SPACE COMBAT

2D GAME USING PYTHON

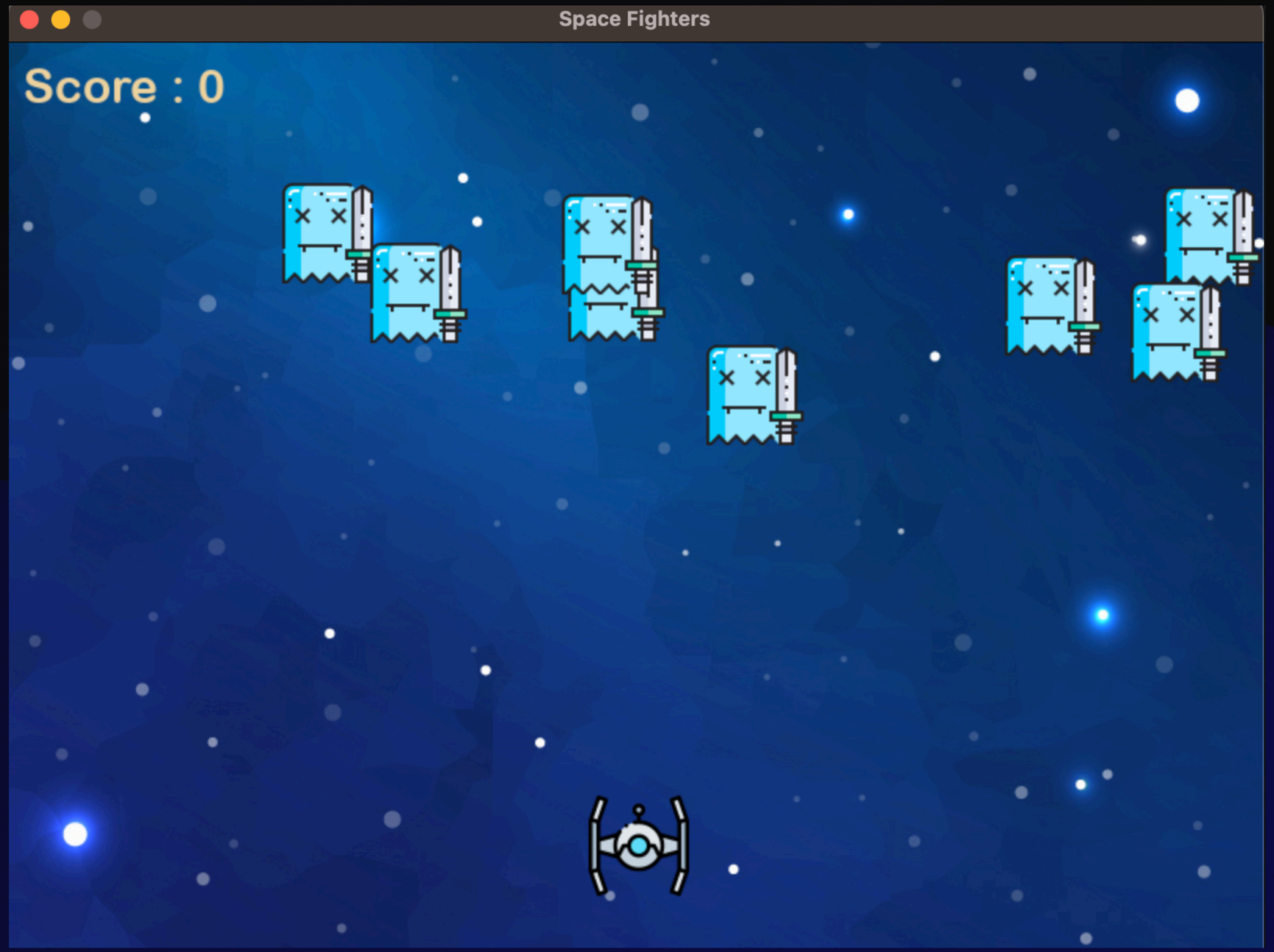
SUBMITTED BY :

SMARAK MARASINI (086)
SASHANK JOSHI (078)
RAJ ADHIKARI KHATRI (064)

SUBMITTED TO :

DHIRAJ PYAKUREL

After debugging the code, the game looks like this:



- This game is basic *2D* game coded on python using pygame library.

- It shows and explains the basic concept of game development, pixel and event counters, rendering, control keys, Key-framing, and concept of how a normal 2D game works using the method of ‘AABB’ - “Axis-Aligned Bounding Box” in collision detections of ‘bullet and enemy’ to keep the score counter, and display it on screen in real-time.

- This project uses the concept of *queue* and *sorting algorithms* as in DSA.

- The game starts automatically and the enemies pops in the screen on random location in top section and starts to move from right to left and vice versa, inside the horizontal boundary.

- The spaceship is fixed in bottom section and can move ‘left or right’ using the respective arrow key.

- The spaceship fires bullet on pressing ‘space’ key and if it hits the enemy, the score increase by one, and the enemy disappear.

- The game loop and restarts every time ‘enter’ key is pressed and the score count restore to zero.

- The game ends if one of enemies reaches the bottom part of the game, and the score is stored in queue.
- The queue will hold high score from every game and sort it in descending order.
- The score-list of top 5 scores are displayed at the end of the game.

There is an interesting feature to select different types of spaceship and slight change in the bullet size and number.

User can change it pressing ‘1, 2 or 3’ key in the game!

- We have used queue to store enemy image their co-ordinates and scores.
- Queue is linear data structure that is based on FIFO order. Here, the data wh is pushed first is pulled out first.
- enemyImg[ ], ememyx[ ], enemy[ ], enemyx_change[ ], enemyy_change[ ], queue[ ], are the queues that we used.
- enemyImg[ ] is used to load the image of enemy.
- enemyx[ ] and enemy[ ] are used to store the x and y co-ordinate of the ene images.
- queue[ ] is used to store the score of the game.

→ We have used bubble sort to sort the scores in descending order and store it in the queue.

→ In bubble sort, each element is compared with its adjacent element and if the second element is larger than the position is interchanged Otherwise it is kept same. This process is repeated for all elements in the queue. Here, we have sorted in descending order.

Algorithm:

1. Initialization (set i = 0)
2. Repeat step 3 to 5 until i < n
3. set j = 0
4. repeat step 5 until j < n-i-1
5. if queue[j] < queue[j+1]:
    a. queue[j], queue[j+1] = queue[j+1], queue[j]
6. End

Thank you!!