Game Development - Assignment 1

Overview

The main goal is that we can have a simple platformer with two levels created with *Tiled 0.17 (not the latest one)*, having a main character moving around while being able to save and load game's state.

Content

Using any combination of art done by yourself and free art found on the internet (including old arcade games), create a simple platformer where the main character should be able to navigate mainly through jumping around. Then each group must pick another feature to make navigation more interesting (like double jump, sliding, hover, bouncing on walls, etc.). Make sure to have a *death transition animation*.

The main control method should be keyboard, even if optional keyboard controls is encouraged. Mouse can be ignored for this assignment. Remember to have **vsync on**, to stabilize the game to 60 fps.

The level should **not** have any UI or enemies. Still, it should require some skill to complete it. The most common way to achieve it is to add fall pits where the player can fall if it does not jump with good timing, but it could be any other challenge (for inspiration, check Super Meat Boy).

The Game should support **two** different levels, so the process of loading and unloading the level should work well to avoid memory leaks. Each level should support, at least, **one background layer** using the parallax effect (not hardcoded but defined in Tiled). Player starting position and tile collisions should be defined inside the Tiled editor.

Minimum debug Keys

F1 Start from the very first level	**F2** Start from the beginning of the current level
F5 Save the current state	**F6** Load the previous state (even across levels)
F9 To view colliders / logic	**F10** God Mode

Submission rules

The delivery must be a **Win32 Release** zipped in its folder inside *"First Assignment"* named after your game (e. G. Ultra_Mario_Bros.zip):

1. Delivery should contain only the minimum assets needed to run the game that should be compiled in Release. Only the required dll to execute the game should be there.
2. Your code with a well commented, and small commits on your changes over time.
3. There must be a text file called "README.md" containing info about the game, authors, a **link to the github** repository and a *license*. Any special instructions for execution should be included in this file, as well as any system that you think could add to the innovation grade. Also remember to add a Credits section to mention all source of art / audio you used.
4. The repository under github.com must contain a copy of the build under the Release section.

The assignment must be submitted before **October the 14th 23:58** *(folder closes automatically)*

Grading Criteria

To **accept** a submission for grading, it must comply with:

1. It followed the submission rules stated above.
2. The code compiles and uses only english.
3. It should be **original**. If code is found to be copied across teams, it won't be accepted.
4. The game did not crashed while testing.

Once accepted, the criteria is as follows:

* 60%: C++ code is clean, well structured and easy to read.
* 20%: Main character can easily navigate the levels and collide with the environment.
* 20%: Navigation is fun and with a balanced degree of challenge. Player animation, music+fx all together make for a good user experience. Proper transitions for load level / death.

Note: *In case of a great imbalance in work between team members, teacher can decide to downgrade an individual score.*

Note: *Remember that any experimental system could potentially impact the innovation grade. If you think you have some system that falls in this category, clearly mention them in the README.*

Helpful Coding article
http://higherorderfun.com/blog/2012/05/20/the-guide-to-implementing-2d-platformers/

Helpful Links with free 2D art
https://opengameart.org
https://www.gamedevmarket.net
http://www.gameart2d.com
https://www.spriters-resource.com/
http://spritedatabase.net/
http://scrollboss.illmosis.net/mainmenu.php