The Game Project 6 â€“ Adding game mechanics

This weekâ€™s project builds upon the sketch from last week. Before starting this project make sure you have completed all steps from last week and tested your game thoroughly.

Create a copy of your sketch directory from last week and rename it to something like `game-project-6`. Keep your completed project from last week safe as a reference, and make the following changes to the code

in your new game directory.

1. Add a score counter [1 marks]

- create a global variable called `gameScore`
- increment `gameScore` by one each time the character collects an item.
- HINT: you may need an extra conditional to prevent the score being added multiple times
- use the text function to draw the score on the screen.

2. Add a flagpole [1 marks]

- We need to add an end to your level. I have chosen a flagpole but you can chose according to the theme of your game.
- Initialise an object called `flagpole`, it should at least have the properties `x_pos` and `isReached`.
- set `isReached` to `false` and `x_pos` to a world position at the very end of your level.
- create a function called `drawFlagpole` and call this from the draw function
- complete the function to draw your flagpole in two states. One for when `isReached` is false,

and one for when it is `true`

3. Flagpole checking function [1 marks]

- create a function called `checkFlagpole`

- call the function from `draw`, but write a conditional so that `checkFlagpole` is only called when `flagpole.isReached` is `false`

- in `checkFlagpole` write a conditional such that when the gameChar is in range of the flagpole

its `isReached` property is set to `true`

4. Add lives [2 marks]

- Your character should begin with three lives, and each time they fall down a canyon the game

should reset and their remaining lives decrement by one.

- Create a global variable `lives`, and initialise it to `3` within `setup`.

- Create a function called `checkPlayerDie`. Call this within draw.

- In this function define a conditional statement that tests if your character has fallen below

the bottom of the canvas. When this is `true`, decrement the `lives` counter by one

- Create a new function called `startGame()`.

- Move everything from `setup` except `createCanvas` and the initialisation of `floorPos_y` and

`lives` into this new function.

- At the end of your now very short `setup` function call `startGame()`.

- In `checkPlayerDie` create a conditional statement to test if the player has

used all of their lives. If there are lives remaining call `startGame`.

- Write some code using a `for` loop to draw life tokens onto the screen so that you can keep track of how many lives you have remaining.

5. "Game over" and "Level complete" text [2 marks]

- In the draw loop, after your drawing code and before your game logic code, write two conditional statements

- The first displays "Game over" when `lives` is less than 1.

- The other displays "Level complete" when`flagpole.isReached` is true

- Write further conditionals in the draw and keyPressed functions to ensure that no further interaction is possible once these end of game states have been reached.

6. Tidy your code [3 marks]

- make sure your code is elegant

- remove all commented blocks of code

- check all indentations

- make your variable names are consistent

- remove any redundant code

- refactor unwieldy drawing code

- break up long commands onto multiple lines

s