

Game Development - Assignment 1

Overview

The goal is having a simple platformer with levels created with *Tiled*. There must be a main character moving around while being able to save and load the game's state.

Content

Create a simple platformer game with the following elements:

- (20%) Tiled TMX map loading and drawing (orthographic)
- (20%) Map collisions detection (platforms)
 - Using Box2D
- (20%) Map navigation: player movement and jumping
 - Player walk/jump/die animations
 - *[Optional] Player double jump (*)*
- (15%) XML config file loading - read modules and entity parameters from XML
 - Player parameters (initial position, speed, acceleration, ...)
 - Texture and audio paths (all including animations)
- (15%) Load/Save game state using XML file
 - Player Position
- (10%) Screens
 - Logo screen with fade-in fade-out
 - Title screen (press ENTER to start)
 - Ending screen (on dying)
- *[Optional] Second level (one level is mandatory) (*)*
- *[Optional] Background with parallax (*)*

Entity System: The Player must hereby from a base Entity class and an EntityManager class must manage it (Initialize, include in a list, Update, Draw, CleanUp...)

Player controls (keyboard):

- WASD for movement
- SPACE for jump

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](#)). Make sure jumping is smooth and can go through platforms when jumping from below. Also, make sure you have [a good camera](#). The game should support **at least one level**, a second level is optional, in any case, the process of loading and unloading the levels should work well to avoid memory leaks.

* *Each optional optional feature adds 0.5 to the grade (with a top of 10)*

** *Remember to have **VSYNC ON**, to stabilize the game to **60 fps**.*

DEBUG keys

Game should include a set of DEBUG options enabled with the following keys:

- **F1/F2** Start from the first/second level
- **F3** Start from the beginning of the current level
- **F5** Save the current game state
- **F6** Load the previous state (*even across levels*)
- **F9** View colliders / logic
- **F10** God Mode (*fly around, cannot be killed*)



Submission Rules

Each team **MUST** upload on the online campus website (Atenea)to the task “*Assignment1*”

- Their **release build** as a **zip** file
- The URL of the GitHub Project. The build **MUST ALSO** be published in the **Release section of the project’s GitHub page**. The source code will be reviewed from the Release version.

Release folder structure and naming conventions:

```
> Team_Name-Platformer-Alpha.zip    // Game directory
    zipped Platformer                // Game directory
        Assets                       // Assets directory, it could contain
                                     // multiple sub-dirs and files
            Game.exe                  // Main binary for the game
            Xxx.dll                   // All required DLLs to run the game
        LICENSE.md                   // Game license file
        README.md                    // Game detailed info
```

NOTE: GitHub release **MUST** contain detailed information on the current release (new features, improvements...)

Submission **will not** be accepted for grading in case:

- It is not delivered on time
- Build is malformed (included not used files or code not compiled in **Release** mode)
- Build is not available in the GitHub Release system
- Game crashes while testing

Once the delivery is accepted, the **grading criteria** is:

- **100% Checklist points:** Evaluation will consider all points completed from the checklist and all the additional gameplay elements and state of polishment.

NOTE 1: Project organization and naming conventions *MUST* be followed. Not following them will translate into a 50% downgrade of the mark received.

NOTE 2: In case of a great imbalance in work between team members, teacher can decide to downgrade an individual score.

Helpful Links

- [The guide to implementing 2D platformers](#)
- [Open Game Art](#)
- [Kenney • Assets](#)
- [GameDev Market](#)
- [Free Game Assets](#)