

**XX Региональная научно-практическая конференция школьников
«Мир вокруг нас»**

МАОУ «СОШ с.Ново-Кусково Асиновского района Томской области»

Разработка компьютерной игры SubX на языке Python

Автор: Шелепнёв Артём, 6 класс

Руководитель: Каширо Светлана Владимировна,
учитель физики и информатики

Ново-Кусково, 2023

ОГЛАВЛЕНИЕ

Введение	3
I. Глава Теоретические основы программирования на Python	4
II. Глава Практическое применение языка программирования Python для создания игры	5
Этапы работы над проектом	5
Создание элементов и персонажей игры в графическом редакторе	6
Элементы программного кода	8
Заключение	13
Список использованных источников и литературы	13

Введение

«Кто сам программирует свои компьютерные игры,
тот наслаждается дважды»

(Ж. Арсак)

Актуальность: В современном мире компьютер и информационные технологии играют все большую роль. Взрослые и дети много времени проводят возле экранов компьютера, осваивая и внедряя в свою жизнь приложения разного назначения: компьютерные игры, конструкторы меню, интернет магазины, чаты для общения и т.д. Но мало кто задумывается как созданы эти приложения. В основе разработки любого приложения лежит язык программирования, а разработкой занимается программист. Языки программирования имеют в своем арсенале много интересных возможностей, их изучение необходимо для реализации профессиональных наклонностей.

Проблема: я увлекаюсь языками программирования, первый язык, который я изучил Scratch. Далее мне захотелось изучать более сложные языки программирования, попробовал C++, C#, но их освоить у меня не получилось. Однажды в школе на ноутбуке я увидел Python и решил попробовать себя в его освоении. По мере изучения возник вопрос «А можно ли создать компьютерную игру на языке Python?».

Цель работы: разработка компьютерной игры SubX, при создании которой будет изучаться язык программирования Python.

Задачи:

- рассмотреть технологию создания игры на языке программирования;
- рассмотреть технологию геймдизайна;
- создать в графическом редакторе объекты компьютерной игры;
- прописать действия каждого объекта;
- написать программный код;

- тестирование игры и исправление ошибок;
- итоговый запуск игры.

Практическая значимость: если приобрести навыки работы с языками программирования, то можно самому разработать любое приложение, в том числе компьютерную игру, любой сложности.

I. Глава Теоретические основы программирования на языке Python

Python (в русском языке распространено название питон) — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. Язык обладает чётким и последовательным синтаксисом, продуманной модульностью и масштабируемостью, благодаря чему исходный код написанных на Python программ легко читаем.

Python — активно развивающийся язык программирования, новые версии с добавлением и изменением языковых свойств выходят примерно раз в два с половиной года. Он находит применение во множестве сфер человеческой деятельности.

Питон поддерживает практически все распространенные операционные системы.

Python относится к наиболее востребованным и популярным языкам программирования. Он достаточно прост, а потому изучение языка не займет слишком много времени.

Для создания программ часто необходимы дополнительные функции. Для этого существуют специальные библиотеки. Библиотеки могут использоваться для создания оконных приложений с кнопками, картинками и так далее. Существуют специальные библиотеки для создания игр. Некоторые из них встроены в Python, некоторые нужно скачивать отдельно.

На основании вышеизложенного можно сделать вывод, что Python - активно развивающийся и самый перспективный язык программирования, кото-

рый поддерживает почти все операционные системы. Он простой и удобный в использовании.

II. Глава Практическое применение языка программирования Python для создания игры

Этапы работы над проектом


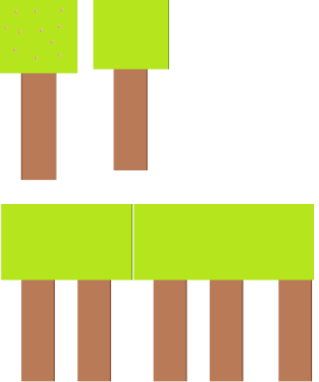
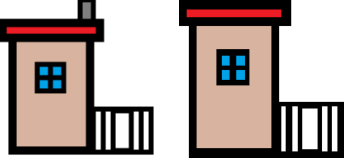

№ пп	Этап	Задача	Ресурсы	Сроки
1	Поисковый этап	<ul style="list-style-type: none"> – Изучение технологии создания компьютерной игры – Изучение технологии геймдизайна 	<ul style="list-style-type: none"> – Компьютер – Интернет-ресурсы – Программное обеспечение 	01.09.2022 – 20.10.2022
2	Реализация проекта	<ul style="list-style-type: none"> – Создание основных элементов и персонажей игры в графическом редакторе – Написание программного кода – Тестирование игры и исправление ошибок 	<ul style="list-style-type: none"> – Компьютер с выходом в Интернет – Программное обеспечение: Python, Paint 	20.10.2022 – 01.02.2023
3	Подведение итогов	<ul style="list-style-type: none"> – Самооценка – Построение планов на будущее для совершенствования игры – Совершенствование навыков в программировании и геймдизайне 	<ul style="list-style-type: none"> – Компьютер с выходом в Интернет – Программное обеспечение: Python, Paint 	01.02.2023 – 01.03.2023

Создание элементов и персонажей игры в графическом редакторе

Первыми шагами к написанию игры было знакомство со всей нужной информацией, а именно с внутренней составляющей компьютерных игр и языком программирования Python. Всю необходимую информацию я искал в интернете и сразу же пробовал применить на практике. Первое применение языка Python было для создания консольной программы Калькулятор. Для создания более сложных операций я стал знакомиться с библиотеками Python Pygame и Pyinstaller. В ходе изучения появилась идея создания игры CubX. Я решил, что главным героем будет куб и игровая среда по возможности в виде кубов. Далее нужно было продумать игровую среду, главного героя, его действия в игре.

Все объекты игры нарисованы в программе Paint. Внешний вид героя и его действия прописаны ниже в таблице.

	Главный герой гуляет по локации влево, вправо, борется с врагами. При движении у него поворачиваются глаза влево и вправо. Чтобы герой перемещался, нужно нажимать клавиши D – для движения в правую сторону, клавишу A – для движения в левую сторону
	Второй герой – житель, по внешности похож на главного героя, отличается костюмом, ходит по миру и живет своей жизнью.
	Третий герой – небольшая свинка с огромными глазами, она также ходит и живет своей жизнью.
	В игре есть два вида камней: обычный и заросший камень.
	Растительность: трава, цветы. Появляются в разных частях локации и с каждым входом в игру они меняют свое положение. Трава, которая жел-

	<p>тая – засохшая. Пышная трава с цветочками и обычная трава.</p>
	<p>В игре есть грибы, два вида мухоморов и обычный гриб. В локации может появляться всего один мухомор.</p>
	<p>В игре три вида деревьев. Уникальное дерево с розовыми цветочками, находится в центре локации, является указателем того, что это центр. Обычные деревья. Граничные деревья – группа деревьев по краям локации, за предел этого леса выйти нельзя.</p>
	<p>Домики</p>
	<p>Солнце и Луна. В игре есть смена дня и ночи, Солнце встает в левой части игрового окна и садится на правом краю игрового поля. Когда Солнце выйдет за предел окна оно незаметно возвращается на левую сторону игрового поля и меняет свою текстуру на Луну, становится темно. Когда Луна достигнет границы правой стороны игрового поля окна, она меняет текстуру на Солнце, вокруг становится светло.</p>
	<p>Оружие. Всегда находится рядом с главным героем, предназначено для борьбы с врагами. Оружием можно управлять управляющими клавишами, при управлении оружием появляется «световой шар» в том направлении, которую клавишу нажали</p>

Элементы программного кода

После прорисовки героев и объектов игры, нужно их привести в движение. Для этого написана программа. Всю программу я разбил условно на блоки. Каждый блок программы отвечает за какое-либо действие: движение главного героя, за главное меню, за движение глаз главного героя и т.д. Эти блоки представлены ниже.

Часть программы отвечает для управления главным меню, позволяет выбрать играть или выйти из игры. С помощью этого цикла настраивается вся игра.

```
if key_menu == True:
    key = pygame.key.get_pressed()
    if key[pygame.K_2]:
        quit()
    if key[pygame.K_1]:
        screen.fill((0, 0, 0))
        pygame.display.update()
        key_menu = False
        key_game = True
        Menu = False
        Game = True

    sound1.stop()
    sound2.play(loops= -1)
    sound2.set_volume(0.3)

    while Game:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                quit()
```

Блок программы позволяет прорисовать объекты в главном меню. Screen – переменная, в котором хранится игровое окно. Blit - нарисовать объект. В скобках тоже переменные, в которых хранятся картинки или текст с определенными координатами). Image1 – это голубой фон главного меню.

```
screen.blit(image1, (0,0))
screen.blit(version, (450,470))
screen.blit(play, (20,220))
screen.blit(quit1, (70,260))
screen.blit(cub, (200,20))
screen.blit(x, (257,20))
screen.blit(details, (0,470))
```


Блок отвечает за отрисовку объектов в игре. Позволяет менять координаты объекта на поле.

```
screen.blit(sun, (xsun, 20))
tree()
stone()
spawn_grass()
mushroom()
house()
screen.blit(citizen, (citizenX,285))
screen.blit(pig, (pigX, 305))
screen.blit(player, (playerX,playerY))
screen.blit(weapon1, (playerX+50, playerY-30))
screen.blit(moon, (0,0))
```

Этот блок программы отвечает за движение жителя и направление его глаз. Этот блок организован с помощью условия. Переменная citizenB принимает каждую секунду случайно число от 0 до 50. Когда выпадает 0 то условие первое отключается, включается второе условие, при этом житель меняет свою текстуру (меняется направление взгляда) и меняет направление движения.

```
if citizenB == True:
    citizenX += 1
    citizenB1 = random.randint(0,50)
    citizen = pygame.image.load("assets/10/Citizen2.png")
    if citizenB1 == 0:
        citizenB = False
elif citizenB == False:
    citizenX -= 1
    citizenB1 = random.randint(0,50)
    citizen = pygame.image.load("assets/10/Citizen1.png")
    if citizenB1 == 0:
        citizenB = True
```

Этот блок отвечает за изменение взгляда главного героя. Если нажать клавишу A, то взгляд направлен в левую сторону, если нажали клавишу D, то взгляд направлен вправо сторону, иначе смотрит прямо.

```
if key[pygame.K_a]:
    player = pygame.image.load("assets/3/player2.png")
elif key[pygame.K_d]:
    player = pygame.image.load("assets/3/player3.png")
else:
    player = pygame.image.load("assets/3/player1.png")
```

Этот блок программы отвечает за расположение различных объектов: деревьев, травы, домиков и т.д. Def – функция, она начинает выполняться, когда ее вызывают. В этом блоке объекты разделены на группы: трава, грибы, деревья, камни, домики. Они начинают прорисовываться, когда эта функция вызывается.

```
def spawn_grass():
    screen.blit(grass1, (grass11, 300))
    screen.blit(grass2, (grass12, 298))
    screen.blit(grass3, (grass13, 301))
    screen.blit(flower1, (grass14, 287))
    screen.blit(grass1, (grass15, 300))
    screen.blit(grass2, (grass16, 298))
    screen.blit(grass3, (grass17, 301))
    screen.blit(flower1, (grass18, 287))
    screen.blit(grass1, (grass19, 300))
    screen.blit(grass2, (grass20, 298))
    screen.blit(grass3, (grass21, 301))
    screen.blit(flower1, (grass22, 287))
    screen.blit(grass1, (grass23, 300))
    screen.blit(grass2, (grass24, 298))
    screen.blit(grass3, (grass25, 301))
    screen.blit(grass1, (grass27, 300))
    screen.blit(grass2, (grass28, 298))
    screen.blit(grass3, (grass29, 301))
    screen.blit(flower1, (grass30, 287))
    screen.blit(grass1, (grass31, 300))
    screen.blit(grass2, (grass32, 298))
    screen.blit(grass1, (grass33, 300))
    screen.blit(grass2, (grass34, 298))
    screen.blit(grass3, (grass35, 301))

def mushroom():
    screen.blit(mushroom2, (mushroom11, 279))
    screen.blit(mushroom2, (mushroom12, 279))
    screen.blit(mushroom1, (mushroom13, 306))
    screen.blit(mushroom1, (mushroom14, 306))
    screen.blit(mushroom1, (mushroom15, 306))
    screen.blit(mushroom1, (mushroom16, 306))
    screen.blit(mushroom1, (mushroom17, 306))
    screen.blit(mushroom1, (mushroom18, 306))
    screen.blit(mushroom1, (mushroom19, 306))
    screen.blit(mushroom1, (mushroom20, 306))

def tree():
    screen.blit(tree2, (treespawn, 55))
    screen.blit(tree1, (tree_1, 55))
    screen.blit(tree1, (tree_2, 55))
    screen.blit(tree1, (tree_3, 55))
    screen.blit(tree1, (tree_4, 55))
    screen.blit(tree1, (tree_5, 55))
    screen.blit(tree1, (tree_6, 55))
    screen.blit(tree3, (tree31, 55))
    screen.blit(tree4, (tree32, 55))

def stone():
    screen.blit(stone1, (stonel1, 279))
    screen.blit(stone2, (stonel2, 294))
    screen.blit(stone1, (stonel2, 279))
    screen.blit(stone2, (stonel6, 294))
    screen.blit(stone1, (stonel5, 279))
    screen.blit(stone2, (stonel1, 294))
    screen.blit(stone1, (stonel3, 279))
    screen.blit(stone2, (stonel4, 294))

def house():
    screen.blit(house1, (house_1, 185))
    screen.blit(house1, (house_2, 185))
    screen.blit(house1, (house_3, 185))
```

Эта часть программы отвечает за движение главного героя в правую сторону

```
if xcoordinat < 310:
    key = pygame.key.get_pressed()
    if key[pygame.K_d]:
        stonell1 -= 1
        stonell2 -= 1
        stonell3 -= 1
        stonell4 -= 1
        stonell5 -= 1
        stonell6 -= 1

        mushroom11 -= 1
        mushroom12 -= 1
        mushroom13 -= 1
        mushroom14 -= 1
        mushroom15 -= 1
        mushroom16 -= 1
        mushroom17 -= 1
        mushroom18 -= 1
        mushroom19 -= 1
        mushroom20 -= 1

        grass11 -= 1
        grass12 -= 1
        grass13 -= 1
        grass14 -= 1
        grass15 -= 1
        grass16 -= 1
        grass17 -= 1
        grass18 -= 1
        grass19 -= 1
        grass20 -= 1
        grass21 -= 1
        grass22 -= 1
        grass23 -= 1
        grass24 -= 1
        grass25 -= 1
        grass26 -= 1
        grass27 -= 1
        grass28 -= 1
        grass29 -= 1
        grass30 -= 1
        grass31 -= 1
        grass32 -= 1
        grass33 -= 1
        grass34 -= 1
        grass35 -= 1

        tree_1 -= 1
        tree_2 -= 1
        tree_3 -= 1
        tree_4 -= 1
        tree_5 -= 1
        tree_6 -= 1
        treespawn -= 1
        tree31 -= 1
        tree32 -= 1

        house_1 -= 1
        house_2 -= 1
        house_3 -= 1

        citizenX -= 1

        pigX -= 1

    xcoordinat += 0.105
```

Эта часть программы отвечает за движение главного героя в левую сторону

```
if key_game == True:
    key = pygame.key.get_pressed()
    if xcoordinat > -310:
        if key[pygame.K_a]:
            stonel1 += 1
            stonel2 += 1
            stonel3 += 1
            stonel4 += 1
            stonel5 += 1
            stonel6 += 1

            mushroom11 += 1
            mushroom12 += 1
            mushroom13 += 1
            mushroom14 += 1
            mushroom15 += 1
            mushroom16 += 1
            mushroom17 += 1
            mushroom18 += 1
            mushroom19 += 1
            mushroom20 += 1

            grass11 += 1
            grass12 += 1
            grass13 += 1
            grass14 += 1
            grass15 += 1
            grass16 += 1
            grass17 += 1
            grass18 += 1
            grass19 += 1
            grass20 += 1
            grass21 += 1
            grass22 += 1
            grass23 += 1
            grass24 += 1
            grass25 += 1
            grass26 += 1
            grass27 += 1
            grass28 += 1
            grass29 += 1
            grass30 += 1
            grass31 += 1
            grass32 += 1
            grass33 += 1
            grass34 += 1
            grass35 += 1

            tree_1 += 1
            tree_2 += 1
            tree_3 += 1
            tree_4 += 1
            tree_5 += 1
            tree_6 += 1
            treespawn += 1
            tree31 += 1
            tree32 += 1

            house_1 += 1
            house_2 += 1
            house_3 += 1

            citizenX += 1

            pigX += 1

            xcoordinat -= 0.105
```

Этот блок отвечает за управление оружием. Если нажата клавиша вправо, то и световой шар от оружия летит вправо. Другие клавиши работают по аналогии.

```
if key[pygame.K_RIGHT]:
    screen.blit(cartridges1, (cartridges1X, cartridges1Y))
    cartridges1X += 1
elif key[pygame.K_LEFT]:
    screen.blit(cartridges1, (cartridges1X, cartridges1Y))
    cartridges1X -= 1
elif key[pygame.K_UP]:
    screen.blit(cartridges1, (cartridges1X, cartridges1Y))
    cartridges1Y -= 1
elif key[pygame.K_DOWN]:
    screen.blit(cartridges1, (cartridges1X, cartridges1Y))
    cartridges1Y += 1
else:
    cartridges1X = 260
    cartridges1Y = 300
    key_game = True
```

Заключение

В заключении хочется сказать, что цель проекта достигнута. В процессе работы я изучил технологии создания компьютерных игр. Результатом проделанной мной работы стала компьютерная игра SubX.

Практическая значимость проделанной мной работы для меня огромна. Изучая языки программирования, в том числе и Python, я буду совершенствовать свои навыки, а в будущем смогу стать хорошим программистом.

Список использованных источников и литературы:

1. Арсак Ж. Программирование игр и головоломок. - Ж. Арсак. – Наука, 1990. – 76с.
2. Tech: Python – краткий обзор языка и его назначения. – <https://techrocks.ru>
3. Основы геймдизайна: этапы разработки видеоигр. - <https://genapilot.ru/osnovy-gejmdizajna-jetapy-razrabotki-videoigr>
4. Программирование для детей на языке Python. – Москва: Издательство АСТ – 2017. – 95 с.
5. Трофимов В. Программирование игр, создание с нуля. -В. Трофимов. – 2020
6. Яндекс Дзен: Дзен питона. – Режим доступа: <https://zen.yandex.ru>