Міністерство освіти і науки України

Національний технічний університет України «КПІ ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Програмування інтелектуальних інформаційних систем

**ЗВІТ**

до лабораторних робіт

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Виконав**  **студент** |  | Бакун Нікіта Валерійович |  |  |
|  |  | (№ групи, прізвище, ім’я, по батькові ) |  |  |
|  |  |  |  |  |
| **Прийняв** |  | ас. Очеретяний О. К. |  |  |
|  |  | (посада, прізвище, ім’я, по батькові ) |  |  |

Київ 2021

# **Завдання лабораторної роботи**

Завданням даної лаболаторної роботи було створення гри Space Inviders з такими етапами:

* Початок гри по натиску клавіші
* Створення корабля гравця
* Створення рахунку
* Створення ворогів
* Екран кінця гри

# **Опис використаних технологій**

В даній роботі використано мову Python та бібліотеку pygame. Для рандомізації використано бібліотеку random

# **Опис програмного коду**

В головному файлі проекту main.py зібрані основні частини для роботи проекту. У даному файлі реалізований перший фрейм застосунку та створення клавіш для руху корабля. Для більшого розуміння механіки та реалізації елементів розглянемо інші файли

import GlobalVariables as gv  
  
  
def main():  
 gv.CAN\_EDIT = True  
 gv.PG\_LIB.display.set\_caption(**"Space Invaders"**) *# Назва застосунку* run = True  
 clock = gv.PG\_LIB.time.Clock()  
 while run:  
 clock.tick(gv.FPS) *#Виставлено кількість оновлень у секунду* gv.FrameCreator\_LIB.updateFrame()  
 if gv.LIVES <= 0 or gv.GOOD\_SHIP.health <= 0: *# умови програшу* gv.LOST = True  
 gv.LOST\_COUNT += 1  
  
 if gv.LOST: *# час показання екрану програшу* if gv.LOST\_COUNT > gv.FPS \* 3:  
 run = False  
  
 else:  
 continue  
 if len(gv.ENEMIES) == 0: *# збільшення рівня складності* gv.LEVEL += 1  
 gv.WAVE\_LENGTH += 5  
 for i in range(gv.WAVE\_LENGTH): *# створення масиву з ворогів* enemy = gv.ShipCreator.Enemy(gv.RANDOM\_LIB.randrange(50, gv.WIDTH - 100),  
 gv.RANDOM\_LIB.randrange(-1500, -100),  
 gv.RANDOM\_LIB.choice([**"red"**, **"blue"**, **"purple"**])) *# створення місця та колір ворога* gv.ENEMIES.append(enemy)  
 for event in gv.PG\_LIB.event.get():  
 if event.type == gv.PG\_LIB.QUIT:  
 run = False  
  
 keys = gv.PG\_LIB.key.get\_pressed() *# рух корабля по натиску на клавішу та стрільба* if keys[gv.PG\_LIB.K\_a] and gv.GOOD\_SHIP.x - gv.PLAYER\_VEL > 0: *# left* gv.GOOD\_SHIP.x -= gv.PLAYER\_VEL  
 if keys[gv.PG\_LIB.K\_d] and gv.GOOD\_SHIP.x + gv.PLAYER\_VEL + gv.GOOD\_SHIP\_SIZEX < gv.WIDTH: *# right* gv.GOOD\_SHIP.x += gv.PLAYER\_VEL  
 if keys[gv.PG\_LIB.K\_w] and gv.GOOD\_SHIP.y - gv.PLAYER\_VEL > 0: *# up* gv.GOOD\_SHIP.y -= gv.PLAYER\_VEL  
 if keys[gv.PG\_LIB.K\_s] and gv.GOOD\_SHIP.y + gv.PLAYER\_VEL + gv.GOOD\_SHIP\_SIZEY < gv.HEIGHT: *# down* gv.GOOD\_SHIP.y += gv.PLAYER\_VEL  
 if keys[gv.PG\_LIB.K\_SPACE]:  
 gv.GOOD\_SHIP.shoot()  
 for enemy in gv.ENEMIES[:]:  
 enemy.move(gv.ENEMY\_VEL)  
 enemy.move\_lasers(gv.LASER\_VEL, gv.GOOD\_SHIP)  
  
 if gv.RANDOM\_LIB.randrange(0, 20) == 1: *# рандомізація пострілу ворога* enemy.shoot()  
  
 if gv.LaserCreator.collide(enemy, gv.GOOD\_SHIP): *# дотик до ворога* gv.GOOD\_SHIP.health -= 10  
 gv.ENEMIES.remove(enemy)  
 gv.SCORE -= 20  
  
 if enemy.y + gv.BAD\_SHIP\_SIZEY + 10 > gv.HEIGHT: *# проходження ворогу до низу екрану* gv.LIVES -= 1  
 gv.ENEMIES.remove(enemy)  
 gv.SCORE -= 100  
  
 gv.GOOD\_SHIP.move\_lasers(-gv.LASER\_VEL, gv.ENEMIES)  
  
 gv.FrameCreator\_LIB.updateFrame()  
  
  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 title\_font = gv.PG\_LIB.font.SysFont(**"comicsans"**, 70)  
 run = True  
 main\_menu = True  
 while run:  
 gv.WINDOW.blit(gv.BACKGROUND\_PNG, (0, 0)) *#створення графіки для гри* title\_label = title\_font.render(**"Press any button to begin..."**, 1, (255, 255, 255))  
  
 gv.WINDOW.blit(title\_label, (gv.WIDTH / 2 - title\_label.get\_width() / 2,350))  
 if main\_menu:  
 main\_menu = False  
  
 gv.PG\_LIB.display.update()  
 for event in gv.PG\_LIB.event.get():  
 if event.type == gv.PG\_LIB.QUIT:  
 run = False  
 if event.type == gv.PG\_LIB.KEYDOWN or event.type == gv.PG\_LIB.MOUSEBUTTONDOWN:  
 main()  
 gv.PG\_LIB.quit()

У файлі LaserCreator.py реалізовано клас лазер та показано методи та поля даного класу.

import GlobalVariables  
  
  
class Laser:  
 def \_\_init\_\_(self, x, y, img):  
 self.x = x  
 self.y = y  
 self.img = img  
 self.mask = GlobalVariables.PG\_LIB.mask.from\_surface(self.img)  
  
 def draw(self, window): *# створення лазеру* window.blit(self.img, (self.x, self.y))  
  
 def move(self, vel): *# рух лазеру* self.y += vel  
  
 def off\_screen(self, height): *# вихід лазеру за рамки* return not (self.y <= height and self.y >= 0)  
  
 def collision(self, obj): *# дотикання до лазеру* return collide(self, obj)  
  
  
def collide(obj1, obj2): *# чи дотикнулись об'єкти* offset\_x = obj2.x - obj1.x  
 offset\_y = obj2.y - obj1.y  
 return obj1.mask.overlap(obj2.mask, (offset\_x, offset\_y)) is not None

У файлі GlobalVariables.py розміщені усі глобальні змінні. Вони винесені в окремий файл для зручного використання та доступу. В подальшому таке розміщення повинно спростити роботу з проектом

import pygame  
import os  
import time  
import random  
import FrameCreator  
import ShipCreator as sc  
import LaserCreator  
*# created global variables of libraries*OS\_LIB = os  
PG\_LIB = pygame  
TIME\_LIB = time  
RANDOM\_LIB = random  
FrameCreator\_LIB = FrameCreator  
ShipCreator = sc  
LaserCreator = LaserCreator  
*# Created global variables of Integer values*LOST = False  
FPS = 60  
LEVEL = 1  
LIVES = 5  
SCORE = 0  
WIDTH, HEIGHT = 750, 750  
PLAYER\_VEL = 5  
LASER\_VEL = 4  
ENEMIES = []  
WAVE\_LENGTH = 5  
ENEMY\_VEL = 1  
LOST\_COUNT = 0  
GOOD\_SHIP\_SIZEX, GOOD\_SHIP\_SIZEY = 50, 50  
BAD\_SHIP\_SIZEX, BAD\_SHIP\_SIZEY = 40, 40  
LASER\_SIZEX = 10  
LASER\_SIZEY = 20  
*# Created global variables of Font*PG\_LIB.font.init()  
MAIN\_FONT = PG\_LIB.font.SysFont(**"comicsans"**, 50)  
LOST\_FONT = PG\_LIB.font.SysFont(**"comicsans"**, 60)  
  
WINDOW = PG\_LIB.display.set\_mode((WIDTH, HEIGHT))  
  
*# Created global variables of boats images*GOOD\_SHIP\_PNG = PG\_LIB.transform.scale(PG\_LIB.image.load(os.path.join(**"assets"**, **"good\_ship.png"**)), (GOOD\_SHIP\_SIZEX, GOOD\_SHIP\_SIZEY))  
  
BAD\_SHIP\_RED\_PNG = PG\_LIB.transform.scale(PG\_LIB.image.load(os.path.join(**"assets"**, **"bad\_ship\_red.png"**)), (BAD\_SHIP\_SIZEX, BAD\_SHIP\_SIZEX))  
BAD\_SHIP\_BLUE\_PNG = PG\_LIB.transform.scale(PG\_LIB.image.load(os.path.join(**"assets"**, **"bad\_ship\_blue.png"**)), (BAD\_SHIP\_SIZEX, BAD\_SHIP\_SIZEX))  
BAD\_SHIP\_PURPLE\_PNG = PG\_LIB.transform.scale(PG\_LIB.image.load(os.path.join(**"assets"**, **"bad\_ship\_purple.png"**)), (BAD\_SHIP\_SIZEX, BAD\_SHIP\_SIZEX))  
  
BAD\_BULLET\_PNG = PG\_LIB.transform.scale(PG\_LIB.image.load(os.path.join(**"assets"**, **"bad\_bullet.png"**)), (LASER\_SIZEX, LASER\_SIZEY))  
GOOD\_BULLET\_PNG = PG\_LIB.transform.scale(PG\_LIB.image.load(os.path.join(**"assets"**, **"good\_bullet.png"**)), (LASER\_SIZEX, LASER\_SIZEY))  
  
*# Created global variables of background images*BACKGROUND\_PNG = PG\_LIB.transform.scale(PG\_LIB.image.load(os.path.join(**"assets"**, **"background.png"**)), (WIDTH, HEIGHT))  
  
*# Created ships*GOOD\_SHIP = sc.Player(300, 650)  
  
COLOR\_MAP = {  
 **"red"**: BAD\_SHIP\_RED\_PNG,  
 **"blue"**: BAD\_SHIP\_BLUE\_PNG,  
 **"purple"**: BAD\_SHIP\_PURPLE\_PNG  
}

У файлі ShipCrearor.py реалізовано клас корабля користувача та корабля ворогів з особистими полями. Деякі методи переписані, оскільки рух та постріли відбуваються зеркально.

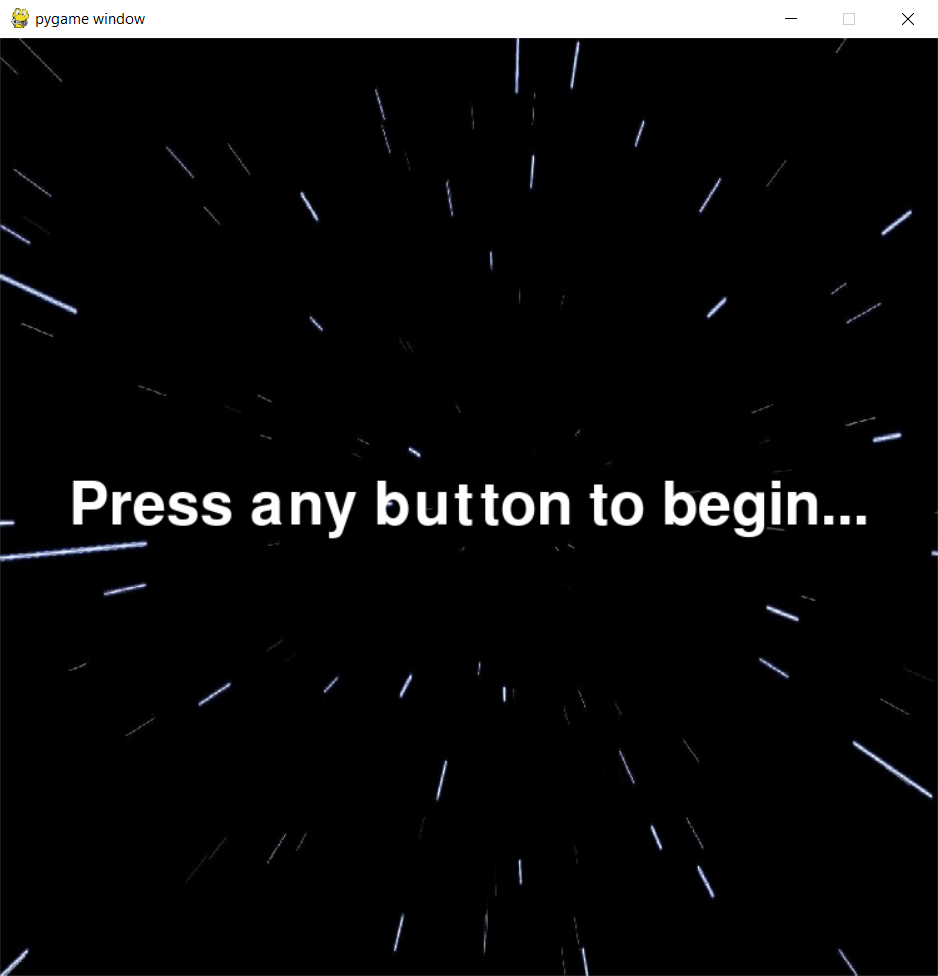
import GlobalVariables as gv  
  
  
class Ship:  
 COOLDOWN = 30 *# час перезарядки човна* def \_\_init\_\_(self, x, y, health=100):  
 self.x = x  
 self.y = y  
 self.health = health  
 self.ship\_img = None  
 self.laser\_img = None  
 self.lasers = []  
 self.cool\_down\_counter = 0  
  
 def draw(self, window): *# створення човна* gv.WINDOW.blit(self.ship\_img, (self.x, self.y))  
 for laser in self.lasers:  
 laser.draw(gv.WINDOW)  
  
 def move\_lasers(self, vel, obj): *# рух пострілів* self.coldown()  
 for laser in self.lasers:  
 laser.move(vel)  
 if laser.off\_screen(gv.HEIGHT):  
 self.lasers.remove(laser)  
 elif laser.collision(obj):  
 obj.health -= 10  
 self.lasers.remove(laser)  
  
 def shoot(self): *# постріл* if self.cool\_down\_counter == 0:  
 laser = gv.LaserCreator.Laser(self.x + 20, self.y, self.laser\_img)  
 self.lasers.append(laser)  
 self.cool\_down\_counter = 1  
  
 def coldown(self): *# механіка перезарядки* if self.cool\_down\_counter >= self.COOLDOWN:  
 self.cool\_down\_counter = 0  
 elif self.cool\_down\_counter >= 0:  
 self.cool\_down\_counter += 1  
  
  
class Player(Ship):  
 def \_\_init\_\_(self, x, y, health=100):  
 super().\_\_init\_\_(x, y, health)  
 self.ship\_img = gv.GOOD\_SHIP\_PNG  
 self.laser\_img = gv.GOOD\_BULLET\_PNG  
 self.mask = gv.PG\_LIB.mask.from\_surface(self.ship\_img)  
 self.max\_health = health  
  
 def move\_lasers(self, vel, objs): *#рух лазерів* self.coldown()  
 for laser in self.lasers:  
 laser.move(vel)  
 if laser.off\_screen(gv.HEIGHT):  
 self.lasers.remove(laser)  
 else:  
 for obj in objs:  
 if laser.collision(obj):  
 objs.remove(obj)  
 gv.SCORE += 50  
 if laser in self.lasers:  
 self.lasers.remove(laser)  
  
 def healthbar(self, window): *# створення здоров'я* gv.PG\_LIB.draw.rect(window, (255, 0, 0), (self.x, self.y + gv.GOOD\_SHIP\_SIZEY + 10, gv.GOOD\_SHIP\_SIZEX, 10))  
 gv.PG\_LIB.draw.rect(window, (0, 255, 0), (self.x, self.y + gv.GOOD\_SHIP\_SIZEY + 10, gv.GOOD\_SHIP\_SIZEX \* (  
 1 - ((self.max\_health - self.health) / self.max\_health)), 10))  
  
 def draw(self, window):  
 super().draw(window)  
 self.healthbar(window)  
  
  
class Enemy(Ship):  
 def \_\_init\_\_(self, x, y, color, health=100):  
 super().\_\_init\_\_(x, y, health)  
 self.ship\_img = gv.COLOR\_MAP[color]  
 self.laser\_img = gv.BAD\_BULLET\_PNG  
 self.mask = gv.PG\_LIB.mask.from\_surface(self.ship\_img)  
  
 def move(self, vel):  
 self.y += vel  
  
 def shoot(self):  
 if self.cool\_down\_counter == 0:  
 laser = gv.LaserCreator.Laser(self.x + 10, self.y, self.laser\_img)  
 self.lasers.append(laser)  
 self.cool\_down\_counter = 1

Останнім файлом застосунку є FrameCreator.py. У даному файлі знаходиться

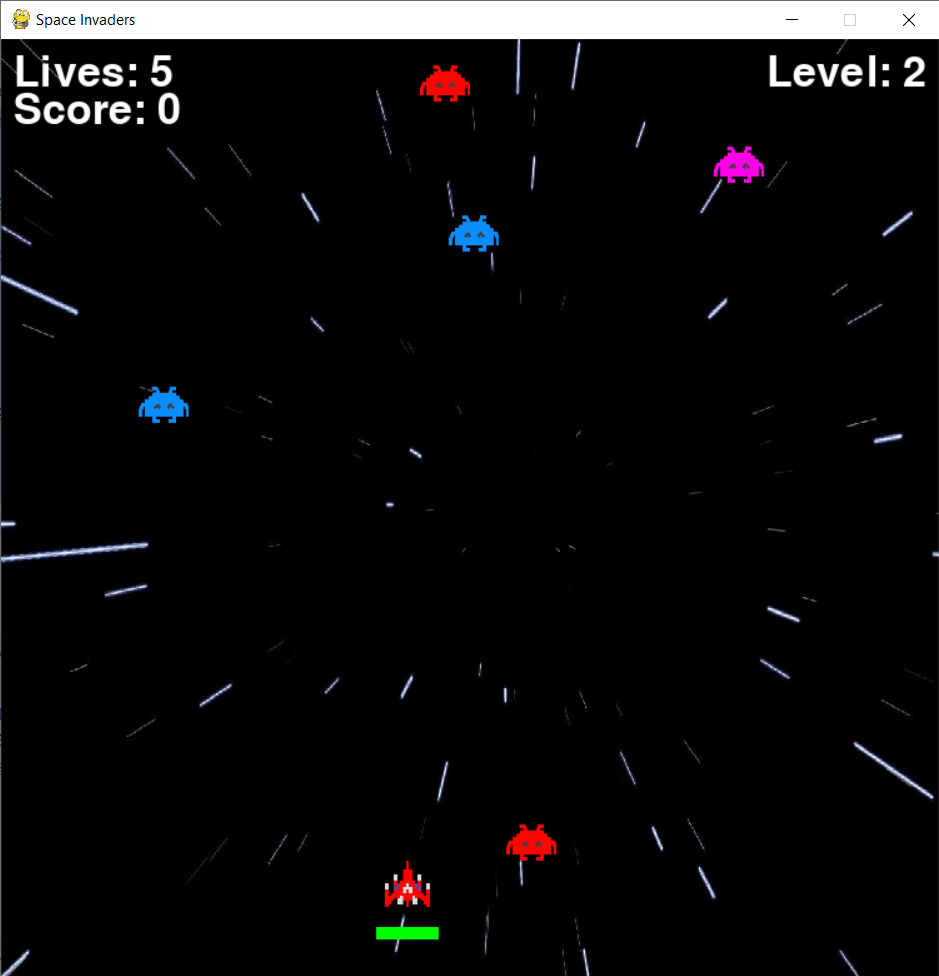
Функція, яка оновлює екран та створює гру після переходу на екран гри. Також тут знаходиться екран програшу.

import GlobalVariables as gv  
  
  
def updateFrame(): *# оновлення фрейму* gv.WINDOW.blit(gv.BACKGROUND\_PNG, (0, 0))  
  
 lives\_label = gv.MAIN\_FONT.render(**f"Lives:** {gv.LIVES}**"**, 1, (255, 255, 255))  
 level\_label = gv.MAIN\_FONT.render(**f"Level:** {gv.LEVEL}**"**, 1, (255, 255, 255))  
 score\_label = gv.MAIN\_FONT.render(**f"Score:** {gv.SCORE}**"**, 1, (255, 255, 255))  
 gv.WINDOW.blit(lives\_label, (10, 10))  
 gv.WINDOW.blit(level\_label, (gv.WIDTH - level\_label.get\_width() - 10, 10))  
 gv.WINDOW.blit(score\_label,(10,40))  
 for enemy in gv.ENEMIES:  
 enemy.draw(gv.WINDOW)  
  
 gv.GOOD\_SHIP.draw(gv.WINDOW)  
  
 if gv.LOST: *# програш* lost\_label = gv.LOST\_FONT.render(**f"YOU LOST!"**, 1, (255, 255, 255))  
 score\_label = gv.LOST\_FONT.render(**f"YOUR SCORE IS:** {gv.SCORE}**"**, 1, (255, 255, 255))  
 gv.WINDOW.blit(lost\_label, (gv.WIDTH / 2 - lost\_label.get\_width() / 2, 250))  
 gv.WINDOW.blit(score\_label, (gv.WIDTH / 2 - score\_label.get\_width() / 2, 350))  
  
 gv.PG\_LIB.display.update()

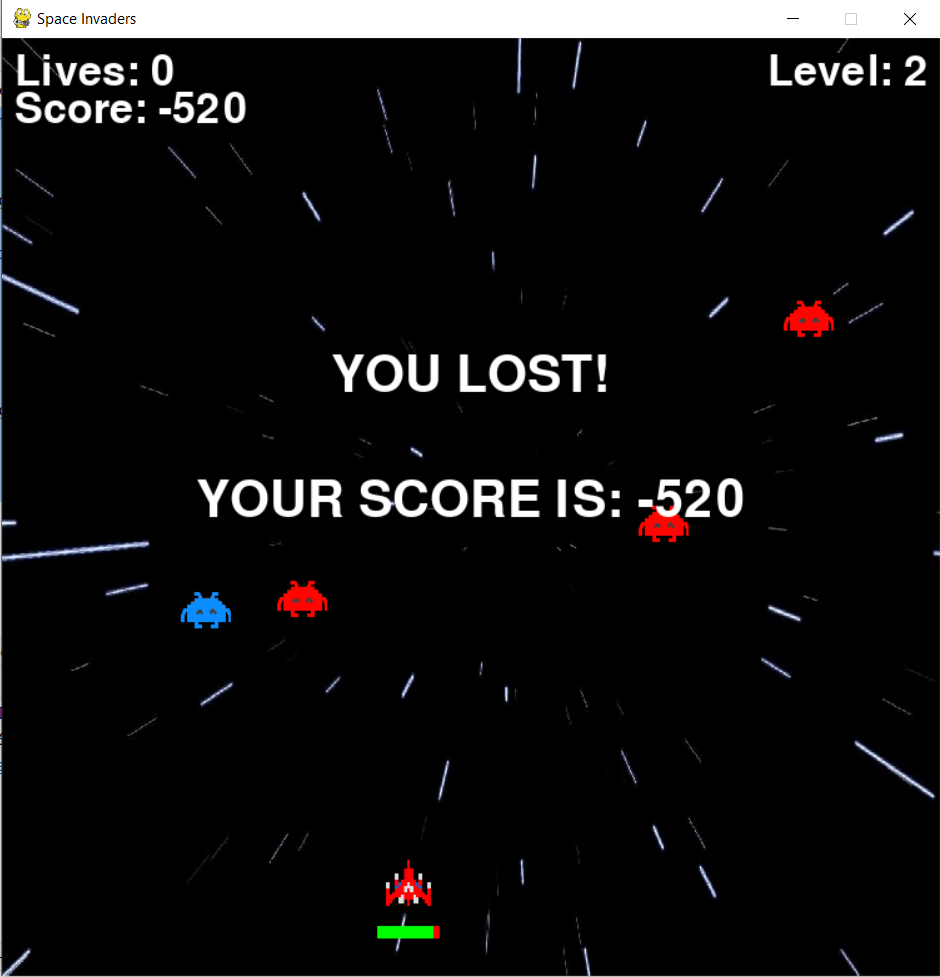
# **Скріншоти роботи програмного застосунку**



1 – Початкове меню гри



2 – момент з гри



3 – фрейм програшу