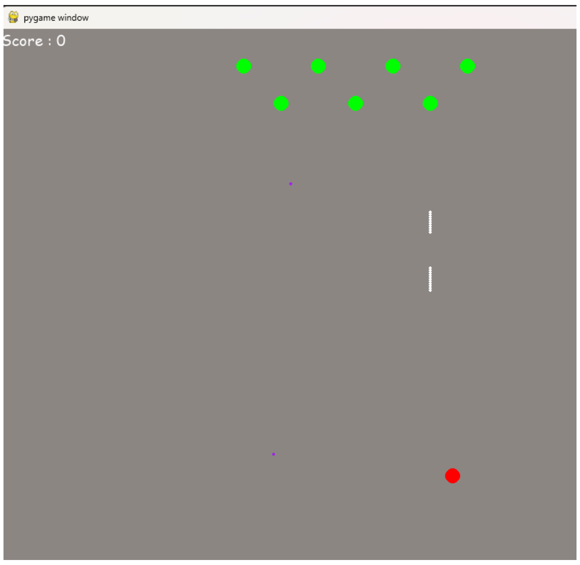
Space shooter game ING202



In this exercise, we will make a very simple arcade space shooter game. The aim in this exercise is to get a bit more used to loops, lists and conditionals.

I have already prepared the "skeleton" of the game using a library called **pygame**, and also made some boring technical things ready for you such as reading the keyboard and game window settings.

You will have 3 files but **you will be interested in only one** in this exercise. I will still give a short description about what each file does for those who are *overly interested.*

1. run_the_game.py: This file contains the necessary code to open a game window, update the window etc.
2. space_shuttle.py: This is a *class* to create the space shuttle and the enemiesin the game. Everything appears on the screen are defined here originally. Whenever you write "shuttle.*SOMETHING*", it actually reaches the variables/methods in this class. Since we haven't covered the classes during the lecture, you can think about this file as a *meta-data* and completely ignore its existence.
3. **game_play.py:** This is the ONLY file that we will be editing today. We will program how to make the shuttle move, shoot, spawn enemies, catching if the enemy is being shot etc.. We will also program how the game ends if the shuttle is being hit by the enemy bullets if we have enough time.

How to run:

The code that you have already will start the *game engine* without any space shuttles, enemies and nothing has implemented in terms of the game play. To start it, just run the script called **run_the_game.py**. You will always be running this script even if you are editing the script called game_play.py.

Steps:

1. Define a list called shuttle.body. It will hold [x, y] pixels of our space shuttle.
    # In the snake game, there were many pixels to define the body
    # Here we have only one.
   You don't need a nested list.
2. Define a list called shuttle.enemies. It will hold [x, y] pixels of the enemIES (plural here 🙂)
    # In the snake game, there were only one pixel to define the fruit
    # Here we several enemies.
   You need a nested list.
3. You can change the color and the size of the objects on the screen if you want.
   Check this link for more colors:
   https://www.pygame.org/docs/ref/color_list.html
   Or you can define your own color as a combination of red-green-blue: color = (r,g,b)
    # use shuttle.X_color or shuttle.X_size
    # X can be body/enemy/bullet/screen/enemy_bullet
4. Similar to the snake game, move your shuttle as you press WASD keys.
5. It is time to shoot. There will be two steps:

   Step-1: Fill the "shuttle.bullets" as you press "space" key.
   Step-2: Remove the bullets from the list as they leave the screen.
6. 🌶️To add extra challenge, make your enemies move left and right. Let's say they will move 250 pixels right and then 250 pixels left.

   You can use the variable "**shuttle.enemy_move_offset**" to keep the value of how much enemies moved from their initial location.

    # You should define a max movement both to the right and left direction.
    # SUGGESTION:
    #   You can first check which direction enemy moves (shuttle.enemy_direction)
    #   and make the pixels move accordingly in another if statement.

7. Hit the enemies with your bullet. It is the same principle as eating fruits in snake game. If the pixel of one of the **shuttle.bullets** is in vicinity pixel of one of the **shuttle.enemies**, remove that enemy from the **shuttle.enemies** list.
8. Enemies should be able to shoot, otherwise there is no way to lose the game. Make one enemy shoot in every five seconds.

   You can get the current time with this command:

```
current_time = time.time()
```

   You can select a random enemy with this command:

```
enemy = random.choice(shuttle.enemies)
```

9. If a bullet hits you, then the game is over.

Note to gizem: https://www.notion.so/Space-shooter-game-ING202-11b4a87c257a809689f7cab80e710e78?pvs=4