

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский Авиационный Институт»
(Национальный Исследовательский Университет)

Институт: №8
«Компьютерные науки и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Реферат по курсу
«Фундаментальная информатика»
I семестр

Тема:
«Разработка игры на языке программирования Python»

Группа:	М8О-109Б-22
Студент:	Концебалов О.С.
Преподаватель:	Сысоев М.А.
Оценка:	
Дата:	

Москва, 2022

Содержание:

1. План выполнения работы
2. История оригинальной игры «Space Invaders»
3. «MAI Invaders» — моя пародия на «Space Invaders» (идеи, код, модельки, проблемы с которыми столкнулся)
4. Заключение

1.План выполнения работы

1. Сбор информации по игре «Space Invaders», история создания, интересные моменты
2. Вспоминаю язык Python, его синтаксис и основные конструкции
3. Изучаю что такое Class в Python'е, как с ним работать
4. Собираю информацию о библиотеке Pygame, основных ее методах
5. Создания моделей для игры
6. Написание основного кода игры, разбитого на файлы
7. Тестирование итоговой игры и написание отчета

2.История оригинальной игры «Space Invaders»

Общая информация:

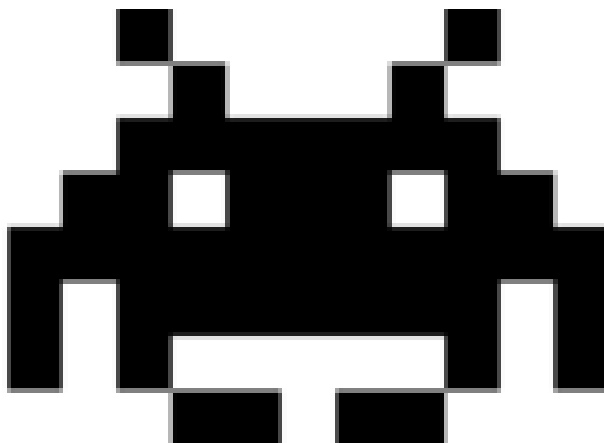
Space Invaders — видеоигра, разработанная Томохиро Нисикадо и выпущенная в 1978 году на аркадных автоматах. Изначально автоматы собирались компанией Taito и продавались только в Японии. Однако позже компания Midway приобрела права на распространение автоматов на территории США.



Space Invaders, несмотря на свою простоту, является основоположницей всех современных игр и помогла игровой индустрии стать одной из самых крупных и популярных в мире.

Через некоторое время, когда Space Invaders стала популярной, игра была перевыпущена на многие платформы, было создано несколько сиквелов (Space Invaders Part II, Return of the Invaders, Super Space Invaders, Mininvaders, Space Invaders DX, Space Invaders '95 от Nintendo, Space Invaders The Beat Attacker,

Space Invaders Frenzy). Версия для Atari 2600 увеличила продажи консоли в четыре раза и стала первой killer app (игрой, которая увеличивает продажи платформы, на которой она выпускается) в истории видеоигр. Space Invaders часто пародируется, является частью нескольких видеоигр. Пиксельное изображение инопланетянина стало символом поп-культуры и часто используется как обобщение видеоигры в целом.



Геймплей:

По жанру Space Invaders — shoot 'em up, в котором игрок управляет лазерной пушкой, передвигая её горизонтально, в нижней части экрана, а также отстреливая инопланетян, надвигающихся сверху экрана. Целью игры является уничтожение пяти рядов по одиннадцать инопланетян, которые двигаются горизонтально, а также вертикально, по направлению к низу экрана. Игрок имеет бесконечное количество патронов. Попадая в инопланетянина, игрок уничтожает его, за что получает очки. При уничтожении инопланетян увеличивается скорость движения оставшихся, а также ускоряется темп звуковых эффектов. При уничтожении всех инопланетян появляется новая, ещё более сильная волна, а игрок получает одно дополнительное очко жизни. Количество новых волн инопланетян неограниченно, что делает игру бесконечной.

Инопланетяне пытаются уничтожить пушку, стреляя по ней. При

попадании в пушку она уничтожается, а количество жизней уменьшается на одну единицу. Если количество жизней становится равным нулю, то игра заканчивается. При достижении хотя бы одним из противников нижней части экрана происходит инопланетный захват, и игра также заканчивается. Периодически в верхней части экрана пролетает «таинственный корабль», за уничтожение которого дают бонусные очки. Лазерная пушка частично защищена несколькими защитными бункерами, число которых отличается в зависимости от версии игры. Эти бункеры могут быть уничтожены игроком и инопланетными захватчиками.

Разработка:

Space Invaders была создана японским разработчиком компьютерных игр Томохиро Нисикадо. На создание игры и оборудования к ней Нисикадо потратил год. Сообщается, что игра основана на различных источниках, среди которых — электромеханическая игра Space Monsters, выпущенная Taito в 1972 году. Однако сам Нисикадо в качестве источника вдохновения привёл игру Breakout, выпущенную Atari в 1976 году. Также он заявляет, что его целью было создание игры с похожей на Breakout игровой механикой, но в то же время с улучшенной графикой. В итоге он изменил механику игры, но оставил вид сверху.

Изначально в игре должны были появиться различные виды врагов, такие как танки, боевая авиация и космические корабли. Однако Нисикадо так и не смог определиться с их движениями: технические ограничения затрудняли симуляцию полёта. Симуляция движений людей была проще, но Нисикадо считал расстрел людей аморальным. Увидев журнал, в котором рассказывалось о Звёздных Войнах, он решил использовать космический стиль. Нисикадо черпал вдохновение из романа Герберта Уэллса «Война миров», киноадаптацию которого он посмотрел, будучи ребёнком. Первый концепт-арт, нарисованный им, — осьминогоподобный инопланетянин. Позже были нарисованы другие монстры, смоделированные на основе

кальмаров и крабов. Первоначальное название игры — Space Monsters, придуманное по аналогии с популярной в то время в Японии песней Monster, было изменено на Space Invaders по просьбе начальства Нисикадо.

Аппаратное обеспечение:

Поскольку на момент создания игры мощность микрокомпьютеров в Японии не была достаточной для задач, связанных с программированием, Нисикадо пришлось разработать собственные аппаратные средства и программное обеспечение. Используя новые микропроцессоры из США, он создал собственную аркадную систему. Она использовала микропроцессор Intel 8080, ЭЛТ-монитор и монофонический звукогенератор. Несмотря на специально разработанное аппаратное обеспечение, Нисикадо так и не смог запрограммировать игру так, как этого хотел. Поэтому создание аппаратной части являлось самой трудной частью всей разработки. Процессор плохо справлялся с отрисовкой множества спрайтов. Когда во время игры пришельцев становилось меньше, процессор начинал отрисовывать экран быстрее, из-за чего игровой процесс начинал заметно ускоряться. Заметив это, Нисикадо не стал ограничивать скорость отрисовки программными средствами, решив использовать данный недостаток в качестве усложняющего фактора при прохождении уровня.

Первоначально Space Invaders была выпущена в горизонтальном варианте корпуса, с чёрно-белым экраном. Однако в США автомат выпускался в вертикальном варианте. В вертикальной версии автомата использовались оранжевые и зелёные полоски целлофана для симуляции цветной графики на чёрно-белом экране. В качестве фона для экрана использовалось изображение луны в космосе. Позже в японской версии также стал использоваться цветной целлофан. В оформлении автомата используется изображение человекоподобного монстра, который не попал в игру. Нисикадо заявляет, что это изображение основано на первоначальном названии игры —

«Космические Монстры»



Влияние Space Invaders:

Через несколько месяцев после релиза в Японии игра стала очень популярной. За первые несколько лет компания Taito создала 100 000 аркадных автоматов для японского рынка. Ещё 60 000 автоматов были проданы в США. С тех пор аркадные автоматы стали предметом коллекционирования, а горизонтальные варианты автоматов с игрой стали редкостью. Игра вызвала дефицит монет номиналом в 100 иен, использовавшихся в монетоприёмниках автоматов, после которого произошёл экономический рост. Хотя за рубежом Space Invaders не была так успешна, как в Японии, она принесла достаточно прибыли для того, чтобы помочь Taito расширить своё влияние в США. По состоянию на 2007 год, игра принесла доход в 500 миллионов долларов. Версия для Atari 2600, выпущенная в 1980 году, стала первой официально лицензированной аркадной игрой и первым мегахитом, увеличив продажи консоли в четыре раза.

Геймдизайнер Сигэру Миямото назвал Space Invaders революционной

игрой, хотя до неё он не интересовался видеоиграми. Ряд изданий связывают развитие игровой индустрии с успехом игры. Согласно The Observer, консольная версия игры, ставшая популярной, побудила пользователей изучать программирование, что впоследствии помогло многим из них стать лидерами индустрии. Сайт 1UP.com заявил, что Space Invaders доказала, что игры могут противостоять крупнейшим индустриям: кино, музыке и телевидению. В 2005 году сайт IGN составил список «The Most Important Games Ever Made», где Space Invaders заняла 3 место, а в 2007 году — рейтинг «Top 10 Most Influential Games», в котором игра заняла первое место. В 2008 году Книга рекордов Гиннесса признала Space Invaders лучшей аркадной игрой.

Ремейки и сиквелы:

Игра Space Invaders была переиздана на многочисленных платформах, было выпущено большое количество сиквелов. Переиздания включают в себя портированные и обновлённые версии оригинальной игры. Портированные версии отличаются графикой и такими дополнительными геймплейными опциями, как передвижные защитные бункеры, невидимые инопланетяне и режим кооперативной игры. Ранние портированные версии, среди которых порт для Atari 2600, использовали упрощённую графику по сравнению с оригиналом, в то время, как версии игры для Super Nintendo Entertainment System и PlayStation использовали улучшенную графику. Сиквелы привнесли в игру новые режимы и элементы дизайна. Например, Space Invaders Extreme, выпущенная для платформ Nintendo DS и PlayStation Portable, использует музыкальные элементы в стандартном геймплее. Space Invaders Get Even, спин-офф для WiiWare, позволяет игроку управлять инопланетянами. Портированные версии игры были неоднозначно приняты критиками. Версия игры для Atari 2600 была успешной, в то время как версия для Nintendo Entertainment System получила отрицательные отзывы.

Taito выпустила несколько продолжений игры, основанных на оригинальном дизайне. Первым продолжением стала игра Space Invaders Part II, выпущенная в 1980 году. В ней были добавлены цветная графика и некоторые геймплейные элементы. В США игра была выпущена под названием Deluxe Space Invaders, однако отличалась цветовой схемой и фоновым изображением. Следующим сиквелом стала игра Space Invaders II, выпущенная эксклюзивно для США в том же году. Она выпускалась только в горизонтальном варианте и предоставляла возможность кооперативной игры. Летом 1985 года была выпущена Return of the Invaders. В ней была улучшена графика, а также добавлены новые виды движений инопланетян. Следующими сиквелами стали Super Space Invaders '91, Space Invaders DX и Space Invaders '95. Каждая из них немного изменила традиционный геймплей. Как и оригинальная игра, некоторые из сиквелов стали предметом коллекционирования, а некоторые из них считаются редкими.

Игра и все сиквелы были представлены в юбилейном издании, Space Invaders Anniversary, выпущенном в 2003 году для PlayStation 2. Оно включало в себя девять различных вариантов игры Space Invaders

Отражение в массовой культуре:

Отсылки к игре появлялись в различных областях массовой культуры. Многочисленные телесериалы, среди которых «Шоу 70-х», «Клиника» и «Робоцып», содержат отсылки или пародии на игру и её элементы. Большое количество элементов игры используется в третьем сезоне «Футурамы», в серии Anthology of Interest II. На различных веб-сайтах и в различных журналах, таких, как Electronic Gaming Monthly и Arstechnica.com, пиксельное изображение инопланетянина используется как изображение видеоигр в целом. Кроме этого, на концерте Video Games Live использовались пиксельное изображение инопланетянина и музыка из игры. В честь 30-летия игры в 2008 году компания Taito спродюсировала альбом Space Invaders 2008. Альбом выпущен на лейбле Avex Trax и содержит музыку, вдохновлённую игрой.

Space Invaders появляются в фильмах «Черри-2000» (1987), «Терминатор 2: Судный день» (1991), «Пиксели» (2015), «Первому игроку приготовиться» (2018) и «Капитан Марвел» (2019). В 2014 году Warner Bros анонсировала фильм, полностью посвящённый компьютерной игре Space Invaders. Режиссёром фильма был назначен Акива Голдсман.

В 2006 году игра была выбрана в качестве представителя видеоигр в рамках проекта, составленного Японским агентством по делам культуры. В том же году Space Invaders была добавлена в экспозицию Game On, размещённую в Лондонском Музее Наук и предназначенную для демонстрации различных аспектов истории, разработки и культуры видеоигр. На фестивале Belluard Bollwerk International 2006, проведённом в Фрибуре, французско-швейцарский художник Гийом Реймонв рамках проекта Gameover создал трёхминутное видео, в котором была изображена игра Space Invaders, однако вместо пикселей использовались люди. На Games Convention 2008 была проведена выставка GH ART, на которой была представлена игра Invaders!, представляющая теракт 11 сентября в стиле Space Invaders. Однако позже автор игры, Дуглас Эдрик Стэнли, попросил убрать её с выставки в связи с критикой. Французский уличный художник, известный под псевдонимом Space Invader, сделал себе имя, создав по всему миру большое количество мозаичных рисунков, посвящённых игре Space Invaders.



3.«MAI Invaders» — моя пародия на «Space Invaders»

Дисклеймер

Все персонажи и события вымышленные и являются плодом моего гениального воображения. Любые совпадения с реальными личностями, живыми или мертвыми, случайны. Разработчик извиняется, если его проект задевает ваши чувства или как-либо оскорбляет вас, но ответственности за это не несет.

Идеи:

Когда я только получил это задание, направление в котором буду двигаться определил сразу. Я решил создать свою игру на языке программирования Python. Но осталась лишь одна проблема — определиться, что именно создавать, какую игру. Вариантов была масса: от банальной «Змейки» до шутера похожего на «Doom» или гоночной игры как на Тетрисе. Было много дней раздумий, но одним холодным ноябрьским вечером, общаясь с другом в мессенджере Telegram, я наткнулся на один крайне интересный стикер из набора MAI, который определил весь мой проект:



Как только я его увидел, сразу понял — это оно, то, что я искал все эти дни. В голове сразу стали появляться мысли как можно это реализовать. Я стал перебирать все игры, в которых можно использовать похожую модельку и

остановился на легендарной «Space Invaders» 1978 года. Это было переломным моментом, после которого началась активная работа над моим проектом.

Следующим сложным этапом стало придумывание названия будущей игры. Опять же в голове была масса вариантов, но все они мне не нравились. Тогда я решил остановиться на простом и лаконичном «MAI Invaders».

Теперь мне осталось придумать лишь внешность пришельцев (врагов) и оставшихся жизней. С этим проблем уже не возникло. Так как за основу, главного героя игры был взят стикер приведенный выше, то врагом стал Python, а жизнями C++ (по известным причинам).

Со звуками для игры определился быстро. Звук выстрела пушки — простой пистолетный выстрел. Звук смерти — легендарный звук ошибки Windows. Вступительный и завершающий звуки — цитаты легендарного Виталия Цаля, более известного под псевдонимом «Папич».

На этом идейно мой проект был завершен и осталась лишь реализовать его на практике.

Код игры:

Код игры был написан на языке программирования Python с использованием библиотек Pygame, Sys и Time. Код разбит на 8 файлов, которые связываются в единое целое с модельками и звуками. Запуск игры и все главные действия происходят в главном исполняемом файле mai_invaders_game_main.py

Основные файлы, код и их краткое описание:

1) mai_invaders_game_main.py — главный файл всей игры, выполняет все функции

```
import pygame as pg
import controls
from gun import Gun
from pygame.sprite import Group
from stats import Stats
from scores import Scores

FPS = 120
clock = pg.time.Clock()
```

```

def run():

    pg.init()
    screen = pg.display.set_mode((1200, 800))
    pg.display.set_caption("MAI Invaders")
    bg_color = (0, 0, 0)
    gun = Gun(screen)
    bullets = Group()
    aliens = Group()
    controls.create_army(screen, aliens)
    stats = Stats()
    sc = Scores(screen, stats)
    pg.mixer.music.load('Sounds/OpeningSound.mp3')
    pg.mixer.music.play()

    while True:
        controls.events(screen, gun, bullets)
        if stats.run_game:
            gun.update_gun()
            controls.screen_update(bg_color, screen, stats, sc, gun, aliens,
bullets)

            controls.update_bullets(screen, stats, sc, aliens, bullets)
            controls.update.aliens(stats, screen, sc, gun, aliens, bullets)
            clock.tick(FPS)

run()

```

2) controls.py — второй по важности файл игры, в нем происходит описание всех событий, которые происходят в игре, обновление экрана, коллизии, обновление рекордов

```

import pygame as pg
import sys
import time
from bullet import Bullet
from alien import Alien

def events(screen, gun, bullets):
    """event processing"""
    for event in pg.event.get():
        if event.type == pg.QUIT:
            sys.exit()
        elif event.type == pg.KEYDOWN:
            """movement to the right"""
            if event.key == pg.K_d or event.key == pg.K_RIGHT:
                gun.move_right = True
            elif event.key == pg.K_a or event.key == pg.K_LEFT:
                gun.move_left = True
            elif event.key == pg.K_SPACE:
                new_bullet = Bullet(screen, gun)
                bullets.add(new_bullet)
                pg.mixer.music.load('Sounds/ShootSound.mp3')
                pg.mixer.music.play()
        elif event.type == pg.KEYUP:
            """movement to the right"""
            if event.key == pg.K_d or event.key == pg.K_RIGHT:
                gun.move_right = False

```

```

        elif event.key == pg.K_a or event.key == pg.K_LEFT:
            gun.move_left = False

def screen_update(bg_color, screen, stats, sc, gun, aliens, bullets):
    """screen update"""
    screen.fill(bg_color)
    sc.show_score()
    for bullet in bullets.sprites():
        bullet.draw_bullet()
    gun.output()
    aliens.draw(screen)
    pg.display.flip()

def update_bullets(screen, stats, sc, aliens, bullets):
    """updating bullets positions"""
    bullets.update()
    for bullet in bullets.copy():
        if bullet.rect.bottom <= 0:
            bullets.remove(bullet)
    collisions = pg.sprite.groupcollide(bullets, aliens, True, True)
    if collisions:
        for aliens in collisions.values():
            stats.score += 10 * len(aliens)
        sc.image_score()
        check_high_score(stats, sc)
        sc.image_lives()
    if len(aliens) == 0:
        bullets.empty()
        create_army(screen, aliens)

def gun_kill(stats, screen, sc, gun, aliens, bullets):
    """gun and aliens collision"""
    if stats.guns_loss > 1:
        stats.guns_loss -= 1
        pg.mixer.music.load('Sounds/DeathSound.mp3')
        pg.mixer.music.play()
        sc.image_lives()
        aliens.empty()
        bullets.empty()
        create_army(screen, aliens)
        gun.create_new_gun()
        time.sleep(1)
    else:
        stats.run_game = False
        pg.mixer.music.load('Sounds/GameOverSound.mp3')
        pg.mixer.music.play()
        time.sleep(2)
        sys.exit()

def update_aliens(stats, screen, sc, gun, aliens, bullets):
    """updating aliens positions"""
    aliens.update()
    if pg.sprite.spritecollideany(gun, aliens):
        gun_kill(stats, screen, sc, gun, aliens, bullets)
        aliens_check(stats, screen, sc, gun, aliens, bullets)

def aliens_check(stats, screen, sc, gun, aliens, bullets):
    """checking if the aliens have reached the edge of the screen"""
    for alien in aliens.sprites():

```

```

        if alien.rect.bottom >= 700:
            gun_kill(stats, screen, sc, gun, aliens, bullets)
            break

def create_army(screen, aliens):
    """creating an alien army"""
    alien = Alien(screen)
    alien_width = alien.rect.width
    number_alien_x = int((1200 - 2 * alien_width) / alien_width)
    alien_height = alien.rect.height
    number_alien_y = int((800 - 100 - 2 * alien_height) / alien_height)

    for alien_row_number in range(number_alien_y - 5):
        for alien_number in range(number_alien_x):
            alien = Alien(screen)
            alien.x = alien_width + (alien_width * alien_number)
            alien.y = alien_height + (alien_height * alien_row_number)
            alien.rect.x = alien.x
            alien.rect.y = alien.rect.height + (alien.rect.height *
alien_row_number)
            aliens.add(alien)

def check_high_score(stats, sc):
    """checking new high scores"""
    if stats.score > stats.high_score:
        stats.high_score = stats.score
        sc.image_high_score()
    with open('Scores/HighScore.txt', 'w') as fp:
        fp.write(str(stats.high_score))

```

3) bullet.py — обработка выстрелов, отрисовка пуль и их перемещение на экране

```

import pygame as pg

class Bullet(pg.sprite.Sprite):

    def __init__(self, screen, gun):
        """create a bullet in the position of the gun"""
        super(Bullet, self).__init__()
        self.screen = screen
        self.rect = pg.Rect(0, 0, 2, 10)
        self.color = 255, 203, 14
        self.speed = 4.5
        self.rect.centerx = gun.rect.centerx
        self.rect.top = gun.rect.top
        self.y = float(self.rect.y)

    def update(self):
        """moving the bullet up"""
        self.y -= self.speed
        self.rect.y = self.y

    def draw_bullet(self):
        """bullet drawing"""
        pg.draw.rect(self.screen, self.color, self.rect)

```


4) gun.py — создание и отрисовка пушки на экране, обновление ее позиции и перемещение, создание новой пушки после разрушения

```
import pygame as pg
from pygame.sprite import Sprite

class Gun(Sprite):

    def __init__(self, screen):
        """gun initialization"""
        super(Gun, self).__init__()
        self.screen = screen
        self.image = pg.image.load('Images/GunToGame.png')
        self.rect = self.image.get_rect()
        self.screen_rect = screen.get_rect()
        self.rect.centerx = self.screen_rect.centerx
        self.center = float(self.rect.centerx)
        self.rect.bottom = self.screen_rect.bottom
        self.move_right = False
        self.move_left = False

    def output(self):
        """gun drawing"""
        self.screen.blit(self.image, self.rect)

    def update_gun(self):
        """gun position update"""
        if self.move_right and self.rect.right < self.screen_rect.right:
            self.center += 1.5
        if self.move_left and self.rect.left > 0:
            self.center -= 1.5

        self.rect.centerx = self.center

    def create_new_gun(self):
        """creates a new gun"""
        self.center = self.screen_rect.centerx
```

5) lives.py — отрисовка жизней в верхнем левом углу экрана

```
import pygame as pg
from pygame.sprite import Sprite

class Lives(Sprite):

    def __init__(self, screen):
        """lives initialization"""
        super(Lives, self).__init__()
        self.screen = screen
        self.image = pg.image.load('Images/Live.png')
        self.rect = self.image.get_rect()
        self.screen_rect = screen.get_rect()

    def output(self):
        """gun drawing"""
        self.screen.blit(self.image, self.rect)
```

6) alien.py — создание и отрисовка пришельцев, их перемещение

```
import pygame as pg

class Alien(pg.sprite.Sprite):
    """one alien class"""

    def __init__(self, screen):
        """initialize and spare initial position"""
        super(Alien, self).__init__()
        self.screen = screen
        self.image = pg.image.load('Images/AlienToGame.png')
        self.rect = self.image.get_rect()
        self.rect.x = self.rect.width
        self.rect.y = self.rect.height
        self.x = float(self.rect.x)
        self.y = float(self.rect.y)

    def draw(self):
        """alien drawing"""
        self.screen.blit(self.image, self.rect)

    def update(self):
        """alien position update"""
        self.y += 0.1
        self.rect.y = self.y
```

7) scores.py — обработка счета, его отрисовка и обновление, проверка на рекорд

```
import pygame.font
from lives import Lives
from pygame.sprite import Group

class Scores:
    """output game information"""
    def __init__(self, screen, stats):
        """scoring initialization"""
        self.screen = screen
        self.screen_rect = screen.get_rect()
        self.stats = stats
        self.text_color = (255, 255, 255)
        self.font = pygame.font.SysFont(None, 36)
        self.image_score()
        self.image_high_score()
        self.image_lives()

    def image_score(self):
        """converts score text to graphics"""
        self.score_img = self.font.render(str(self.stats.score), True,
self.text_color, (0, 0, 0))
        self.score_rect = self.score_img.get_rect()
        self.score_rect.right = self.screen_rect.right - 40
        self.score_rect.top = 20

    def image_high_score(self):
        """converts high score to graphics"""
```

```

        self.high_score_image = self.font.render(str(self.stats.high_score),
True, self.text_color, (0, 0, 0))
        self.high_score_rect = self.high_score_image.get_rect()
        self.high_score_rect.centerx = self.screen_rect.centerx
        self.high_score_rect.top = self.screen_rect.top + 20

    def image_lives(self):
        """number of lives"""
        self.lives = Group()
        for live_number in range(self.stats.guns_loss):
            live = Lives(self.screen)
            live.rect.x = 15 + live_number * live.rect.width
            live.rect.y = 20
            self.lives.add(live)

    def show_score(self):
        """score drawing"""
        self.screen.blit(self.score_img, self.score_rect)
        self.screen.blit(self.high_score_image, self.high_score_rect)
        self.lives.draw(self.screen)

```

8) stats.py — жизни пушки, вывод рекорда на экран, обработка статистики, которая изменяется по ходу игры

```

class Stats:
    """game statistics tracking"""

    def __init__(self):
        """initializes statistics"""
        self.reset_stats()
        self.run_game = True
        with open('Scores/HighScore.txt', 'r') as fp:
            self.high_score = int(fp.readline())

    def reset_stats(self):
        """statistics that change during the game"""
        self.guns_loss = 3
        self.score = 0

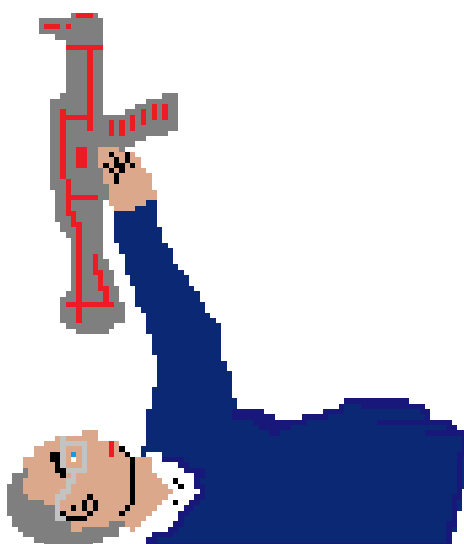
```

Модельки:

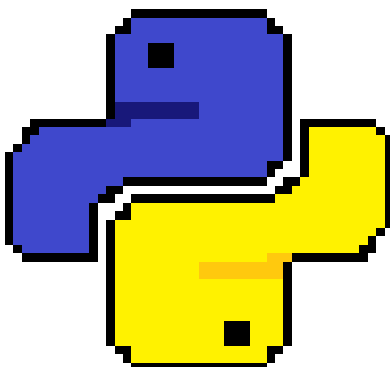
Модельки, нарисованные лично мной для игры. Каждая из них создавалась с душой и является моей интеллектуальной собственностью

1) GunToGame.png — пушка и по совместительству главный герой игры.

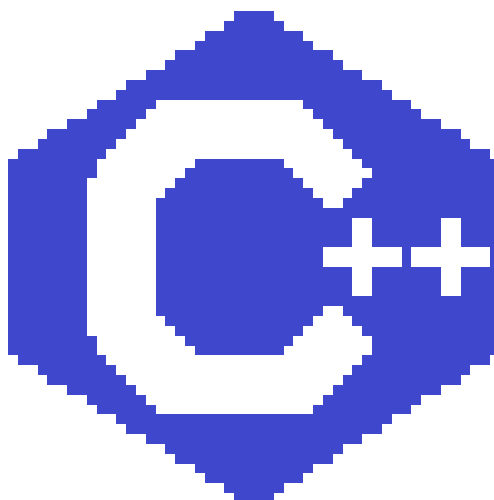
Примечание: смотри дисклеймер!



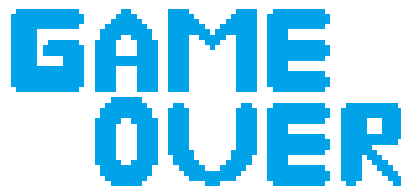
2) AlienToGame.png — заклятый враг главного героя



3) Live.png — жизни нашего главного героя, причина по которой он улыбается



4) GameOver.png — нарисовал, но не использовал



Проблемы:

Во время выполнения проекта я столкнулся с рядом проблем, которые приходилось фиксировать. Приходилось очень часто заходить на StackOverflow. Зато благодаря этому получилось немного подтянуть свой английский, как и в 10 лабораторной.

Часто вылазило много мелких багов, что-то не отрисовывалось, что-то отрисовывалось не там, где надо, двигалось быстрее чем надо. Много проблем возникло, когда прописывал коллизию (осталась одна фишка, не баг!!!).

Самая гениальная моя ошибка: когда при создании класса пушки прописал `__int__` вместо `__init__` и около полутора часа не понимал почему она не отрисовывается и искал ошибку.

Самым сложным на удивление оказалось установить библиотеку Pygame. PyCharm постоянно ругался, не хотел ничего устанавливать. Я обновлял pip (тоже не получилось), версию Python'a, залазил в PATH и делал что-то непонятное для меня, переустанавливал Python и PyCharm, долго сидел на StackOverflow. В итоге выяснил, что моя версия Python'a 3.11 не поддерживается этой библиотекой. Пришлось понизить версию Python'a до 3.9 и все заработало.

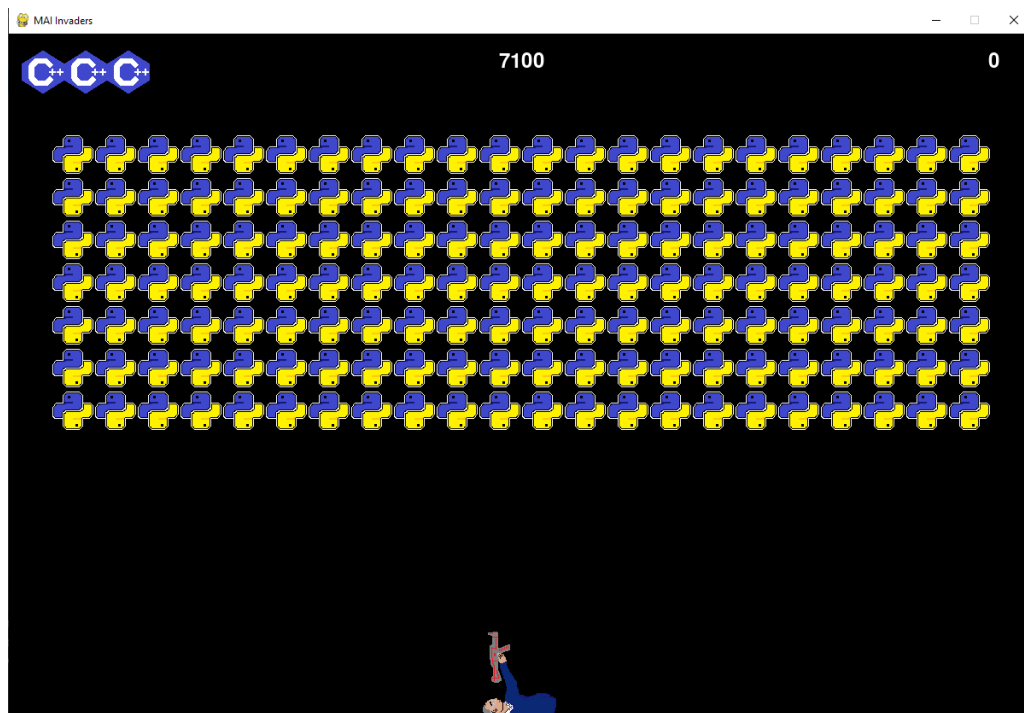
Также я пытался перевести весь файл игры в .exe. Но это закончилось неудачей. Пакеты pyinstaller ни в какую не хотели устанавливаться, искал решения в интернете, пытался повторить те же действия, что и при установке Pygame, но ничего не получилось и я сдался.

4. Заключение

Работать над этим проектом было очень интересно для меня. Благодаря ему я значительно прокачал свои навыки в программировании на Python'е (с практически 0 до пойдёт) и даже английский. получил небольшой опыт создания игр, понял, как все это происходит и насколько это непросто.

Создавать этот проект было крайне весело, особенно во время некоторых тестов и когда показывал наработки и модельки друзьям. Понравилось подбирать звуки для игры и рисовать модельки, почувствовал себя дизайнером-инвалидом.

В итоге получил готовую игру, но над ней еще можно долго работать и оттачивать некоторые моменты, доводя до идеала (AAA уровня ха-ха). Но для этого нужно время, которого у меня мало(



Отдельно хочется отметить актуальность и практическую пользу этого задания. Благодаря ему можно неплохо прокачаться в выбранной области, изучить какие-либо новые библиотеки и технологии, попробовать себя в определенных направлениях разработки. Это на голову выше простого доклада о языке программирования или об ученом/разработчике. За это

отдельный лайк, спасибо Макс.

Очень жалко, что идею с публичными защитами не получилось реализовать из-за нехватки времени. Хотелось посмотреть на проекты других людей, их код, идеи для представления своей работы.

Задание на 10/10 очень понравилось

В честь ВЕликого

