SSAD Assignment - 1 (*Mario Game*)

Welcome to your first assignment for SSAD! You will be building a variant of the classic Mario game - https://www.youtube.com/watch?v=zHyO01R4yzQ

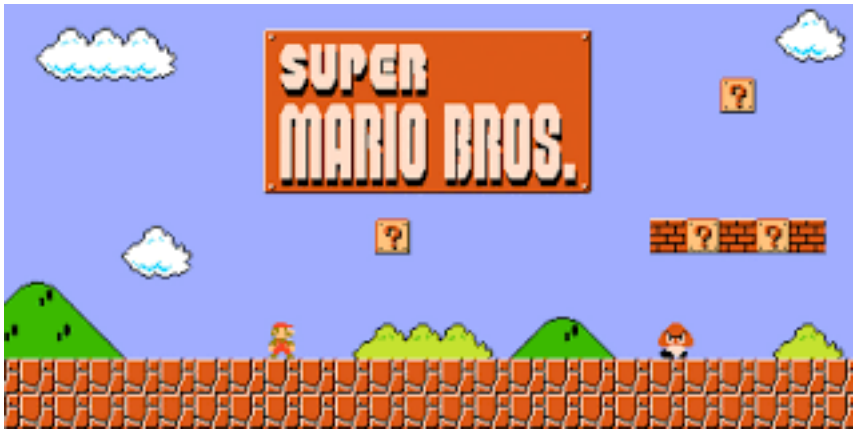

Figure 1: mario_img

Introduction

You have to write a python application (terminal-based) that simulates a basic version of Mario. A tutorial will be conducted soon to give you a brief idea of how your code should be structured. Concepts of Object oriented programming must be present within your code.

Requirements

- Code should exhibit OO concepts (**40 marks**):
  - Inheritance: You could have one person class and have both the enemies and player inherit from it.
  - Polymorphism: Have one obstacle class and *override* various characteristics to exhibit different properties.
  - Encapsulation: Class and object based approach for all the functionality implemented.
  - Abstraction: Intuitive functionality like `move()`, `attack()`, etc, stowing away inner details from the end user.
- Movement (**10 marks**):
  - WAD (directional - left, right, jump)
  - Jump should have a gravity-like effect.
  - Jump on springs or bridges and have different characteristics.

- Obstacles (**30 marks**):
  - Enemies will move left, right automatically
  - *Additionally*, you can have enemies with different speed, different behaviour (chase the player, etc).
  - Boss enemies with extra lives and additional powerups
- Score (**10 marks**):
  - Score for duration of the game
  - Bonus score for coins and gems collected
  - Score increments for killing enemies and boss
  - Display the score and life on the screen
- Background & Scenery (**10 marks**):
  - when moving out of the window, change the scenery
  - Have different scenes in the background, underground, underwater, bridges, etc

Bonus (20 marks)

You will be judged for additional marks if you have implemented the following:

- Color: Have different colors for various objects in the scene
- Sound: Sound effects on bonus collection and/or enemy kills.
- Smart enemies: Enemies that show non-random directed behaviour with jump and different speed characteristics.

Submission

- Please follow a proper directory structure, have different files for different roles (enemies, player, objects).
- Further details will be intimated over Moodle.

Library Usage

- Language: Python3 (`python2` will **strictly** not be allowed)
- Ensure that you bundle your project with a `requirements.txt` that states all the packages used. This will help us track whether you've used certain packages that aren't allowed. Do note that if your submission does not contain a `requirements.txt` file but contains non-standard packages, you will **DIRECTLY** be given a **ZERO** for the assignment.
- It is recommended that you run your code in a virtual environment. This will also make it easier to generate the `requirements.txt` file.
- NOTE: No `curses` libraries (`pygame` and the like) are allowed.
- Allowed libraries:
  - `colorama`

  - `numpy`
- In case of any doubts regarding whether a particular library is allowed or not, post on Moodle to get it confirmed by a TA

Additional Notes

- Plagiarism is a strict NO. If found, you will be given 'F' grade for the course.
- OOP Principles should be strictly followed.
- End the game when all lives are over or User quits by pressing 'q'.
- All the details along with controls for your game should be submitted with the code in a README file.
- It is your game, make it as creative and fun as you can.