УДК 004(075.8)

2D-МОДЕЛИРОВАНИЕ. СОЗДАНИЕ ИГРЫ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

Жолдас Рамазан Бауыржанұлы
zoldasramazan0@gmail.com
Жарылкасынов Арнур Жанатович
arnur09.07.06@icloud.com
Студенты 1 курса образовательной программы «Информационные системы»
Алматинский технологический университет, г.Алматы, Республика Казахстан
Научный руководитель, магистр ИС, лектор – **Искакова А.Т.**

Информационные технологии охватывают широкий спектр деятельности, связанный с обработкой, передачей и хранением информации с использованием компьютерных технологий. Это включает в себя разработку программного обеспечения, создание веб-сайтов и приложений, администрирование компьютерных сетей, кибербезопасность, анализ данных и многое другое. ИТ играет ключевую роль в различных отраслях, современный мир не представим без его влияния.

Язык программирования Python является фундаментальным инструментом в арсенале любого разработчика и исследователя. Python прочно закрепился на вершине списка самых популярных и востребованных языков программирования, начиная с момента своего появления в конце 1980-х годов. Его успех объясняется не только элегантным и интуитивно понятным синтаксисом, но и широтой применения – от создания веб-приложений до научных вычислений и искусственным интеллектом. Основная привлекательная черта Python является его философия "читаемости кода", которая делает разработку и поддержку программного обеспечения более простой и эффективной. Это очень важно в современном мире, где коллаборация и обмен знаниями играют ключевую роль в разработке качественного программного продукта. Python обладает огромным сообществом разработчиков, которые постоянно создают новые библиотеки, фреймворки и инструменты, расширяя его функциональность и делая её еще более мощным и универсальным. Эта активная экосистема поддерживает рост популярности языка и его применение в самых разных областях от научных исследований до разработки приложений для мобильных устройств. В данной работе мы проведем глубокий анализ особенностей и преимуществ языка Python, рассмотрим его основные принципы и концепции, а также проанализируем его влияние на современную индустрию информационных технологий. Мы рассмотрим некоторые примеры успешного использования Python в реальных проектах и исследованиях, чтобы лучше понять его потенциал и перспективы развития в будущем.

В наши планы также входит создание двухмерной компьютерной игры с использованием языка программирования Python. Мы собираемся внимательно изучить все необходимые аспекты разработки игр, углубленно изучить язык программирования Python включая его основные концепции, структуры данных, алгоритмы и практическое применение в разработке игр является важным этапом в этом процессе.

Python это интерпретируемый язык программирования высокого уровня, появившийся в конце 1980-х годов. Его создатель, Гвидо ван Россум, поставил перед собой цель разработать простой, но мощный язык, который было бы легко читать и понимать.

История Python началась в декабре 1989 года, когда голландский программист Гвидо ванн Россум задался целью разработать новый язык программирования Python был основан на идее простоты и читабельности кода. Название языка было навеяно комедийным шоу Monty Python's Flying Circus, которое стало популярным в Великобритании в 1970-х годах.

Развитие и популяризация языка Python прошли несколько важных этапов с момента его первого выпуска в 1991 году: ключевым фактором популярности стала его простота, эффективность и мощный функционал, который привлек внимание широкого круга разработчиков и ИТ-инженеров.

Изначально Python широко использовался в академическом и научном сообществе благодаря своей простоте и гибкости, позволяя исследователям и ученым быстро решать разнообразные задачи вычислений и анализа данных.

Позже, благодаря мощным веб-фреймворкам, таким как Django и Flask, Python стал популярен среди веб-разработчиков, поскольку облегчал создание веб-приложений и сайтов.

Появление библиотек и фреймворков для анализа данных и машинного обучения, таких как NumPy, pandas, scikit-learn и TensorFlow, сделало Python одним из ведущих языков в области искусственного интеллекта и анализа данных.

Что можно написать на Python?

1. Программы, автоматизирующие выполнение таких задач, как поиск данных на веб-сайтах, автоматическая отправка электронных писем или составление расписаний.
2. Веб-приложения с использованием таких фреймворков, как Django и Flask. Эти приложения могут обрабатывать запросы пользователей, взаимодействовать с базами данных и возвращать результаты в браузер.
3. Анализ данных и машинное обучение с использованием таких библиотек, как Pandas, NumPy, scikit-learn и TensorFlow. Это включает в себя обработку и визуализацию данных, обучение моделей машинного обучения и оценку эффективности.
4. Создание игр на Python с использованием таких библиотек, как Pygame и Unity.
5. Разработка приложений для научных и инженерных вычислений с использованием таких библиотек, как SciPy и SymPy.
6. Системное администрирование, включая администрирование серверов, автоматизацию резервного копирования данных и написание скриптов для мониторинга сети.
7. Разработка мобильных приложений с использованием фреймворков Kivy или SL4A.
8. Создание ботов для социальных сетей, таких как Twitter и Telegram, для автоматической публикации контента и ответа на запросы пользователей.
9. Работа с базами данных, включая создание, чтение, запись и обновление данных, с использованием таких библиотек, как SQLite, MySQL и PostgreSQL.
10. Разработка клиент-серверных приложений и сетевых приложений, таких как чат-боты, с использованием таких библиотек, как socket и Twisted.
11. Разработка приложений для настольных компьютеров с использованием таких фреймворков, как PyQt и Tkinter. Эти приложения могут быть графическими пользовательскими интерфейсами (GUI), играми или инструментами обработки данных.
12. Создание скриптов для автоматизации рутинных компьютерных задач, таких как управление файлами, обработка текстов и архивирование данных.
13. Использование Python API для разработки расширений для других приложений, например, расширений для текстовых редакторов (например, Sublime Text или VS Code) или графических программ (например, Blender)
14. Писать программы для робототехники и автоматизации процессов с использованием библиотек robot framework и управлять оборудованием через GPIO (ввод/вывод общего назначения) на одноплатных компьютерах, таких как Raspberry Pi
15. Создавать инструменты для анализа и визуализации данных из социальных сетей, таких как Twitter и Facebook, используя API этих платформ
16. Разрабатывать приложения для Интернета вещей (IoT), например, для управления домашней автоматикой и сенсорными устройствами с помощью библиотеки MicroPython.
17. Поддержка и разработка систем управления контентом (CMS) с использованием Python и таких фреймворков, как Wagtail для Django и WordPress.

18. Создание инструментов для анализа текста и обработки естественного языка (NLP). Например, программы для анализа тональности текста, извлечения ключевых слов и классификации текстов.

19. Разработка систем контроля версий и совместной работы над кодом с использованием систем контроля версий, таких как Git, или хостинга кода на таких платформах, как GitHub, GitLab или Bitbucket.

20. Конечно, любой другой проект или приложение, которое может прийти в голову разработчику; Python предлагает обширный набор инструментов и библиотек для реализации практически любой идеи.

Компьютерные игры - это вид развлечений, в которых игроки управляют виртуальными персонажами или объектами в соответствии с определенными правилами и целями. Существует множество различных жанров игр, включая стратегии, симуляторы, экшены и головоломки.

История компьютерных игр началась в 1962 году с игры "Космическая война!", созданной для компьютера PDP-1. История компьютерных игр началась в середине XX века, когда появились такие простые игры, как По мере развития технологий компьютерных систем и игровых приставок игры становились все сложнее и разнообразнее.

1980-е годы, когда появились первые домашние игровые приставки, такие как Atari 2600 и Nintendo Entertainment System (NES), а также персональные компьютеры, стали золотым веком видеоигр. С тех пор игровая индустрия продолжала развиваться и внедрять новые технологии, такие как 3D-графика, онлайн-игры и виртуальная реальность.

Сегодня компьютерные игры стали огромным явлением, охватывающим многие аспекты культуры и общества. Они предлагают игрокам широкий спектр впечатлений, социального взаимодействия и возможностей для развлечения.

Компьютерные игры всегда поднимают людям настроение. Это связано с тем, что они обладают определенными характерными особенностями, с помощью которых они не дают скучать и дарят положительные эмоции игроку.

Поскольку в компьютерные игры могут играть как новички, так и профессионалы, стоит заранее ознакомиться с инструкциями, деталями и правилами, чтобы не столкнуться с проблемами. Также важно помнить, что у каждой игры есть системные требования.

Итак, приступаем к нашей работе. Первыми шагами к написанию нашей работы было ознакомление со всей нужной нам информацией, а именно с внутренней составляющей компьютерных игр и языком программирования Python. После чего мы начали прописывать определенные части своей работы, чтобы в последствии упростило нам написание нашего кода. Когда мы были готовы приступать к нашей работе, мы открыли свой редактор кода, который используем для написания программ на Python. Им является PyCharm, потому что он очень прост в чтении языка, имеет приятное и понятное для глаз оформление и удобен в устранении своих ошибок при написании кода программы. Далее мы создали файл своей работы, чтобы производить дальнейшие сохранения по ходу написания кода.

Написание нашей игры начиналось с выбора библиотек, классов и модулей. У Python существует 2 библиотеки, которые подходят для создания игр: Pygame и Arcade. Прочитав про каждую из них, мы решили остановиться на Pygame, потому что наша игра одноуровневая и не требует множества кнопок для управления, как, например, на платформере. Pygame — это «игровая библиотека», набор инструментов, помогающих программистам создавать игры. К ним относятся: графика, анимация, звук, управление. Конечно, создавать какую-то новейшую игру, которую никто никогда не делал мы не собираемся, ведь превыше всего для нас является лишь ознакомление с организацией процесса написания игры, а также усовершенствование этой игры на более яркую и интересную, с нашим кодом и нашими созданными моделями. Мы создали игру, чем-то схожую на очень старую классическую аркадную игру Space Invaders, созданную еще в 1978 году для аркадных автоматов. Наша же игра называется Космические воины.

Мы установили библиотеку Pygame, импортировали ее и начали писать код. Помимо этой библиотеки мы так же использовали модуль sys, который предоставляет доступ к переменным, используемым или поддерживаемым интерпретатором, а также к функциям, которые тесно взаимодействуют с интерпретаторами, а также Спрайт — это элемент компьютерной графики, представляющий объект на экране, который может двигаться. В двухмерной игре все, что вы видите на экране, является спрайтами. Начали мы с создания окна, цвета, текста и фона (Рисунок 1).

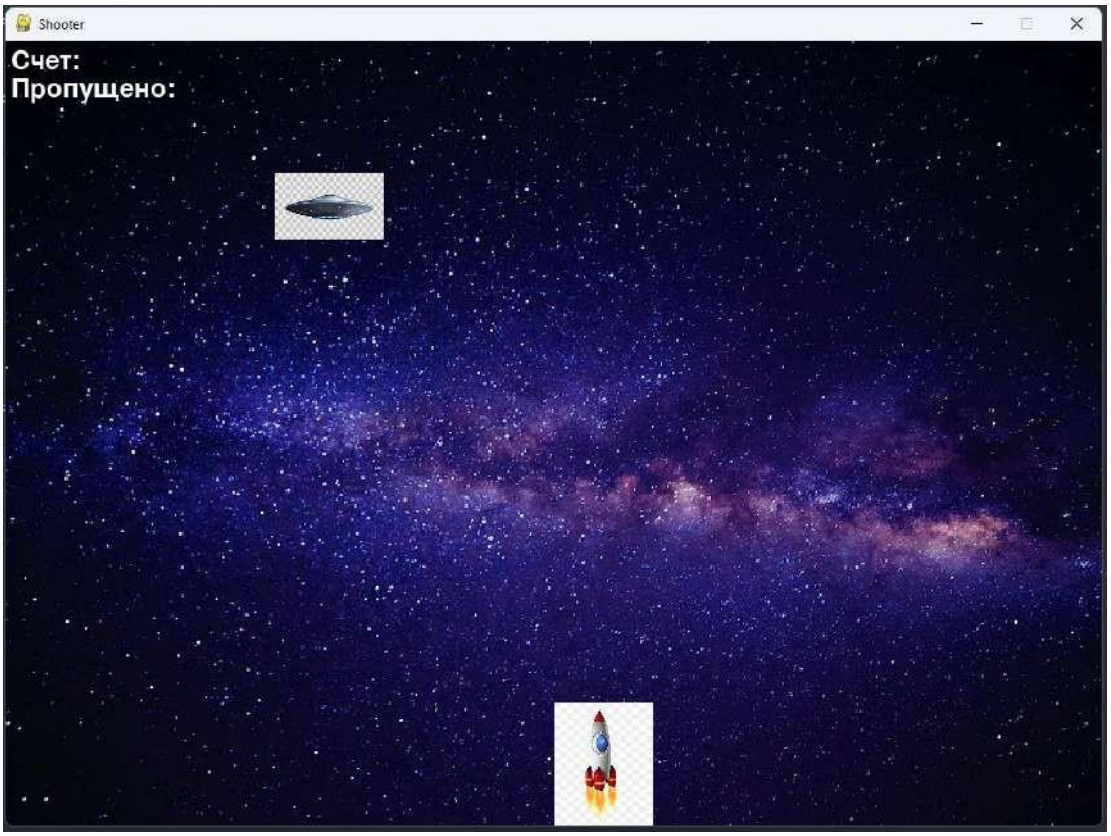

Рисунок 1. Создания окна, цвета, текста и фона.

Затем приступили к написанию кода работы пушки, управлению и основным скриптам (Рисунок 2, Рисунок 3 и Рисунок 4). Игра готова!

```
#============================================
#Display
window_width=1000
window_height=700
window=display.set_mode((window_width,window_height))
display.set_caption('Shooter')
#Text
font.init()
font=font.Font(None,36)
text=font.render("Счет: ",True,(255,255,255))
text2=font.render("Пропущено: ",True,(255,255,255))

#============================================
#Images
img=image.load('rocket.png')
img_size=100
#im=transform.scale(img,(img_size,img_size))
background=image.load('galaxy.jpg')
background=transform.scale(background,(window_width,window_height))

#============================================
#Values-Ставим значение
rect=img.get_rect()
rect.x=window_width//2
rect.y=window_height//2
speed =4

#============================================
#Кадры в секунды
FPS=60
clock=time.Clock()

#============================================
#Player
rocket=Player('rocket.png',window_width/2,window_height-110,7)

#Bullets
bullet=Bullet('bullet.png',rect.x,rect.y,speed)

#============================================
```

Рисунок 2. Скрипт для создания фонов и спрайтов.

```
random_x=randint(0,window_width-img_size)
ufo=Enemy('ufo.png',random_x,0,3)

#================================================
end=True
while end:
    #================================================
    #Drawning
    window.blit(background,(0,0))
    rocket.reset()
    rocket.update()
    ufo.reset()
    ufo.update()
    window.blit(text,(5,5))
    window.blit(text2,(5,30))
    display.update()
    window.fill((0,0,0))

    #================================================
    #Exit from game
    for i in event.get():
        if i.type==QUIT:
            quit()
        if i.type==KEYDOWN:
            if i.key==K_ESCAPE:
                quit()
    clock.tick(FPS)


    if i.type==KEYDOWN:
        if i.key==K_SPACE:
            bullet.reset()
            bullet.update()
while True:
    for i in event.get():
        if i.type==QUIT:
            quit()
        if i.type==KEYDOWN:
            if i.key==K_ESCAPE:
                quit()
```

Рисунок 3. Скрипты спавна противников, окончания игры и управления.

```python
from pygame import*
from random import*

#-----------------------
#Классы
class GameSprite(sprite.Sprite):
    def __init__(self,player_image,player_x,player_y,player_speed):
        super().__init__()
        self.image=transform.scale(image.load(player_image),(100,60))
        self.speed=player_speed
        self.rect=self.image.get_rect()
        self.rect.x=player_x
        self.rect.y=player_y
    def reset(self):
        window.blit(self.image, (self.rect.x, self.rect.y))
class Player(GameSprite):
    def __init__(self, player_image, player_x, player_y, player_speed):
        GameSprite.__init__(self,player_image,player_x,player_y,player_speed)
        self.image=transform.scale(image.load(player_image),(90,110))
    def update(self):
        keys=key.get_pressed()
        if keys[K_RIGHT] and self.rect.x<window_width-img_size:
            self.rect.x+=self.speed
        elif keys[K_LEFT] and self.rect.x>5:
            self.rect.x-=self.speed
class Enemy(GameSprite):
    def __init__(self, player_image, player_x, player_y, player_speed):
       GameSprite.__init__(self,player_image, player_x, player_y, player_speed)
    def update(self):
        self.rect.y+=self.speed
        if self.rect.y > window_height:
            self.rect.y=0
            self.rect.x=randint(0,window_width)
class Bullet(GameSprite):
    def __init__(self, player_image, player_x, player_y, player_speed):
        GameSprite.__init__(self,player_image, player_x, player_y, player_speed)
    def update(self):
        self.rect.y-=self.speed
        if self.rect.y<0:
            self.kill()
```

Рисунок 4. Скрипт передвижения, хитбоксы и спрайты (внутриигровые модели).

Заключение. В ходе нашей проделанной научной работы нам удалось добиться желаемого результата. У нас получилось создать игру на языке программирования Python, конечно, не без каких-либо проблем и недочётов. К такому недочёту можно отнести отсутствие звуков, что в дальнейшем мы постараемся исправить. Мы ожидали, что создание игры не будет какой-то тяжелой работой, но оказалось все совсем иначе. Мы справились с нашими задачами и поставленной целью и в дальнейшем постараемся доработать свою игру и выйти на еще более высокий уровень.

Список использованных источников:

1. Гуриков С.Р. Основы алгоритмизации и программирования на Python [Текст]: учебное пособие. – М.: Инфра-М, 2018, 343 с. ISBN 978-5-00091-487-8, ISBN 978-5-16-013349-2, ISBN 978-5-16-102278-8: https://library.atu.edu.kz/files/43474.pdf

2. Федоров Д.Ю. Программирование на языке высокого уровня Python [Текст]: учебное пособие // Д.Ю. Федоров, 2-е издание. – М: Юрайт, 2022, 210 с. ISBN 9785534146387
3. Руководство по языку программирования Python: Онлайн курс, https://metanit.com/python/tutorial/
4. Как программировать игры на Python с помощью библиотеки Pygame // wikiHow URL: https://ru.wikihow.com/программировать-игры-на-Python-с-помощью-библиотеки- Pygame