RiceRocks - Code Clinic tips

Help Center

This page include helpful tips based on our experience in helping students in the "Code Clinic" (interactivepython@online.rice.edu). Be sure and take a look at these tips if you get stuck.

1. Spawning rocks

   To handle multiple rocks, replace `a_rock` by `rock_group`. Inside `rock_spawner`, you should spawn a local copy of `a_rock` and add it to the global set `rock_group.` To avoid going over 12 rocks, you can use a test based on the length of `rock_group`. To test your code, you can add a `for` loop to the draw handler of the form: for each rock in `rock_group`, draw that rock. This draw will then be pulled into `process_sprite_group` in the last step of phase 1.

2. Computing collisions for sprites:

   When computing collisions for sprites, you'll use getters like `get_position()` and `get_radius()` to pull fields out of the sprite objects. Remember to use () at the end of these methods, a common error is try to call them without ending in (), e.g; `a_sprite.get_position` as opposed `a_sprite.get_position()`

   Also, remember to choose names for the class fields that are distinct from the names of the class methods. Another common error is to have a field radius and then try to define a `radius()` getter. This will throw an error in Python.

3. Restarting the game

   Several of you have sent me code that almost works except that your program seems to ignore mouse clicks when you try to restart the game after the previous game ends. In this case, you need to remember to reset the `lives = 3` and `score = 0` (as globals) inside the mouse handler.

4. Keeping track of lives/score

   Many of you in writing `group_collide` and `group_group_collide` try to keep track of collisions in one (or two) global variables. Just use local variables to keep track of the collisions inside each call and return the value of those variables. For example, `group_collide` can return the either `True` or `False` depending on whether the ship hit a rock. `group_group_collide` can return the number of rocks that the missiles struck. These values can be used to update lives and score in the draw handler.

5. Testing for correct collisions

   To help you in determining whether your `collide` method for the Sprite class and the `group_collide` function work correctly, we have created the testing template below:

   http://www.codeskulptor.org/#examples-collide_template.py

   As with previous testing templates, paste your implementation of these two items into the template and compare the output produced by your implementation with that produced by our implementation.