```
from sys import exit
from random import randint



class Scene(object):

    def enter(self):
        print ("This scene is not yet configured. Subclass it and implement enter().")
        exit(1)

class Engine(object):

    def __init__(self, scene_map):
        self.scene_map = scene_map

    def play(self):
        current_scene = self.scene_map.opening_scene()
        last_scene = self.scene_map.next_scene('finished')

        while current_scene != last_scene:
            next_scene_name = current_scene.enter()
            current_scene = self.scene_map.next_scene(next_scene_name)

        # be sure to print out the last Scene
        current_scene.enter()

class Death(Scene):

    quips = [
        "You died. You kinda suck at this.",
        "Your mom would be proud...if she were smarter.",
        "Such a loser.",
        "I have a small pappy that is better at this."
    ]

    def enter(self):
        print (Death.quips[randint(0, len(self.quips)-1)])
        exit(1)

class CentralCorridor(Scene):

    def enter(self):
        print ("The Gothons of Planet Percal #25 have invaded your ship and destoryed,")
        print ("your entire crew. You are the last surviving member and your last,")
        print ("mission is to get the neutron destruct bomb from the Weapons Armory,")
        print ("put it in the bridge, and blow the ship up after getting into an")
        print ("escape pod.")
        print ("\n")
```

```
        print ("You are running down the central corridor to the Weapons Armory when")
        print ("a Gothon jumps out, red shin, dark teeth, and evil clown costume")
        print ("flowing around his hit filled body. He's blocking the door to the")
        print ("Armory and about to pull a Weapon to blast you.")

        action = input("> ")

        if action == "shoot!":
            print ("Quick on the draw you yank out blasetr and fire it at the Gothon.")
            print ("His clown costume is flowing and moving around his body, which throws")
            print ("off your aim. Your laser hits his coetume but misses his entirely. This")
            print ("completely ruins his brand new costume his mother bought him, which")
            print ("makes him fly into an insane rage and blast you repeatedly in the face")
            print ("you are dead. Then he eats you.")
            return ('death')

        elif action == "dodge!":
            print ("Like a world class boxer you dodge, weave, slip and slide right")
            print ("as the Gothon's blaster cranks a laser past your head.")
            print ("In the middle of your artful dodge your foot slips and you")
            print ("bang your head on the metal wall and pass out.")
            print ("You wake up shortly after only to die as the Gothon stomps on")
            print ("your head and eats you.")
            return ('death')

        elif action == "tell a joke":
            print ("Lucky for you they made you learn Gothon insults in the academy.")
            print ("You tell the one Gothon joke you know:")
            print ("hinv civnjvn jjkjewfj odpofjhv  dewfhfh fjjjf.,fjfjifj,jodjejf,ooqwoq.")
            print ("The Gothon stops, tries not to laugh, then busts out laughing and cannot move.")
            print ("While he is laughing you run up and shoot square in the head")
            print ("Putting him down, then jump through the Weapon Armory door.")
            return ('laster_weapon_armony')

        else:
            print ("Does not compute!")
            return ('central_corridor')
class LaserWeaponArmory(Scene):

    def enter(self):
        print ("You do a dive roll into the Weapon Armony, crough and scan the room")
        print ("for more Gothons that might be hiding, It's dead quiet, too quiet.")
        print ("You shand up and run to the far side of the room and find the")
        print ("neutron bomb in its container. There is a keypad lock on the box")
        print ("and you need the code to get the bomb out. If you get the code")
        print ("wrong 10 times then the lock closes forever and you can't")
        print ("can't get the bomb. The code is 3 digits.")
        code = "%d%d%d" % (randint(1,9), randint(1,9), randint(1,9))
        guess = input("[keypad]> ")
```

```
        guesses = 0

        while guess != code and guesses < 10:
            print ("BZZZZZZZZEDDDDD")
            guesses += 1
            guess = input("[keypad]> ")
        if guess == code :
            print ("The container clicks open and the seal breaks, letting gas out.")
            print ("You grab the neutron bomb and run as fast as you can to the")
            print ("bridge where you must place it in the right spot.")
            return ('the_bridge')
        else:
            print ("The lock buzzes one last time and then you hear a sickening")
            print ("melting sound as the mechanism is fused together.")
            print ("You decide to sit there, and finnally the Gothons blow up the")
            print ("ship from their ship and you die.")
            return ('death')

class TheBridge(Scene):

    def enter(self):
        print ("You burst onto the bridge with the neutron destruct bomb")
        print ("under your arm and surprise 5 Gothons who are trying to")
        print ("take control of the ship. Each of them has a even uglier")
        print ("clown costume than the last. They haven't pulled their")
        print ("weapons out yet, as they see the active bomb under your")
        print ("arm and donnot want to set it off.")

        action = input("> ")

        if action == "throw the bomb":
            print ("In a panic you throw the bomb at the groups of Gothons")
            print ("and make a leap for the door. Right as you drop it a")
            print ("Gothen shoot you right in the back killing you")
            print ("As you die you see another Gothon frantically try to disarm")
            print ("the bomb. You die knowing they will probably blow up when")
            print ("it goes off.")
            return ('death')

        elif action == "slowly place the bomb":
            print ("You point your blaster at the bomb under you arm")
            print ("and the Gothons put their  hands up and start to sweat.")
            print ("You inch backward to the door, open it, and then carefully")
            print ("place the bomb on the floor, pointing your blaster at it.")
            print ("You then jump back through the door, punch the close butten")
            print ("and blast the lock so the Gothens can't get out.")
            print ("Now that the bomb is placed you run to the escape pod to")
            print ("get off this tin can.")
            return ('escape_pod')
```

```
        else:
            print ("DOES NOT COMPUTE")
            return ("the_bridge")

class EscapePod(Scene):

    def enter(self):
        print ("You rush through the ship desperately trying to make it to")
        print ("the escape pod before the whole ship explodes. It seems like")
        print ("hardly any Gothons are on the ship, so your run is clean of")
        print ("interference. You get to the chamter with the escape pods, and")
        print ("now need to pick one to take. Some of them could ne demaged")
        print ("but you donot have time to look. There's 5 pods, which one")
        print ("do you take?")

        good_pod = randint(1,5)
        guess = input("[pod #]> ")

        if int(guess) != good_pod:
            print ("You jump into pod %s and hit the eject button." % guess)
            print ("The pod escapes out into the void of space, then")
            print ("implodes as the hull ruptures, crushing your body")
            print ("into jam jelly.")
            return ('death')
        else:
            print ("You jump into pod %s and hit the eject button." % guess)
            print ("The pod easily slides out into space heading to")
            print ("the planet below. As it flies to the planet, you look")
            print ("back and see your ship implode then explode like a")
            print ("bright star, taking out the Gothon ship at the same")
            print ("time. You won!")

            return ('finished')

class Finished(Scene):

    def enter(self):
        print ("You won! Good job.")
        return ('finished')


class Map(object):

    scenes = {
        'central_corridor': CentralCorridor(),
        'laster_weapon_armony': LaserWeaponArmory(),
        'the_bridge': TheBridge(),
        'escape_pod': EscapePod(),
```

```
        'death': Death(),
        'finished': Finished(),
    }

    def __init__(self, start_scene):
        self.start_scene = start_scene

    def next_scene(self, scene_name):
        val = Map.scenes.get(scene_name)
        return val

    def opening_scene(self):
        return self.next_scene(self.start_scene)

a_map = Map('central_corridor')
a_game = Engine(a_map)
a_game.play()
```