МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «ВИТЕБСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ П.М. МАШЕРОВА»

Факультет математики и информационных технологий

Кафедра информатики и информационных технологий

КУРСОВОЙ ПРОЕКТ

по дисциплине «Языки программирования»

РАЗРАБОТКА ИГРЫ «КОСМИЧЕСКИЙ ФЛОТ»

НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ PYGAME

Борисенко Анастасия Владимировна,

курс 2, группа 24

Руководитель:

Шпаков Сергей Андреевич,

старший преподаватель кафедры информатики и информационных технологий

Витебск, 2020

**Содержание**

[Введение 3](#_Toc59364104)

[1. Разработка компьютерных игр 6](#_Toc59364105)

[1.1. Варианты разработки компьютерных игр 6](#_Toc59364106)

[1.2. PyGame 6](#_Toc59364107)

[2. Реализация приложения 9](#_Toc59364108)

[3. Интерфейс приложения 16](#_Toc59364109)

[Заключение 17](#_Toc59364110)

[Список использованных источников 19](#_Toc59364111)

[Приложение 21](#_Toc59364112)

# **Введение**

Компьютерные игры давно вошли в жизнь современного человека и заняли практически первое место среди всех видов отдыха. И это можно объяснить: виртуальная реальность имеет свойство притягивать своими неограниченными возможностями, а разработчики не перестают радовать любителей игр новыми продуктами. [3]

Многие считают, что современные компьютерные игры не представляют собой абсолютно никакой пользы, оказывая негативное влияние как на наше сознание, так и на физическое состояние. Однако, следует отметить, что компьютерные игры обладают рядом преимуществ и положительных для здоровья свойств:

* повышают самооценку, делают человека более уверенным в себе и целеустремленным. По словам ученого из Трентского университета Ноттингема Марка Гриффитса: «Когда вы победили в игре, вы чувствуете себя хорошо - и это поднимает вашу самооценку»;
* геймеры благодаря играм повышают уровень владения английским языком. World of Warcraft, Call of Duty и другие подобные экшены и шутеры способствуют популяризации английского языка. Например, одно из исследований в Швеции показало, что молодежь, играющая в компьютерные игры, около 10-12 часов в неделю общаются онлайн на английском языке (письменно или устно). Причем, в играх часто используется разговорный язык с выражениями и словами, которых нет ни в общей школьной, ни в университетской программах;
* компьютерные игры – источник новых знаний. Опрос создателей популярной игры WOT показал, что игроки стали больше интересоваться военной историей, в том числе, танкостроением, благодаря чему стали чаще посещать музеи и тематические выставки. И, что важно, не в одиночку, а вместе со своими супругами и детьми.
* игры улучшают координацию и память, развивают логическое мышление. Лучшим подтверждением этому служит тот факт, что некоторые школы в Европе используют компьютерные игры (к примеру, Minecraft), чтобы развивать у детей моторику и логическое мышление, а многие хирурги предпочитают играть в шутеры, чтобы тренировать координацию движений;
* шутеры и экшены – отличный способ выплеснуть негатив и агрессию. Исследования доказывают, что игры помогают больным отвлекаться от боли, а при накоплении агрессии и негатива в реальной жизни – это отличная возможность выплеснуть ее виртуально;
* компьютерные игры развивают творческий потенциал и фантазию. Выполняя различные задания, находя возможные способы решения поставленных задач, человек развивает свое творческое мышление и умение мыслить вне рамок.

Космический флот — жанр компьютерных игр, воспроизводящих с различной степенью достоверности управление космическим кораблём. Действие таких игр происходит, как правило, в космосе, главным элементом которых являются космические сражения.

Целью данной работы является создание игры «Космический флот» на языке программирования Python с использованием библиотеки PyGame.

Для достижения поставленной цели необходимо решить следующие задачи:

* выработать умение логически строить и последовательно излагать материал по выбранной теме;
* освоить и овладеть особенностями создания компьютерных игр при помощи с библиотеки PyGame;
* разработать модуль дисплея, который работает с окном и экраном дисплея, а также модуль клавиш для работы с клавиатурой.
* изучить и проработать литературу, раскрывающую основы языка программирования Python а также – средства, главные принципы, методы программирования на языке Python, описания стандартов кодирования на данном языке программирования; и принцип программирования с использованием библиотеки PyGame.
* получить навыки разработки компьютерных игр на языке программирования Python с использованием библиотеки PyGame.

# **Разработка компьютерных игр**

## **Варианты разработки компьютерных игр**

Создание игры — это продолжительный и трудоёмкий процесс, состоящий из самых разнообразных этапов, включающий в себя как технические, так и творческие моменты.

Перед созданием игры программисту нужно решить, по какому пути её разработки он хочет пойти.

Первый вариант – это написание исходного кода с нуля. Такой путь удобен, так как дает свободу действий и практически неограниченный функционал, который можно реализовать. Среди минусов необходимо выделить трудоемкость и растянутость во времени, а также необходимость тщательно тестировать готовый продукт на предмет ошибок и недоработок.

Второй вариант – это использование фреймворков. При создании игр с их помощью у вас уже будет готовая основа, которую необходимо будет заполнить нужными компонентами. В PyGame есть фреймворк. Однако его сложно назвать "мощным инструментом". По своему объему и функционалу - скорее библиотека. В любом случае, фреймворк является более мощным по-сравнению с библиотекой, он накладывает свою специфику на особенности программирования и сферу использования продукта. С точки зрения специфики.

При разработке приложения «Космический флот» нами был выбран первый вариант.

## **PyGame**

PyGame — набор модулей (библиотек) языка программирования Python,предназначенный для написания компьютерных игр и мультимедиаприложений. PyGame базируется на мультимедийной библиотеке SDL.

Событие — это то, как PyGame сообщает о том, что что-то случилось за пределами кода программы. События создаются, например, при нажатии клавиш клавиатуры, мыши и размещаются в очереди, дожидаясь обработки.

PyGame отслеживает все сообщения о событиях через очередь событий. Процедуры в этом модуле помогают управлять этой очередью событий. Входная очередь сильно зависит от модуля отображения (display) PyGame. Если дисплей не был инициализирован и видеорежим не установлен, очередь событий не будет работать.

Изначально PyGame был написан Питом Шиннерсом (Pete Shinners). Начиная примерно с 2004/2005 года поддерживается и развивается сообществом свободного программного обеспечения. [4]

PyGame – это библиотека модулей для языка Python, созданная для разработки 2D игр. Также PyGame могут называть фреймворком. В программировании понятия "библиотека" и "фреймворк" несколько разные. Но когда дело касается классификации конкретного инструмента, не все так однозначно.

Также существует понятие "игрового движка" как программной среды для разработки игр. По своему назначению PyGame можно считать игровым движком. В то же время, с точки зрения классификации программного обеспечения, PyGame является API для Питона к API библиотеки SDL.

API – это интерфейс (в основном набор функций и классов) для прикладного (часто более высокоуровневого) программирования, который предоставляет, например, та или иная библиотека.

SDL – это библиотека, которая работает с мультимедийными устройствами компьютера.

Существует множество способов доступа к очереди событий. Просто проверять существование событий, захватывать их непосредственно из стека.

Функция get в модуле PyGame.event возвращает последнее событие, ожидающее в очереди и удаляет его из очереди.

Данная библиотека нам позволяет перемещать изображение на экране с клавиатуры, назначая клавиши для перемещений: вверх, вниз, влево, вправо.

Проверить, является ли полученное событие нажатием на клавиши со стрелками: event.type == KEYDOWN.[9]. Основные модули пакета PyGame, используемые нами в работе перечислены в Таблице 1.1.

Таблица 1.1

|  |  |
| --- | --- |
| **Модуль** | **Назначение** |
| PyGame.display | Доступ к дисплею |
| PyGame.event | Управление внешними событиями |
| PyGame.font | Использует системные шрифты |
| PyGame.image | Загружает и сохраняет изображение |

# **Реализация приложения**

Для создания приложения необходимо написания компилируемого кода с использованием библиотеки PyGame на языке Python.

Разбираем частично код, написанный нами:

Стандартная инициализация PyGame.init() — это команда, которая запускает PyGame(Листинг 2.1.)

Листинг 2.1. Инициализация PyGame

|  |
| --- |
| PyGame.init() |

Теперь переходим к созданию окна приложения (Листинг 2.2.), фиксируем фон (Листинг 2.3.), для изменения иконки и названия мы воспользуемся следующими команды и фиксируем их (Листинг 2.4.).

Листинг 2.2. Создание окна приложения

|  |
| --- |
| screen = PyGame.display.set\_mode((800, 600)) |

Листинг 2.3. Фиксируем фон

|  |
| --- |
| background = PyGame.image.load('background.png') |

Листинг 2.4. Задаем название и иконку

|  |
| --- |
| PyGame.display.set\_caption("Космический флот")  icon = PyGame.image.load('ufo.png')  PyGame.display.set\_icon(icon) |

Для фиксации внешнего вида игрока будем использовать модуль PyGame.image, а так же укажем начальную точку.( Листинг 2.5.) [10]

Листинг 2.5. Фиксация внешнего вида игрока

|  |
| --- |
| playerImg = PyGame.image.load('player.png')  playerX = 370  playerY = 480  playerX\_change = 0 |

Создали окно игры 800х600 (Рисунок 2.1.) и наш космический корабль в начальной позиции находится на 370 пикселей (Листинг 2.9.).



Рисунок 2.1. Созданное игровое окно

Для перемещения изображения в нужном направлении (как указано на картинке) мы будем добавлять пять пикселей к уже существующим 370 пикселей, и изображение будет двигаться вправо, а если мы добавляем пять пикселей непрерывно, то и изображение будет продолжать двигаться к правой стороне. Поэтому добавляем эти действия в цикл while (Листинг 2.11.).

Укажем внешний вид и позиции космического захватчика (Рисунок 2.2.) (Листинг 2.6.1) [10]

Листинг 2.6.1. Враг

|  |
| --- |
| enemyImg = []  enemyX = []  enemyY = []  enemyX\_change = []  enemyY\_change = []  num\_of\_enemies = 6 #количество |

Добавим переменную - число врагов и цикл for (шесть раз) (Листинг 2.6.2.), все эти значения добавим в массив изображений врага. А цикл while (Листинг 2.13.) - траектория врага, двигающегося влево или вправо и вниз.

Листинг 2.6.2. Создание нескольких врагов

|  |
| --- |
| for i in range(num\_of\_enemies):  enemyImg.append(PyGame.image.load('enemy.png'))  enemyX.append(random.randint(0, 736))  enemyY.append(random.randint(50, 150))  enemyX\_change.append(4)  enemyY\_change.append(40) |

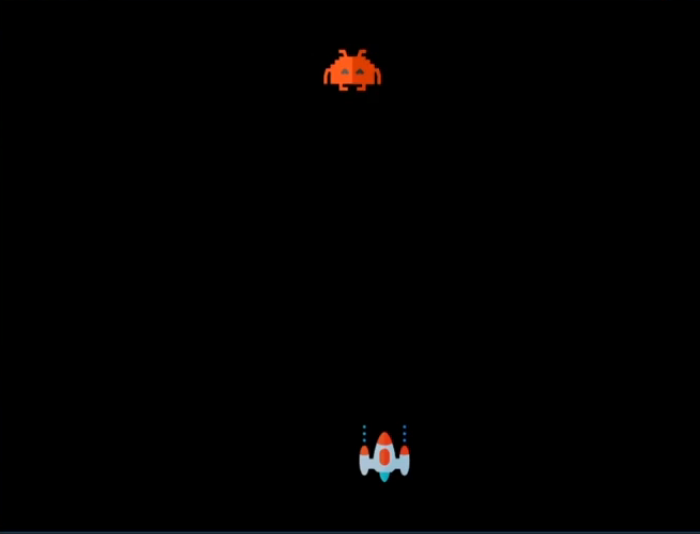


Рисунок 2.2. Появление космического захватчика

Теперь переходим к начальную этапу реализации стрельбы. Так же вводим понятия (Ready - вы не видите пулю на экране, Fire - пуля в данный момент в движении). [2]

Даем вид, как это делали для игрока и врага. ( Листинг 2.7. ) [10]

Листинг 2.7. Пуля

|  |
| --- |
| bulletImg = PyGame.image.load('bullet.png')  bulletX = 0  bulletY = 480  bulletX\_change = 0  bulletY\_change = 10  bullet\_state = "ready" |

Теперь объявим переменную score: до игрового цикла переменная с начальным значением 0, к которой будет добавляться +1 при каждом уничтожении.  (Листинг 2.9.) Так же задаем шрифт, воспользуемся модулем PyGame.font, а text — строка, которую нужно отображать. (Листинг 2.8.)

Листинг 2.8. Счет[7]

|  |
| --- |
| score\_value = 0  font = PyGame.font.Font('freesansbold.ttf', 32)  textX = 10  testY = 10  over\_font = PyGame.font.Font('freesansbold.ttf', 64) |



Рисунок 2.3. Счет

Листинг 2.9. Координаты и цвет

|  |
| --- |
| def show\_ (x, y):  score = font.render("Счет : " + str(score\_value), True, (255, 255, 255))  screen.blit(score, (x, y))  def game\_over\_text():  over\_text = over\_font.render("Конец игры", True, (255, 255, 255))  screen.blit(over\_text, (200, 250))  def player(x, y):  screen.blit(playerImg, (x, y))  def enemy(x, y, i):  screen.blit(enemyImg[i], (x, y))  def fire\_bullet(x, y):  global bullet\_state  bullet\_state = "fire"  screen.blit(bulletImg, (x + 16, y + 10)) |

В следующем элементе происходит обнаружения столкновения пули и врага для его нахождения использовалась формула 2.1. (Листинг 2.14.)

Формула 2.1

[5]

Листинг 2.14.1.Столкновение

|  |
| --- |
| def isCollision(enemyX, enemyY, bulletX, bulletY):  distance = math.sqrt(math.pow(enemyX - bulletX, 2) + (math.pow(enemyY - bulletY, 2)))  if distance < 27:  return True  else:  return False |

Листинг 2.10. Работа с фоном[9]

|  |
| --- |
| running = True  while running:  # RGB = Red, Green, Blue (черный)  screen.fill((0, 0, 0))  # изображение  screen.blit(background, (0, 0))  for event in PyGame.event.get():  if event.type == PyGame.QUIT:  running = False |

Реализация управления движения игрока с помощью стрелочек на клавиатуре (Листинг 2.11.) Для этого пригодиться модуль PyGame.key. Этот модуль содержит функции для работы с клавиатурой. Очередь событий получает события PyGame.KEYDOWN и PyGame.KEYUP при нажатии и отпускании клавиш клавиатуры.

Листинг 2.11. Управление игроком с помощью клавиш

|  |
| --- |
| if event.type == PyGame.KEYDOWN:  if event.key == PyGame.K\_LEFT:  playerX\_change = -5  if event.key == PyGame.K\_RIGHT:  playerX\_change = 5  if event.key == PyGame.K\_SPACE:  if bullet\_state is "ready":  bulletX = playerX  fire\_bullet(bulletX, bulletY)  if event.type == PyGame.KEYUP:  if event.key == PyGame.K\_LEFT or event.key == PyGame.K\_RIGHT:  playerX\_change = 0 |

Листинг 2.12. Добавление границ в игру

|  |
| --- |
| playerX += playerX\_change  if playerX <= 0:  playerX = 0  elif playerX >= 736:  playerX = 736 |

Листинг 2.13. Механика движения вражеского космического захватчика[11]

|  |
| --- |
| for i in range(num\_of\_enemies):  if enemyY[i] > 440:  for j in range(num\_of\_enemies):  enemyY[j] = 2000  game\_over\_text()  break  enemyX[i] += enemyX\_change[i]  if enemyX[i] <= 0:  enemyX\_change[i] = 3  enemyY[i] += enemyY\_change[i]  elif enemyX[i] >= 736:  enemyX\_change[i] = -3  enemyY[i] += enemyY\_change[i] |

Листинг 2.14.2. Обнаружение столкновений

|  |
| --- |
| collision = isCollision(enemyX[i], enemyY[i], bulletX, bulletY)  if collision:  bulletY = 480  bullet\_state = "ready"  score\_value += 1  enemyX[i] = random.randint(0, 736)  enemyY[i] = random.randint(50, 150)  enemy(enemyX[i], enemyY[i], i) |

Листинг 2.15. Стрельба несколькими выстрелами по космическим захватчикам.

|  |
| --- |
| if bulletY <= 0:  bulletY = 480  bullet\_state = "ready"  if bullet\_state is "fire":  fire\_bullet(bulletX, bulletY)  bulletY -= bulletY\_change |

Листинг 2.16.Отображение результата игры

|  |
| --- |
| player(playerX, playerY)  show\_score(textX, testY)  PyGame.display.update() |

# **Интерфейс приложения**

В разделе представлены результаты выполнения основных функциональных частей приложения «Игра «Космический флот»». Описание результатов реализации приводится вместе с описанием написания приложения.

При запуске приложения открывается окно игры. Окно содержит космических врагов и самого игрока. Управления производиться с помощью стрелочных клавиш и пробела (Рисунок 3.1.).

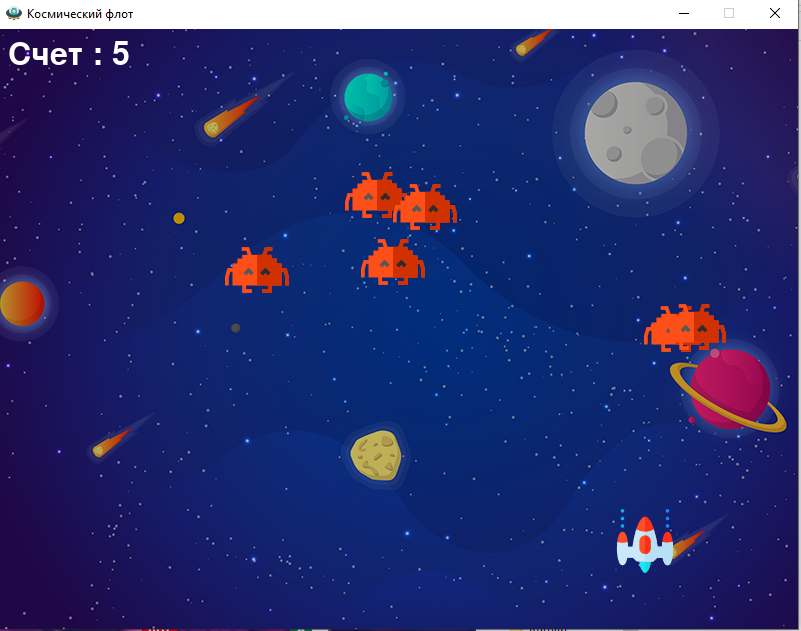


Рисунок 3.1. – Окно игры «Космический флот» в процессе

В случае проигрыша выводиться текст "Конец игры" и исчезают все враги с поля.( Рисунок 3.2.)

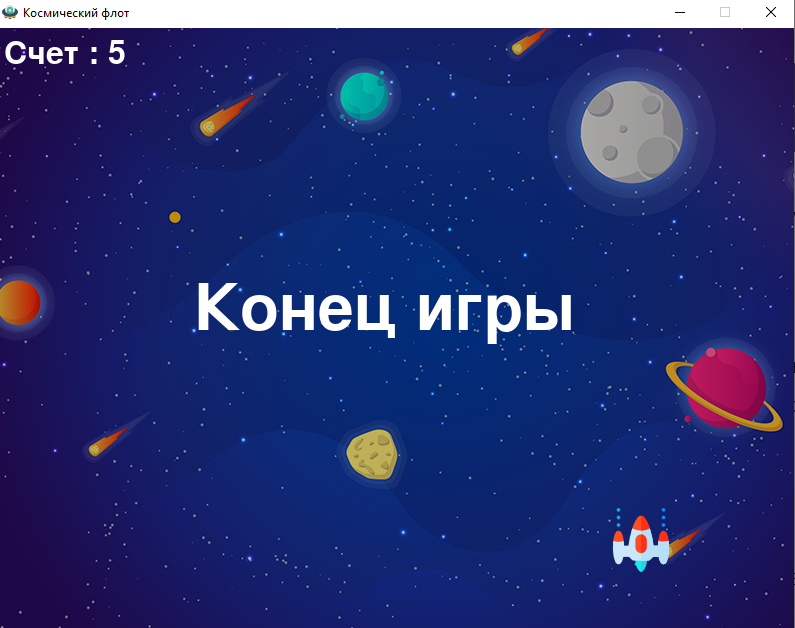


Рисунок 3.2. – Окно игры «Космический флот» конец игры

# **Заключение**

В данном курсовом проекте был рассмотрен один из множества способов создания простой 2D игры, а также основные принципы работы с изображениями в наборе модулей «PyGame» для языка программирования Python.

В мире существует множество платформ и утилит, упрощающих разработку компьютерных игр. Но, к сожалению, многие из них могут по-разному работать с разными операционными системами. Например, игра, написанная с использованием технологии Microsoft XNA, не будет работать на операционной системе класса Unix. Именно поэтому в качестве основной платформы для разработки компьютерной игры был выбран Python. Для расширения стандартных возможностей языка был использован набор библиотек и модулей для разработки мультимедиа приложений и компьютерных игр – PyGame.

Разработка компьютерной игры «Космический флот», созданной на языке программирования Python с использованием библиотеки PyGame, превратилась в увлекательное занятие, не требующее глубоких знаний технологий обработки изображения. Ведь Python в купе с Pyagame образует довольно приятную и удобную среду разработки, которая бесспорно подойдёт каждому независимому разработчику компьютерных игр. Несмотря на то, что Python является не самым распространённым языком программирования (в плане создания игр), это не мешает людям создавать исключительные шедевры игровой индустрии именно на этой платформе.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Синтаксис языка Python достаточно прост и интуитивно понятен, что понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки вебприложений до машинного обучения.

Научившись программировать на языке Python, начинающий разработчик получит мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит потом с лёгкостью выучить любой другой язык программирования.

Проанализировав предметную область, можно сказать, что разработка компьютерной игры «Космический флот» на языке Python актуальна. Цель работы достигнута, так как разработанное приложение соответствует поставленной цели.

# **Список использованных источников**

1. Объектно-ориентированный Python — Введение [Электронный ресурс] Режим доступа: <https://coderlessons.com/tutorials/python-technologies/izuchite-obektno-orientirovannyi-python/obektno-orientirovannyi-python-kratkoe-rukovodstvo> Дата доступа 8.11.2020
2. Стрелялка с PyGame №14: Игра закончена [Электронный ресурс] Режим доступа: https://pythonru.com/primery/streljalka-s-PyGame-14-igra-zakonchena Дата доступа 8.11.2020
3. Компьютерная игра [Электронный ресурс] Режим доступа: [https://ru.wikipedia.org/wiki/Компьютерная\_игра](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0) Дата доступа 8.11.2020
4. PyGame [Электронный ресурс] Режим доступа: [https://ru.wikipedia.org/wiki/PyGame#:~:text=PyGame%20–%20это%20библиотека%20модулей%20для,и%20"фреймворк"%20несколько%20разные.&text=Также%20существует%20понятие%20"игрового%20движка,программной%20среды%20для%20разработки%20игр](https://ru.wikipedia.org/wiki/Pygame#:~:text=Pygame%20%E2%80%93%20%D1%8D%D1%82%D0%BE%20%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0%20%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D0%B5%D0%B9%20%D0%B4%D0%BB%D1%8F,%D0%B8). Дата доступа 8.11.2020
5. Distance between two points and the midpoint [Электронный ресурс] Режим доступа: <https://www.mathplanet.com/education/algebra-2/conic-sections/distance-between-two-points-and-the-midpoint> Дата доступа 8.11.2020
6. Стрелялка с PyGame №4: графика [Электронный ресурс] Режим доступа: [https://pythonru.com/primery/streljalka-s-PyGame-4-grafika](https://pythonru.com/primery/streljalka-s-pygame-4-grafika) Дата доступа 9.11.2020
7. Стрелялка с PyGame №7: счет и рендеринг текста [Электронный ресурс] Режим доступа: [https://pythonru.com/primery/streljalka-s-PyGame-7-schet-i-rendering-teksta](https://pythonru.com/primery/streljalka-s-pygame-7-schet-i-rendering-teksta) Дата доступа 9.11.2020
8. Создание игр на Python 3 и PyGame: Часть 5 [Электронный ресурс] Режим доступа: https://habr.com/ru/post/347286/ Дата доступа 9.11.2020
9. Python - программирование игр - PyGame [Электронный ресурс] Режим доступа: [https://web-start.top/ru/progru/pythonru/PyGameru](https://web-start.top/ru/progru/pythonru/pygameru) Дата доступа10.11.2020
10. Создание игрока и его передвижение [Электронный ресурс] Режим доступа: [https://itproger.com/course/PyGame/2](https://itproger.com/course/pygame/2) Дата доступа 10.11.2020
11. PyGame: учебник по программировани [Электронный ресурс] Режим доступа: [https://www.codeflow.site/ru/article/PyGame-a-primer](https://www.codeflow.site/ru/article/pygame-a-primer) Дата доступа 10.11.2020

# **Приложение**

Содержание электронного носителя

На электронном носителе расположены следующие файлы и директории:

* Файл «Kurs.Rab.docx» с документацией к курсовому проекту.
* Архив «Kur.rab.zip», содержащий проект курсовой работы.
* Файл «Презентация.pptx», содержащий презентацию к курсовому проекту.