*# from pygame.examples import aliens  
# aliens.main()***import** sys  
**import** pygame  
  
**from** settings **import** Settings  
**from** ship **import** Ship  
  
**class** AlienInvasion:  
 *"""Overall class to manage game assets and behavior."""* **def** \_\_init\_\_(self):  
 *"""Initialize the game, and create game resources."""* pygame.init()  
 self.settings = Settings()  
  
 self.screen = pygame.display.set\_mode((0, 0), pygame.FULLSCREEN)  
 self.settings.screen\_width = self.screen.get\_rect().width  
 self.settings.screen\_height = self.screen.get\_rect().height  
  
 pygame.display.set\_caption(**"Alien Invasion"**)  
 self.ship = Ship(self)  
 **def** run\_game(self):  
 *"""Start the main loop for the game."""* **while True**:  
 self.check\_events()  
 self.ship.update()  
 self.\_update\_screen()  
 *# #Redraw the screen during each pass through the loop.  
 # Watch for keyboard and mouse events.  
 # Redraw the screen during each pass through the loop.  
 # self.screen.fill(self.settings.bg\_color)  
 # Make the most recently drawn screen visible.* self.screen.fill(self.settings.bg\_color)  
 **def** check\_events(self):  
 *"""Respond to keypresses and mouse events."""* **for** event **in** pygame.event.get():  
 **if** event.type == pygame.QUIT:  
 sys.exit()  
 **elif** event.type == pygame.KEYDOWN:  
 self.\_check\_keydown\_events(event)  
 *#  
 # if event.key == pygame.K\_RIGHT:  
 # # Move the ship to the right.  
 # self.ship.moving\_right = True  
 # elif event.key == pygame.K\_LEFT:  
 # self.ship.moving\_left = True* **elif** event.type == pygame.KEYUP:  
 self.\_check\_keyup\_events(event)  
 **def** \_check\_keydown\_events(self, event):  
 *"""Respond to keypresses."""* **if** event.key == pygame.K\_RIGHT:  
 self.ship.moving\_right = **True  
 elif** event.key == pygame.K\_LEFT:  
 self.ship.moving\_left = **True  
 elif** event.key == pygame.K\_q:  
 sys.exit()  
 **def** \_check\_keyup\_events(self, event):  
 *"""Respond to keyp releases."""* **if** event.key == pygame.K\_RIGHT:  
 self.ship.moving\_right = **False  
 elif** event.key == pygame.K\_LEFT:  
 self.ship.moving\_left = **False  
  
 def** \_update\_screen(self):  
 *"""Update images on the screen, and flip to the new screen."""* self.screen.fill(self.settings.bg\_color)  
 self.ship.blitme()  
 pygame.display.flip()  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 *#Make a game instance, and run the game.* ai = AlienInvasion()  
 ai.run\_game()

**class** Settings:  
 *"""A class to store all settings for ALien Invasion."""* **def** \_\_init\_\_(self):  
 *"""Initialize the game's settings."""  
 # Screen settings* self.screen\_width = 1200  
 self.screen\_height = 700  
 self.bg\_color = (230, 230, 230)  
 self.ship\_speed = 1.5

**import** pygame  
  
**class** Ship:  
 *# A class to manage the ship.* **def** \_\_init\_\_(self, ai\_game):  
 *"""Initialize the ship and set its starting position."""* self.screen = ai\_game.screen  
 self.settings = ai\_game.settings  
 self.screen\_rect = ai\_game.screen.get\_rect()  
 *# Load the ship image and get its rect.* self.image = pygame.image.load(**'images/ship.bmp'**)  
 self.rect = self.image.get\_rect()  
 *# Start each new ship at the bottom of the screen.* self.rect.midbottom = self.screen\_rect.midbottom  
 *# Store a decimal value for the ship's horizontal position.* self.x = float(self.rect.x)  
 *# Movement flags* self.moving\_right = **False** self.moving\_left = **False  
 def** update(self):  
 *"""Update the ship's position based on the movement flag."""  
 # Update the ship's x value, not the rect.* **if** self.moving\_right **and** self.rect.right < self.screen\_rect.right:  
 self.x += self.settings.ship\_speed  
 **if** self.moving\_left **and** self.rect.left > 0:  
 self.x -= self.settings.ship\_speed  
 self.rect.x = self.x  
 **def** blitme(self):  
 *"""Draw the ship at its current location."""* self.screen.blit(self.image, self.rect)