



Major Project Fall 2020
COMP-3770
Game Design, Development, and Tools
School of Computer Science
University of Windsor



Due: Monday, December 14, 2020 by 11:59pm.

Note: To submit your project, in your project folder, add a folder called **Documentation**. Add a file called **Contributions.txt** that outlines the work done by each member. If there is disagreement, each member may add a self-assessment. Add a file called **Readme.txt** which includes any instructions for the marker, if necessary. You may wish to add other documents here such as screenshots, or .docx files that highlight or explain your work. **Rename your project folder to “Major Project – Name” which makes it clear which assignment this is and who made it.** Then your team leader should upload your project to Blackboard (one upload per team).

Note 1: You may work in **groups of 3 - 5** (be sure to indicate team members in your submission). Sorry, due to class size, no 1-2 person projects. Also 5 is a strict upper limit.

Note 2: Unity 3D student version has support for teams of up to 5.

For this major project, you are going to make a side-scrolling 2.5D platformer video game, similar to Super Mario. This project will have several small research components (how to do specific, simple things not discussed in class). All game elements must be locked to the $z = 0$ plane.

IF YOU WANT TO DO SOMETHING OTHER THAN WHAT I AM PROPOSING HERE, YOU MUST DISCUSS WITH ME ABOUT IT FIRST. I WILL REQUIRE SUFFICIENT DOCUMENTATION OF YOUR PLAN IN ORDER TO ACCEPT YOUR PROPOSED PROJECT. THIS INCLUDES A LAYOUT OF HOW THE GAME WILL FLOW BETWEEN SCENES, A DETAILED DESCRIPTION OF THE GAME YOU PLAN TO MAKE, AND AN IN-GAME GUI LAYOUT PLAN. IF YOUR PLAN IS INSUFFICIENT, I MAY PROPOSE SOMETHING ELSE, ADD MORE COMPLEX ELEMENTS TO YOUR PROPOSED GAME, OR DECLINE YOUR PROPOSAL ALTOGETHER.

This project is worth 40% of your final grade, and will be graded in the following manner:

1. [2 percent of your grade] Your player character must be configured such that it can interact with its environment in an appropriate way (including all platforms, enemies, and boundaries). A capsule is sufficient to model the player character. The character must be able to double jump (hitting *space* should make the character jump), switch direction in mid-air, and run (holding *ctrl* must make the character run). The character must be able to jump on top of (most) enemies in order to destroy them, and collisions with enemies from the side of the character should be damaging to the character. It is up to you to determine how the player character takes damage, how the player loses, and what happens if the player loses. Progress through the game's levels must be left to right, but the player **character should be able to move in any direction**. The main camera must follow this player character in a reasonable manner, and the main camera must never move left. The player character must never be able to move off-screen.
2. [5 percent of your grade] You must have 5 different types of enemies, each with different behaviours. The behaviours of 2 of the enemies are stated at the end of this document. The remaining enemies are up to you. The behaviours of the enemies you design must be sufficiently

complex, different, and interesting to get full grades here. Be creative. Perhaps they patrol in an area or explode when they die. This is entirely up to you, but you must make it sufficiently interesting for full marks. These enemies must all be turned into prefabs that can be spawned in through code.

3. [2 percent of your grade] You must have collectable items in the levels. These collectables are considered currency, the number of collectables collected by the player must be permanently saved and displayed in the main menu and in game. This 2% strictly refers to the generation or placement of these objects and their behaviour when collided with. Collectables should be prefabs so they can be spawned in through code.
4. [4 percent of your grade] You must have 5 different types of powerups, each providing different functionality or behavior to the player character. The powerups must be concrete objects that can be collected by the player. The player character must change colours (or textures) depending on which powerup is currently active. For instance, one powerup may provide a shield that allows the character to take an extra hit before dying, and another might make the character shoot a projectile when *alt* is hit. Be creative. The powerups must be sufficiently different and interesting for full marks. Powerups should be prefabs so they can be spawned in through code. Two of the powerups are specified at the end of this document, and the remaining three are up to you.
5. [2 percent of your grade] You must have platforms that move in your game. Some must move horizontally, and some must move vertically. You must also have platforms that do not move, but the character should be able to jump on them and enemies/powerups/collectables should spawn on them. Lastly, you must have death zones that do damage or destroy the player character on entry.
6. [14 percent of your grade] Level generation. Each level must be procedurally generated. This means that as the player moves in the level, the next zone that the character reaches is generated by code. Positions of the enemies, powerups, collectables, platforms and any other game elements will be generated using code. After 10 procedurally generated sections spawn, a “level complete” zone must spawn. Within a given section, some game elements may come as packaged prefabs rather than just spawned on their own. For instance, rather than just spawning a platform, you may have a prefabricated platform that has 5 collectables and an enemy spawn point on it, where the enemy that spawns is randomly selected. When the player is halfway through a given section, the next one must be constructed. When a section is completely off-screen, it must be destroyed. Further, you must have several different floors for the zones, such that there are gaps that the player character can fall through and die. Each gap created MUST be traversable given the physics of your game.
 - a. Easy way out #1: Rather than true procedural generation, you can create prefabricated level sections and add them in as the player moves through the level. Powerup and enemy spawns must still be procedural but setting spawn points for these elements is sufficient. If you take this option, you will get a maximum of 8/14 on this part of the project. You must have at least 10 separate prefabricated sections for the entire 8/14. After 10 randomly selected prefabricated level sections spawn, a “level complete” zone must spawn. When the player is halfway through a given section, the next one must be constructed. When a section is completely off-screen, it must be destroyed.
 - b. Easy way out #2: Rather than true procedural generation, you can create entire levels by hand. If you take this option, you will get a maximum of 5/14 on this part of the project. Levels must be sufficiently long, complex, and interesting. You must have a “level complete” zone at the end of each level.
7. [2 percent of your grade] Flow between scenes. For full grades here, your game must have a splash screen scene (the first scene that automatically loads when played showed the name of the game), a main menu scene, at least 3 different levels that differ by the types of enemies and powerups that exist in them, and a scene that contains a description of the game. The main menu

scene must provide buttons to navigate to the various levels, and also show the current total count of collectables that the player has. When the player succeeds or fails at a level, you must handle these cases as you feel appropriate.

8. [2 percent of your grade] In-game display. You must have an in-game display that shows the following things:
 - a. [0.5/2] Current total collectable count, and a count of enemies destroyed in this level attempt.
 - b. [0.5/2] Current level that the player is on.
 - c. [0.5/2] A button that pauses/resumes the game.
 - d. [0.5/2] A button that exits to the main menu.
9. [2 percent of your grade] Presentation or write-up. You must schedule an appointment with me prior to the deadline of the project to present to me what you've created and describe to me the different game elements and techniques used. Alternatively, you can create a write-up that describes all of the game elements and techniques used, in sufficient detail. Keep and discuss all of your prototyping (drawings by hand, early unused scripts testing specific game mechanics) for full marks.
10. [5 percent of your grade] This is for you to add your own twists (a "boss" enemy character, extra powerups or collectables, audio) to the game, outside of this specification. You must not go directly against anything specified here (for example change how your camera functions), but you can add extra elements to the game. Make this game your own.

Mandatory Enemies:

1. The first enemy must shoot projectiles that are affected by gravity but bounce off of boundary elements and platforms. For this, you will have to research how to make something bouncy in Unity. The firing rate (how often projectiles are shot) and movement of projectiles is up to you. The projectiles should last for 5 seconds. Projectiles should be always shot in the direction of the player character.
2. The next enemy teleports to the location of the player character every 5 seconds, unaffected by gravity. One second after teleportation, the enemy must create an explosion that applies force to the player character and somehow negatively impacts the player character (reduces health, removes a powerup, etc.). The amount of negative impact must be proportional to the player character's distance from the explosion. During the 1 second phase after teleportation, the explosion's radius must be visible. This enemy is immune to all damage during the 1 second phase after teleportation. The enemy must be transparent during this phase to indicate immunity. There should be a visible explosion. You must research particle systems in order to figure out how to make a visible explosion.

Mandatory Powerups:

1. One powerup must allow the player to click on a position and teleport to it. The player must be invulnerable to any damage for 2 seconds after teleporting, and teleporting must reset the double-jumping mechanic (that is, the player should always be able to double jump after teleporting). There must be 5 charges of this powerup, and a visual indicator of how many charges of the powerup remain.
2. One powerup must allow the player to click in any direction and shoot a laser beam from the player character that kills any enemy in its path. It should be able to go through multiple enemies and all other objects in the world. This powerup must last 30 seconds before being disabled.