UNITY EXERCISE: MARIO GAME

GOAL:

Implement a basic prototype of the arcade Mario Game. Demonstrate your abilities in designing and implementing a complete solution, using best practices, engineering principles, and game development tools of your choice

Please refer to this mario example video (start at 0:01)

Solution requirements

All the design decisions are up to **you**! We simply ask you to adhere to the following notes.

1. The game should be adaptive to different screen aspect ratios.
2. The solution can be submitted with primitive shapes for all visual objects (player, ammo, levels, etc. no custom sprites\shaders are required), yet in this case, make sure to use distinct shapes for every object type. (custom visuals grant extra credit).
3. Make sure the game does not start immediately on launch (have a "menu" type, input-oriented state), has at least 1 distinct level, and can be restarted at any point by returning to the menu.
4. Your solution should work on mobile devices in landscape mode. An Android .apk is required as a part of the submission process.
5. You should pay attention to the performance characteristics of your solution (execution time, memory usage, GC, etc). The solution is expected to be designed using proper software engineering principles, code conventions, and should be properly documented when needed (so we know what decisions were made, and why).
6. Remember; the visuals can be basic, yet the gameplay should be polished.
7. Please use Unity version 2019.4.xx LTS for the purpose of this exercise.

Extra credit (optional, are you up to it?)

1. Architecture designed with FSM
2. Three or more distinct consecutive levels, with increasing difficulty +++
3. Custom visuals and shaders ++
4. Custom soundtracks and SFX +
5. Two-player, same screen local multiplayer ++
6. Leaderboards with player names supplied at the end of the game +

What to Submit

1. Link to a public git repository, containing all the assets (scripts, scenes, etc) that are required to run the exercise. Please include only the relevant folders (no need to send Temp/ or Library/ folders). Also, include any unit tests for testing your code, if available.
2. Working .apk and .exe builds via a public link to any large file sharing system (google drive, s3, dropbox,etc)

3. In case you choose to implement one or more of the bonus tasks, your git repo should include a README.txt with a short explanation about which extra credit was implemented and why you choose it\them over the others.

4. Send the submission links to robin.arias@purplebugmail.net

5. glhf!

Questions?

Feel free to contact us with any questions or clarifications:
Mail: robin.arias@purplebugmail.net

Confidential: The contents of this document are confidential and intended solely for the recipient. Reproduction of, or forwarding to anyone not directly sent this document is strictly forbidden.