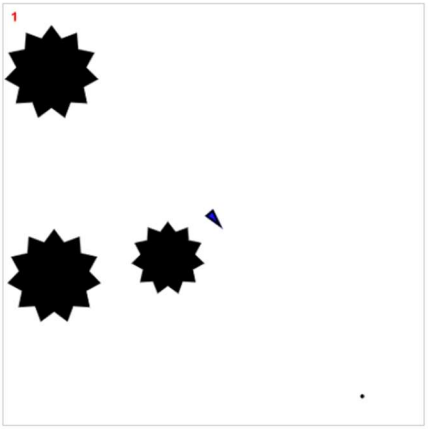
Final Python Project in CMU CS

You are building an asteroids game! Look at the provided example on CMU CS for insights how it works.



1. Ship can rotate (using RIGHT & LEFT arrow keys)
    a. Pay attention to ship.angle for this
2. Ship can move at all (using UP & DOWN arrow keys)
    a. Consider using “onKeyHold” function for these keys
3. Ship will maintain movement speed on it’s own
    a. Pay attention to ship.speed for this
4. Ship loops around screen
5. Shots can be fired (using SPACE key)
    a. Consider the “onKeyPress” function
6. Shots are limited to one shot on the screen at a time
7. Asteroid(s) move
8. Asteroids move independently (not all together in the same fashion)
9. Asteroids loop around screen
10. Asteroids shrink when shot
11. Asteroids disappear when hit 4 times
12. Asteroids can spawn (re-appear as a ‘new asteroid’)
13. Score is kept in corner
14. Game Over appears when ship hits an asteroid, also prints the score and stops motion
15. Visuals are at least as good as example
16. Extra visuals
    (better looking asteroids, explosions when asteroids are hit, “thrust” from back of the ship when changing speed, fancy background while maintaining contrast, etc)

Level 1: Develop at least 5 of these features

Level 2: Develop at least 8 of these features

Level 3: Develop at least 11 of these features

Level 4: Develop all of these features (meaning, at least one extra visual)

Tips and Tricks

Rotating shapes

Simply set the "rotateAngle" to whatever angle you want and the shape will rotate to match. In this environment, an angle of 0 means "UP", and an angle of 90 means "RIGHT"

onKeyPressed vs. onKeyHold

```
def onKeyPressed(key):

  #this function is fired for each key pressed
  #use it with an if statement like this
  if key == "space":
      #do something
```

```
def onKeyHold(keys):
  #this function is fired over and over and
  #the 'keys' variable is a list of all keys held
  #use it a bit differently, like this
  if "space" in keys:
      #do something
```

Moving in a direction based on an angle

If you are trying to figure out how to move a ship or asteroid based on an angle, you can do this:

```
newX,newY = getPointInDir(ship.centerX, ship.centerY, ship.angle, ship.speed)
ship.centerX = newX
ship.centerY = newY
```

Adding / Removing from a group

You can add shapes with "groupName.add(newShape)"

You can remove shapes with "groupName.remove(shapeToRemove)"

Cycling through items in a group

When you have a group of shapes, you can cycle through the entire group with something like this:

(assuming you have a group of star Shaped called 'asteroids')

```
for a in asteroids:
       a.centerX += 5
       a.centerY += 5
```

Checking for collisions

You can cycle through all your asteroids and see if they have collided with your ship like this:

```
for a in asteroids:
     if a.hitsShape(ship):
          #do something
```

Adding on to shape properties

Say you wanted a rectangle, but wanted to keep track of some extra information like angle and speed…. Just make it with a variable first, add the properties, THEN add it to the group…

```
newSquare = Rect(...)
newSquare.angle = ...
newSquare.speed = ...
asteroids.add(newSquare)
```