**Review criteria**

**less**

1. Add a score counter [1 mark]

2. Add a flagpole [1 mark]

3. Flagpole checking function [1 mark]

4. Add lives [2 marks]

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

6. Tidy your code [3 marks]

**Step-By-Step Assignment Instructions**

**less**

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 mark]

* Create a global variable called `game\_score`.
* Increment `game\_score` by one each time the character collects an item.
* Use the text function to draw the score on the screen.

2. Add a flagpole [1 mark]

* We need to add an end to your level. I have chosen a flagpole but you can choose 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 `renderFlagpole` 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 mark]

* 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. Press space to continue." when `lives` is less than 1.
* The other displays "Level complete. Press space to continue." when `flagpole.isReached` is true.
* For each conditional, you should return at the end of the statement. This prevents any further game logic from happening when play is over.

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 consistent.
* Remove any redundant code.
* Refactor unwieldy drawing code.
* break up long commands onto multiple lines.

Once your done record a short video (2mins max) talking through your code and demonstrating your code working.