Game Development - Assignment 3

Overview

Building on top of our previous assignment, we are going to add an intro menu and HUD elements to the game.

Forking from other student's previous code

Students have the option to start from the code from other students just on the moment of delivery on the previous. Be sure to mention this in the README and properly fork the other team's repository exactly at the moment of delivery of previous assignment. Forking late will mean not accepting the delivery.

Content

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

- Intro menu with a background screen and, at least, buttons for "Play", "Continue", "Settings", "Credits" and "Exit". Those buttons need to be responsive on mouse hover and click.
- "**Play"** button must transition (not abruptly) start the game from scratch.
- **"Continue"** button must be enabled only if there is a saved game. Should transition to the last saved game.
- **"Settings"** must send / open another menu where you can, at least, adjust the music and fx volume.
- **"Credits"** should send / open to another menu where you can read the authors and the license.
- **"Exit"** should quit the game.
- There must be a specific music track for the menu and the buttons need to give audio feedback.
- Try making the UI as confortable as possible, with transitions, feedback and animations.

- While playing the game we should have UI for, at least, player lifes, coins, timer and score. This is a base that can be adapted to each game after discussing with the teacher (do you run out of time in your game ? Do you get an extra life after X score/coins ? instead of coins you have stars that speed up the player ? So your enemies each the coins ?)
- There must be some sort of coins that the player must be able to collect thought the game. The HUD must show the amount collected so far. Their effect into the game is up to you. Coins must be respected when saving/loading the game state.
- Player should have some amount of lifes. The game will cycle from the beginning of each level until the player consumes all its lifes. At that point, the game must transition to the intro menu.
- The timer should contain the time so far accumulated from the player while playing the game. This value must be properly recovered when loading a previous game.
- The HUD must not be static! It must attract player's attention when something important happens, like getting to 100 coins to gain a new life, or running out of time to finish the game.
- **Brofiler** should measure correctly the amount of processing time devoted to UI.
- With the delivery you must provide a pdf that contains the UML structure for your UI subsystem.

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)
F8 Debug UI	**F9** To view colliders and pathfinding
F10 God Mode (allow flying around)	**F11** Enable/Disable FPS cap to 30

Submission rules

The delivery must be a **Win32 Release** zipped in its folder inside *"Third 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 (with exception of the UML scheme delivered as a **pdf** file that describes the UI system).

2. Your code good .gitignore and well commented, small commits on your changes over time.
3. The repository under github.com must contain a copy of the build under the Release section.
4. There must be a text file called “README.md” containing: info about the game, authors, a **link to the website (that contains 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.

The assignment must be submitted before **December the 17th 23:58** *(folder closes automatically)*

Game’s website

Each team must create a single page website that contains:

1. Short introduction to the game and it’s context (university subject).
2. Team photo and quick explanation of the individual work of each member.
3. Short explanation of the main core subsystems, including any innovation done out of the core requirements.
4. Link to the code repository and last release to download.
5. One minute video that shows the gameplay.
6. Instructions to play
7. License.

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:

- 50%: C++ code is clean, well structured and easy to read.
- 30%: Intro menu is easy to use, responsive and transitions properly between all its screens. The HUD is clear and attracts the user attention when is required.
- 20%: The new HUD and it’s mechanics game the game interesting and **replayable**.

***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.*

Useful Links

Unity Asset Store Free UI Art
UI Art Atlas inspiration


BUTTONS 6 FOLDER
BUTTONS 1 FOLDER
POTIONS 1 FOLDER
BUTTONS 5 FOLDER
SLIDES 1 FOLDER