Activity 1 - Space Invaders Program

Create a Space Invaders game that follows the below output.

| Welcome to Space Invaders!  Current enemies:  0 - Type: Squid - Health: 10 : COLOR - White  1 - Type: Squid lvl 2 - Health: 20 : COLOR - Black  2 - Type: Octopus - Health: 20: SHIELD 40  a)ttack, e)nemies, q)uit  >>> r  Invalid option  a)ttack, e)nemies, q)uit  >>> a  Choose alien (number): 0  You damaged the Squid for 14 damage!  Current HP -4  The Squid has been defeated!  a)ttack, e)nemies, q)uit  >>> e  0 - Type: Squid lvl 2 - Health: 20 : COLOR - Black  1 - Type: Octopus - Health: 20: SHIELD 40  a)ttack, e)nemies, q)uit  >>> a  Choose alien (number): 1  You knocked down the Octopus's shield to 28  a)ttack, e)nemies, q)uit  >>> a  Choose alien (number): 1  You knocked down the Octopus's shield to -15  a)ttack, e)nemies, q)uit  >>> e  0 - Type: Squid lvl 2 - Health: 20 : COLOR - Black  1 - Type: Octopus - Health: 20  a)ttack, e)nemies, q)uit  >>> a  Choose alien (number): 1  You damaged the Octopus for 29 damage! Current HP -9  The Octopus has been defeated!  a)ttack, e)nemies, q)uit  >>> e  0 - Type: Squid lvl 2 - Health: 20 : COLOR - Black  a)ttack, e)nemies, q)uit  >>> a  Choose alien (number): 0  You damaged the Squid lvl 2 for 39 damage!  Current HP -19  The Squid lvl 2 has been defeated!  a)ttack, e)nemies, q)uit  >>> e  All enemies have been defeated!  a)ttack, e)nemies, q)uit  >>> q  Thanks for playing Space Invaders. |
| --- |

The program will consist of 4 different files:

* alien.py
* invader.py
* boss.py
* space\_invaders.py

The alien.py file is the base class that invader.py and boss.py will inherit from. Alien.py has 3 methods:

* Initialiser - Initialize an alien instance with a “type” and “health” arguments.
* String representation - Return a string representation.
* Attack - Attack the alien with a random amount of damage to reduce its health.

The invader class has 2 methods and will inherit from the alien class.

* Initialiser - Initialize an invader instance and add the “color” argument.
* String representation - Overwrite the base class’s string representation to include a color. Be careful of DRY here!

The boss class has 3 methods and will inherit from the alien class.

* Initialiser - Initialize a boss instance and add the “shield” argument.
* String representation - Overwrite the base class’s string representation to include a shield. Be careful of DRY here!
* Attack - Overwrite the base class’s attack method to damage the shield first and then the health. Be careful of DRY here!

The space\_invaders.py is the main file that will run the game. It will have 3 functions and the main menu has 3 options:

* main
  + Attack - choose an enemy to attack with a random amount of damage
  + Enemies - display all the enemies
  + Quit - Quit the program
* Print\_aliens - This will print all the aliens that are stored in the alien list
* Choose\_alien (optional) - Error checking for selecting an alien from the list to attack

Start coding on the next page

alien.py

| """An alien class - PASS"""   class Alien:  """Represents an alien invader."""   def \_\_init\_\_(self, alien\_type=’’, health=0):  """Initialize an alien instance. An alien has a alien type and health."""  self.alien\_type = alien\_type  Self.health = health   def \_\_str\_\_(self):  """Return an alien in the form of: 'Type: {alien\_type} - Health: {alien\_health}."""  return f”Type: {self.alien\_type} - Health: {alien\_health}.”   def attack(self, damage):  ""Attack the alien to reduce its health.  If its health is greater than 0, output:  'You damaged the {alien\_type} for {damage} damage!  Current HP {health}'  If the alien runs out of health, output:  'The {alien\_type} has been defeated!}’"""  if self.health > 0:  self.health = self.health - damage  print(f”You damaged the {self.alien\_type} for {damage} damage! Current HP {self.health}”)  if self.health <= 0:  print(f”The {self.alien\_type} has been defeated!”) |
| --- |
|  |

invader.py

| """Invader class - PASS""" from alien import Alien   class Invader(Alien):  """Specialized version of Alien."""   def \_\_init\_\_(self, color=’’, \*\*kwargs):  """Initialize an Invader type alien. It has an extra variable called color."""  super().init(\*\*kwargs)  Self.color = color   def \_\_str\_\_(self):  """Return a string representation:  'Type: {self.alien\_type} - Health: {self.health} : COLOR - {self.color}'  This looks similar to the base class string... maybe we can use that? Remember no DRY here!"""  return f”{super().\_\_str\_\_()} : COLOR - {self.color}” |
| --- |

boss.py

| """Boss class - PASS""" from alien import Alien   class Boss(Alien):  """Specialized version of alien - Boss."""   def \_\_init\_\_(self):  """Initialize a Boss type alien. It has an extra variable called shield."""  #start coding here   def \_\_str\_\_(self):  """Return a string representation if it has a shield:  'Type: {self.alien\_type} - Health: {self.health}. SHIELD {self.shield}'  Or return the base class string if there is no shield."""  #start coding here   def attack(self, damage):  """Attack the boss and knock down its health once it has no shield. Remember, no DRY here! The base class already handles the health..."""  #start coding here |
| --- |

space\_invaders.py

| """Space invaders game - PASS""" import random  from boss import Boss from invader import Invader  MENU = "a)ttack, e)nemies, q)uit"   def main():  """The Space Invader program consists of a menu with 3 options.  Attack - choose an enemy to attack  Enemies - display all the enemies  Quit - Quit the program  HINT: Make a list of aliens to store the invader and boss objects  HINT: Remember to remove the alien from the list once it's defeated!  """  aliens = [Invader(alien\_type=”Squid”, health=”100”, color=”yellow”), Invader(alien\_type=”Octopus”, health=”50”, color=”red”, Invader(alien\_type=”Crab”, health=”25”, color=”purple”))  print(“Current enemies:”  print\_aliens(aliens)  choice = input(">>> ").lower()  while choice != "q":  if choice == "a":  Alien\_index = int(input(“Choose Alien (number): “))  damage = random.randint(0, 100)  aliens[alien\_index].attack(damage)  If aliens[alien\_index].health <= 0:  aliens.remove(aliens[alien\_index]    elif choice == "e":  print\_aliens(aliens)  else:  print(“Invalid option”)  print(MENU)  choice = input(">>> ")  print("Thanks for playing Space Invaders.")   def print\_aliens(aliens):  """Print all the aliens in the list on a new line, each with their list index in the form of:  {list\_index} - {alien\_type}  If all enemies have been defeated, output: 'All enemies have been defeated'"""  For i, alien in enumerate(aliens):  print(f”{i} - {alien.alien\_type}”)   def choose\_alien(aliens): # OPTIONAL  """Choose an alien from the list of aliens with error checking.  If the user choice is greater than the amount of aliens in the list output: 'Invalid (too high)'  If the user choice is less than 0, output: 'Invalid (must be greater than 0)'  If the user choice is not an integer, output: 'Invalid (not an integer)'"""  #start coding here   main() |
| --- |
|  |