Game Development - Assignment 2

Overview

Building on top of the previous assignment, we are going to add enemies with pathfinding.

Content

Expanding the platformer from the previous assignment, we need to add:

- **(30%) Walking enemy type** that can **pathfind** to the player. It is not needed that the enemy can jump (although is encouraged) but it should detect that it can reach the player by normal walking and falling down to other platforms.
- **(30%) Flying enemy type** that can **pathfind** the player, avoiding non-walkable areas.
- **(15%) Load/Save** must consider each enemy state.
- **(15%) Destroy enemies**: jumping onto them, shooting, …
- **(10%) Follow-up submission rules**: Release publication, code organization, deliver on time, debug features

Entity System: Enemies must hereby from a base Entity class and an EntityManager class must manage them (Initialize, include in a list, Update, Draw, CleanUp…)

Enemies behavior: Enemies normally have a range of perception and not react to the player until they are close by.

Optional (Each optional feature adds 0.5 to the grade, with a top of 10):

- *[Optional] Skills with cooldown*
- *[Optional] Audio feedback*
- **DEBUG keys**

NOTE: Any remarkable addition to the game beside the proposed ones could be evaluated with extra points to the Assignment mark.

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 / **paths**
- **F10** God Mode *(fly around, cannot be killed)*
- **F11** Enable/Disable FPS cap to 30

Submission Rules

Each team MUST upload their **release build** as a **zip** file to the folder *"Assignment2"* on the campus website. If more than one file is uploaded, only the last one will be evaluated.

The build **MUST ALSO** be published in the **Release section of the project's GitHub page**, source code will be reviewed from the Release version.

Release folder structure and naming conventions:

```
> Team_Name-Platformer-Beta.zip      // Game zipped
    Output                           // Game directory
          Assets                     // Assets directory, it could contain
                                     // multiple sub-dirs and files
                                     // Assets license files must be near
                                     // the asset file
          Game.exe                   // Main binary for the game (release)
          config.xml                 // Configuration file
          save_game.xml              // Save game
          Xxx.dll                    // ONLY required DLLs to run the game
    LICENSE                          // 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**
  - **Delivered files are wrongly named**
  - **Not compiled in Release mode**
  - **Includes DEBUG temporal files**
- **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.

Grades could be **downgraded to 50%** if not fulfilling the following rules:

- **Assets must follow the Files/Directories naming conventions.**
- **Code must follow the Code Naming Conventions.**
  - Be very careful to use TABs instead of 4 spaces!
  - Be very careful to use only one space between *DataType field;*
- **Directories organization is not the expected one**
- **Release includes assets not used in game**

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

Helpful Links

- Adapting A* to platformers