**import arcade  
  
SCREEN\_WIDTH = 800  
SCREEN\_HEIGHT = 600  
  
  
class MyBlock(arcade.Window):  
    """ Main application class. """  
  
    def \_\_init\_\_(self, width, height):  
        super().\_\_init\_\_(width, height)  
  
        arcade.set\_background\_color(arcade.color.AMAZON)  
  
    def setup(self):  
          
        pass  
  
    def on\_draw(self):  
          
        arcade.start\_render()  
          
  
    def update(self, delta\_time):  
          
        pass  
  
  
def main():  
    game = MyBlock(SCREEN\_WIDTH, SCREEN\_HEIGHT)  
    game.setup()  
    arcade.run()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
    main()**

**#Sprite and Sprite lists**

**def setup(self):  
   
    *# Create the sprite lists*  
    self.player\_list = arcade.SpriteList()  
    self.block\_list = arcade.SpriteList()  
  
    *# Score*  
    self.score = 0  
  
      
    self.player\_sprite = arcade.Sprite("images/character.png", SPRITE\_SCALING\_PLAYER)  
    self.player\_sprite.center\_x = 50 *# Starting position*  
    self.player\_sprite.center\_y = 50  
    self.player\_list.append(self.player\_sprite)  
  
    *# Create the block*  
    for i in range(BLOCK\_COUNT)  
          
        *# Position the block*  
        block.center\_x = random.randrange(SCREEN\_WIDTH)  
        block.center\_y = random.randrange(SCREEN\_HEIGHT)  
  
        *# Add the block to the lists*  
        self.block\_list.append(block)**

#We can easily draw all the block in the block lists:

**def on\_draw(self):**

**arcade.start\_render()  
    self.block\_list.draw()  
    self.player\_list.draw()**

**#Detecting sprite collisions**

**def update(self, delta\_time):  
        block\_hit\_list = arcade.check\_for\_collision\_with\_list(self.player\_sprite, self.block\_list)  
  
      
    for block in blocks\_hit\_list:  
        block.kill()  
        self.score += 1**

**MOVEMENT\_SPEED = 5  
  
def on\_key\_press(self, key, modifiers):  
      
  
    if key == arcade.key.UP:  
        self.player\_sprite.change\_y = MOVEMENT\_SPEED  
    elif key == arcade.key.DOWN:  
        self.player\_sprite.change\_y = -MOVEMENT\_SPEED  
    elif key == arcade.key.LEFT:  
        self.player\_sprite.change\_x = -MOVEMENT\_SPEED  
    elif key == arcade.key.RIGHT:  
        self.player\_sprite.change\_x = MOVEMENT\_SPEED  
  
def on\_key\_release(self, key, modifiers):  
  
  
    if key == arcade.key.UP or key == arcade.key.DOWN:  
        self.player\_sprite.change\_y = 0  
    elseif key == arcade.key.LEFT or key == arcade.key.RIGHT:  
        self.player\_sprite.change\_x = 0**