```
# Created by: Mr. Coxall
# Created on: Sep 2016
# Created for: ICS3U
# This scene shows the main game.

from scene import *
import ui
from numpy import random
#from copy import deepcopy


class GameScene(Scene):
    def setup(self):
        # this method is called, when user moves to this scene

        # updated to not use deepcopy
        self.size_of_screen_x = self.size.x
        self.size_of_screen_y = self.size.y
        self.screen_center_x = self.size_of_screen_x/2
        self.screen_center_y = self.size_of_screen_y/2

        self.score_position = Vector2()
        self.left_button_down = False
        self.right_button_down = False
        self.ship_move_speed = 20.0
        self.missiles = []
        self.aliens = []
        self.alien_attack_rate = 1
        self.alien_attack_speed = 20.0
        self.scale_size = 0.75
        self.score = 0
        self.dead = False

        # add background color
        background_position = Vector2(self.screen_center_x,
                                      self.screen_center_y)
        self.background = SpriteNode('./assets/sprites/
        star_background.png',
                                     position =
                              background_position,
                                     parent = self,
                                     size = self.size)

        spaceship_position = Vector2()
        spaceship_position.x = self.screen_center_x
        spaceship_position.y = 100
        self.spaceship = SpriteNode('./assets/sprites/
```

```
        spaceship.png',
46                                  parent = self,
47                                  position = spaceship_position,
48                                  scale = self.scale_size)
49
50          left_button_position = Vector2()
51          left_button_position.x = 100
52          left_button_position.y = 100
53          self.left_button = SpriteNode('./assets/sprites/
            left_button.png',
54                                    parent = self,
55                                    position =
                              left_button_position,
56                                    alpha = 0.5,
57                                    scale = self.scale_size)
58
59          right_button_position = Vector2()
60          right_button_position.x = 300
61          right_button_position.y = 100
62          self.right_button = SpriteNode('./assets/sprites/
            right_button.png',
63                                     parent = self,
64                                     position =
                              right_button_position,
65                                     alpha = 0.5,
66                                     scale = self.scale_size)
67
68          fire_button_position = Vector2()
69          fire_button_position.x = self.size_of_screen_x - 100
70          fire_button_position.y = 100
71          self.add_alien()
72          self.fire_button = SpriteNode('./assets/sprites/
            red_button.png',
73                                    parent = self,
74                                    position =
                              fire_button_position,
75                                    alpha = 0.5,
76                                    scale = self.scale_size)
77
78          self.score_position.x = 100
79          self.score_position.y = self.size_of_screen_y - 50
80          self.score_label = LabelNode(text = 'Score: 0',
81                                       font=('Helvetica', 40),
82                                       parent = self,
83                                       position =
                              self.score_position)
84
```

```
85
86      def update(self):
87          # this method is called, hopefully, 60 times a second
88
89          # move spaceship if button down
90          if self.left_button_down == True:
91              spaceshipMove = Action.move_by(-1*self.ship_move_speed,
92                                             0.0,
93                                             0.1)
94              self.spaceship.run_action(spaceshipMove)
95
96          if self.right_button_down == True:
97              spaceshipMove = Action.move_by(self.ship_move_speed,
98                                             0.0,
99                                             0.1)
100             self.spaceship.run_action(spaceshipMove)
101
102         # every update, randomly check if a new alien should be
            created
103         alien_create_chance = random.randint(1, 120)
104         if alien_create_chance <= self.alien_attack_rate:
105             # only create new aliens if you are not dead
106             if self.dead == False:
107                 self.add_alien()
108
109         # check every update if a missile is off screen
110         for missile in self.missiles:
111             if missile.position.y > self.size_of_screen_y + 50:
112                 missile.remove_from_parent()
113                 self.missiles.remove(missile)
114                 #print('missile removed')
115
116         # check every update if an alien is off screen
117         #print(len(self.aliens))
118         for alien in self.aliens:
119             if alien.position.y < -50:
120                 alien.remove_from_parent()
121                 self.aliens.remove(alien)
122                 # you might want to end the game, or take points
                    away
123                 self.score = self.score - 2
124
125         # check every update to see if a missile has touched a
            space alien
126         if len(self.aliens) > 0 and len(self.missiles) > 0:
127             #print('missile check')
128             for alien in self.aliens:
```

```
129                 for missile in self.missiles:
130                     if alien.frame.contains_rect(missile.frame):
131                         missile.remove_from_parent()
132                         self.missiles.remove(missile)
133                         alien.remove_from_parent()
134                         self.aliens.remove(alien)
135                         self.score = self.score + 1
136                         # since you destroyed one, make more show
                            up
137                         #self.alien_attack_rate =
                            self.alien_attack_rate + 1
138         else:
139             pass
140             #print(len(self.aliens))
141
142         # check every update to see alien touches spaceship
143         if len(self.aliens) > 0:
144             #print('checking')
145             for alien_hit in self.aliens:
146                 #print('alien ->' + str(alien_hit.frame))
147                 #print('ship  ->' + str(self.spaceship.frame))
148                 if
                    alien_hit.frame.intersects(self.spaceship.frame):
149                     #print('a hit')
150                     self.spaceship.remove_from_parent()
151                     alien_hit.remove_from_parent()
152                     self.aliens.remove(alien_hit)
153                     # since game over, move to next scene
154                     self.dead = True
155                     self.menu_button = SpriteNode('./assets/
                        sprites/menu_button.png',
156                                                   parent = self,
157                                                   position =
                                Vector2(self.screen_center_x,
158
                                self.screen_center_y),
159                                                   alpha = 1.0,
160                                                   scale = self.scale_size)
161         else:
162             pass
163             #print(len(self.aliens))
164
165         # update every frame the current score
166         self.score_label.text = 'Score: ' + str(self.score)
167
168     def touch_began(self, touch):
169         # this method is called, when user touches the screen
```

```
170
171         # check if left or right button is down
172         if self.left_button.frame.contains_point(touch.location):
173             self.left_button_down = True
174
175         if self.right_button.frame.contains_point(touch.location):
176             self.right_button_down = True
177
178     def touch_moved(self, touch):
179         # this method is called, when user moves a finger around on
            the screen
180         pass
181
182     def touch_ended(self, touch):
183         # this method is called, when user releases a finger from
            the screen
184
185         # if start button is pressed, goto game scene
186         if self.fire_button.frame.contains_point(touch.location):
187             # only shoot if you are not dead
188             if self.dead == False:
189                 self.create_new_missile()
190         else:
191             # if I removed my finger, then no matter what spaceship
192             #    should not be moving any more
193             self.left_button_down = False
194             self.right_button_down = False
195
196         # if game over, check to go back to main menu
197         if self.dead == True:
198             # if start button is pressed, goto game scene
199             if
                self.menu_button.frame.contains_point(touch.location):
200                 self.dismiss_modal_scene()
201
202     def did_change_size(self):
203         # this method is called, when user changes the orientation
            of the screen
204         # thus changing the size of each dimension
205         pass
206
207     def pause(self):
208         # this method is called, when user touches the home button
209         # save anything before app is put to background
210         pass
211
212     def resume(self):
```

```
213         # this method is called, when user place app from
            background
214         # back into use. Reload anything you might need.
215         pass
216
217     def create_new_missile(self):
218         # when the user hits the fire button
219
220         missile_start_position = Vector2()
221         missile_start_position.x = self.spaceship.position.x
222         missile_start_position.y = 100
223
224         missile_end_position = Vector2()
225         missile_end_position.x = missile_start_position.x
226         missile_end_position.y = self.size_of_screen_y + 100
227
228         self.missiles.append(SpriteNode('./assets/sprites/
            missile.png',
229                              position = missile_start_position,
230                              parent = self))
231
232         # make missile move forward
233         missileMoveAction = Action.move_to(missile_end_position.x,
234                                            missile_end_position.y +
                                 100,
235                                            5.0)
236
            self.missiles[len(self.missiles)-1].run_action(missileMoveA
            ction)
237
238     def add_alien(self):
239         # add a new alien to come down
240
241         alien_start_position = Vector2()
242         alien_start_position.x = random.randint(100,
243                                                 self.size_of_screen_x -
                                 100)
244         alien_start_position.y = self.size_of_screen_y + 100
245
246         alien_end_position = Vector2()
247         alien_end_position.x = random.randint(100,
248                                               self.size_of_screen_x -
                                 100)
249         alien_end_position.y = -100
250
251         self.aliens.append(SpriteNode('./assets/sprites/alien.png',
252                              position = alien_start_position,
```

```
253                                 parent = self))
254
255         # make missile move forward
256         alienMoveAction = Action.move_to(alien_end_position.x,
257                                          alien_end_position.y,
258                                          self.alien_attack_speed,
259                                          TIMING_SINODIAL)
260         self.aliens[len(self.aliens)-1].run_action(alienMoveAction)
261
```