecting two types of coins results in 200 points and 100 points each, respectively. A new life is awarded when 2000 points are collected.
     5. Proper UI with number of points, number of lives, and remaining time.
     6. Proper background music and sound effects.
     7. Generate animations and particle systems to model various movements and visual effects in your game.
  3. Optional elements and tasks. You can additional elements and implement additional tasks for extra credits, which will be determined by the instructor based on complexity and performance. An incomplete list of examples is shown below.
     1. Additional power-ups, e.g., Super Mushroom, coins, Fire Flower, invincibility star, and 1-Up mushroom.
     2. Koopa Troopa (aka Bowser)
     3. Additional levels
     4. New game items and game plays that are designed by you and must be put a scene that is different from that fulfills the minimum requirements.
  4. Write C# scripts to develop the mechanics and logic for your game by reusing and modifying the scripts in the 2D Platformer game.
  5. Create a folder named “Builds” inside your Unity project folder (at the same as Assets folder) and create a standalone PC build of your game in the “Builds” folder, Figure 3.

Figure 3. Creating builds of your game

### 4. Code (65 points)

Exercise good coding practices, such as naming variables meaningfully and properly, including comments of key classes, variables, functions, and code segments, and using proper indentations and spacing between tokens (e.g., variables, operators). Points will be deducted for lack of comments or bad formatting (e.g., no indentation or large blank spaces in the code). The following is the grade allocation for different tasks or functionalities of your program.

* Game scenes: 25 points
* Gamy play: 40 points

### 5. Report and Video (35 points)

The report should contain at least the following sections and use the same formatting in this assignment description or other professional formats. Points will be deducted for improper formatting.

* **Introduction (2 points)**: overview of the assignment, such as background, objectives, and brief summary of what you have done for this assignment. Don’t repeat the assignment description verbatim.
* **Game Design and Implementation (10 points)**
  + Briefly describe the key components in your game.
    - Scenes
    - Key sprites, animations, and prefabs
    - Other key assets or elements in your game
  + For each script: describe its functionalities and its references.
  + User controls
* **Results (15 points)**: Describe what you have done for each task specified in Section 3: Tasks. Include results and necessary screen captures for each task. Answer the questions in the task if any.
* **Conclusion and Discussion (3 points)**: Discuss your accomplishments, learning outcomes, and difficulties and how you can do better for this assignment.
* **Recorded Videos (5 points)**. Produce a video that demonstrate a full play of your game.

### 6. Deliverables

### You need to compress your entire Unity project, report, and screen recording videos and compress them into one single file and name it following the requirement specified in the lecture "Course Introduction", Slide 43. Submit the compressed file to Blackboard.

### 7. References

[1] Wikipedia. (2019). *Super Mario*. Available: <https://en.wikipedia.org/wiki/Super_Mario>

[2] Wikipedia. (2019). *Super Mario Bros.* Available: <https://en.wikipedia.org/wiki/Super_Mario_Bros>.