CS6.301: Design and Analysis of Software Systems **Spring 2020**

Assignment-1: The Mandalorian

Released: January 9th *Deadline: January 22nd*

Instructions

- This assignment is designed to get you familiar with object oriented programming (OOPS) concepts. You have to use classes and objects to write good and modular code.
- Submit all your code files and a README.md as a zip file, named as ⟨rollnumber⟩.zip.
- The readme should describe your game, its rules, all the functionalities and any cool features! It must act like a quick start guide, describing all the controls and features.
- The deadline is January 22nd, 23:55. This is a hard deadline and no submissions will be accepted after this.
- Start early! This assignment may take fairly long.
- Plagiarism detectors will be run on all submissions, so please do not copy. If found, you would be given a straight zero for the assignment :(

Introduction

Din is a mandalorian living in the post-empire era. He is one of the last remaining members of his clan in the galaxy and is currently on a mission for the Guild. He needs to rescue The Child, who strikingly resembles Master Yoda, a legendary Jedi grandmaster. But there are lots of enemies and obstacles in his way, trying to prevent Din from saving Baby Yoda. Din is wearing classic mandalorian armour and has a jetpack as well as a blaster with unlimited bullets. You've got to help him fight his way through and rescue Baby Yoda.

For this assignment, you need to build an arcade game in **Python3 (terminal-based)**, heavily inspired by Jetpack Joyride where the user will control the mandalorian, move it up, forward and backward, while collecting coins and fighting/dodging its enemies on the way. **Concepts of object oriented programming must be present within your code and the game must simulate a basic version of jetpack joyride.**

The objective of the game is to collect as many coins as possible, fight the obstacles on the way, defeat the boss enemy and rescue Baby Yoda. You are free to use your creativity and design your own rules and enemies. But ensure that your game contains the bare minimum requirements that are mentioned below.

OOPS Concepts (30 points)

Your code must exhibit OOPS concepts, that is is the main objective of the assignment.

- **Inheritance**: You could have one person class and have both the player and the enemies inherit from it.
- **Polymorphism**: You could have one obstacle class and override various characteristics to exhibit different properties.
- **Encapsulation**: The fact that you're using classes and objects should suffice for this! Use a class and object based approach for all the functionality that you implement.
- **Abstraction**: Intuitive functionality like move(), shoot(), attack() etc should be methods within the class. So for instance (no pun intended), if you declare an object 'player' of the person class, you can call the methods player.move() and player.attack(), stowing away inner details from the end user.

Movement (5 points)

- A for left and D for right. W for activating the jetpack and jumping up. And some button for firing bullets at the enemies.
- You have to control the Mandalorian and move it up, forward and backward while collecting coins suspended in the air. When the Mandalorian moves upwards, the jet pack is activated as long as the 'W' key is pressed. When 'W' is released, the jetpack deactivates and gravity comes into play. **There must be a gravity-like effect when the he moves up or down.**

Background and Scenery (5 points)

- The scenery and the obstacles must change as you move in and out of the window. There must be a ground/platform and the sky, and the Mandalorian can't go below the ground or above the sky.
- Lots of coins suspended which the Mandalorian can collect and increase his score.

Obstacles (25 points)

- **Fire Beams**: Beam like structures should appear (like in the figure above) as obstacles. There must be three kinds of beams: horizontal, vertical and some at 45° with the ground/platform. The Mandalorian must ensure to not collide with these beams, else he will lose a life. He can use his blaster to shoot at them and clear his way. (10 points)

- **Magnet**: A magnet should randomly appear on the way, which will influence the motion of the Mandalorian. So if he is in the range of the magnet, he would be continuously attracted towards the magnet. Assume that the magnet causes a constant attractive force in its direction. (15 points)

The Boss Enemy!! (15 points)

- The boss enemy must appear in the end and once the Mandalorian defeats it, he can rescue Baby Yoda and complete the game. The boss enemy is Viserion, the flying dragon that **adjusts its position according to the player** (with respect to the movement along the Y axis). **It should throw ice balls** aimed at the Mandalorian, which he must dodge.
- The boss enemy will have multiple lives, which will decrease when the Mandalorian shoots bullets at it. Once the boss enemy is defeated, Baby Yoda can be rescued and the game is complete.
- Use your creativity and imagination for this one, this should be the most difficult obstacle to clear. The dragon need not look like an exact dragon :P But do look up ASCII arts online, some of them are beautiful :)

Score and Lives (5 points)

- Score must be displayed on the top. You can calculate it as you like, taking into account the number of coins collected. Have score increments for killing enemies and the boss enemy.
- Both time and the lives of the Mandalorian are limited. So the 'time remaining' and 'lives remaining' must be displayed on the top along the score. End the game when all lives are over, or if the user quits by pressing 'Q'.

Power-Ups (15 points)

- **Speed boost**: The speed of the game will increase upon taking this power-up. (For 5 points)
- **Shield**: A shield should appear around the Mandalorian using which, he will not be affected by the enemies and obstacles. The shield will be activated using the 'Space' key and will last for 10 seconds. It will take 60 seconds for shield power-up to refill again after use. (For 10 points)

Bonus (25 points)
That's a 100 points so far! Additionally, you could implement the following to be eligible for some bonus marks.

- Drogon(15 marks):
  - **Ultimate Dragon** : We need a dragon on our side too :)
  - Drogon PowerUp will pop up once in the middle of the game and will be destroyed only on contact with an obstacle or enemy. Check this video out.
  - Marks are mainly for the wriggly effect; the dragon should move in a wave-like manner, continuously emitting flames. More marks for a handsome dragon :)

- **Colour(10 marks)**: Have different colours for different objects in the scene. The colorama library is allowed for this. This should not be very difficult, easy 10 marks :P

Library Usage

- Your game must mandatorily be coded in **Python3**. Python2 is being retired soon :P
- No curses libraries (like pygame) are allowed. **Only libraries allowed are colorama and numpy.** In case of any doubts about whether a particular library is allowed, please post it on the Moodle thread and get it clarified from one of the TAs.

Deliverable

A zip file, ⟨rollnumber⟩.zip containing the code files and README. For instance, 20171010.zip when extracted must generate a folder with the name '20171010'. Inside that folder, all code files must be present. Ensure your code is modular with multiple python files. Please ensure to follow the submission format and submit ONLY .zip files.

Plagiarism is a strict NO, and Puru is running moss xD So please do not copy from anyone. We'd be forced to give you a zero for the assignment :(
It is your game, make it as creative and fun as you can!

Happy coding, and may the Force be with you :)