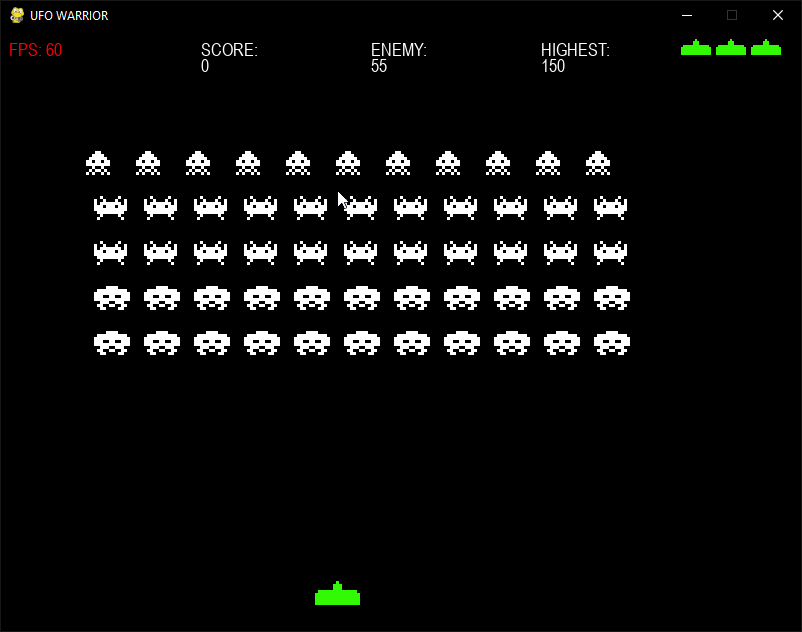
Проект реализовал студент 3 курса группы БСБО-07-18 Шалаев Артем.

Игра **UFO WARRIOR**

Введение в игру:

Игрок управляет движением и стрельбой космического корабля с помощью мыши. Если все враги будут уничтожены до того, как враг достигнет нижней части игрового интерфейса, игра будет выиграна, иначе игра завершится неудачей.



Основные классы игры являются (находятся в Utils.py):

* Класс аircraftSprite() - **Летающий корабль**. Наш космический корабль управляется с помощью мыши, и он должен иметь возможность стрелять, двигаться и взрываться после удара врага.
* Класс enemySprite() – **Класс нашего врага**. Враг автоматически управляется компьютером, и он должен иметь возможность стрелять, двигаться и взрываться после удара с нашей стороны.
* Класс ufoSprite() – **НЛО.** НЛО управляется компьютером и может рассматриваться как пасхальное яйцо. Если вы ударите это пасхальное яйцо, вы получите много бонусных очков. Когда бонусные очки накопятся до определенной суммы, вы получите дополнительное здоровье. НЛО нужно только иметь возможность двигаться и взрываться после удара с нашей стороны.
* Класс myBulletSprite() – **Пули Летающего корабля**

Класс enemyBulletSprite**() – Пули наших врагов**

* И наш космический корабль, и противник должны стрелять пулями, и они должны иметь возможность непрерывно двигаться вперед.

Какие вещи реализованы в игре:

1. pygame.mixer – Музыкальное сопровождение игры.
2. Реализация GAME OVER - def endInterface() с выводом о поражении.
3. showLife() - Показывает количество кораблей, равное значению здоровья (верхний правый угол).
4. shot() – стрельба.
5. boom() - Взрыв после столкновения.
6. resetBoom() – Сбрасывает данные, после взрыва (Используется при ударах и взрывах. Когда вы умираете, идет эффект смерти)
7. draw() – отрисовка
8. if not os.path.isfile('score') – создание файла (save) в котором хранится лучший результ.
9. Move() – функция движеня
10. ShowLife() - Показать оставшееся здоровье
11. Каждое увеличение очков на 2000 увеличивает нашу жизнь на одну жизнь

*if* (myaircraft.score % 2000 == 0) *and* (myaircraft.score > 0) *and* (myaircraft.score != myaircraft.old\_score):  
 myaircraft.old\_score = myaircraft.score  
 myaircraft.num\_life = *min*(myaircraft.num\_life + 1, myaircraft.max\_num\_life)