

The Game Project 5 - multiple interactables

It's time to level up our game by making multiple interactable items to populate our game world. Our code is getting quite complex now so first we'll take this opportunity to do some refactoring making it more navigable.

1. Refactor background items rendering [3 marks]

- At the bottom of your code, create three new functions called ``drawClouds``, ``drawMountains``, and ``drawTrees``
- Cut and paste each ``for loop`` from your ``draw`` function into the relevant new function.
- In the ``draw`` function, where the drawing code was previously, call each of the three new functions (eg. ``drawClouds()``)
- You should now see all of the above items drawn to the screen as before

2. Refactor collectable rendering [3 marks]

- We're going to do something slightly different in this case, we're going to pass the collectable as an argument to our render function
- At the bottom of your code, create a function called ``drawCollectable``
- It should take a single argument called ``t_collectable`` like so ``function drawCollectable(t_collectable)``
- Cut and paste the drawing code for the collectable from the ``draw`` function and into your ``drawCollectable`` function
- Now replace any reference to `collectable` with `t_collectable`
- Finally in the ``draw`` function, where the drawing code was previously, call ``drawCollectable`` passing `collectable` as an argument
- You should see your collectable appearing in the same place as previously

3. Refactor canyon rendering [3 marks]

- Repeat the same process for canyon as in step 2
- You should end up with a function called ``drawCanyon`` which is called from the ``draw`` function passing ``canyon`` as an argument

4. Refactor collectable interaction [3 marks]

- Your code which sets the ``isFound`` property of collectable to true should still be in the draw loop. Lets refactor that too
- Create a function called ``checkCollectable`` which takes an argument called ``t_collectable``
- Cut and paste your collectable interaction code from the ``draw`` function into this new function replacing all references to ``collectable`` with ``t_collectable``
- Finally in the ``draw`` function, where the interaction code was previously, call ``checkCollectable`` passing collectable as an argument
- You should find that your collectable interaction works just as before

5. Refactor canyon interaction [3 marks]

- Repeat stage 4 but for the canyon
- You should end up with a function called ``checkCanyon`` which is called from the ``draw`` function passing ``canyon`` as an argument
- Make sure your canyon interaction works for all of your canyons

6. Create multiple collectables [3 marks]

- Create a global variable called ``collectables`` and initialise it to an array
- Fill the array with lots of collectable objects. You can copy and paste from your original collectable object but be sure to alter the `x_pos` property so that the collectables appear in different places in your game world.
- In the ``draw`` function, wrap the calls to ``drawCollectable`` and ``checkCollectable`` in a for loop which traverses collectables

- Instead of passing ``collectable`` to your functions pass a reference to the particular collectable in your ``collectables`` array (eg. ``drawCollectable(collectables[i])``).

- Once you've done this delete the declaration and initialisation of your old single ``collectable`` . It's now redundant code.

- If every thing is working correctly you should see multiple collectables and be able to interact with them independently.

7. Create multiple canyons [3 marks]

- Repeat stage 6 but for canyons.

- Once you're satisfied that the interaction and rendering is working try varying the width of the canyons

Things to watch out for

- For collectables and canyons, make sure you are using the passed in arguments in your refactored code . You should be using ``t_canyon`` , NOT ``canyon`` or ``canyons[i]``

- Delete your unused variables. You will lose marks for leaving redundant code in.