

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Повышение квалификации и переподготовка

Кафедра проектирования информационных компьютерных систем

Дисциплина "Современные технологии проектирования информационных систем"

К защите допустить:
Руководитель курсовой работы
старший преподаватель
кафедры
_____ А.В.Михалькевич
10.07.2025

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе
на тему

Разработка игры "Новый мир"
БГУИР КР 1-39 03 02 № 164 ПЗ

Студент

(подпись студента)

И.И.Иванов

Курсовая работа
представлена на проверку
10.07.2025

(подпись студента)

Минск 2025

Реферат

БГУИР КР 1-39 03 02 № 164 ПЗ, гр. 90421

И.И.Иванов, Разработка игры "Новый мир", Минск: БГУИР - 2025.

Пояснительная записка 92309 с., 1 рис., 0 табл.

Ключевые слова: ооп, современные технологии, проектирование систем

Предмет Современные технологии проектирования информационных систем,
А.В.Михалькевич

-
-

Содержание

[Введение](#)

1 [Описание проекта](#)

2 [Обоснование выбора технологий](#)

3 [Инструментарий](#)

4 [Архитектурный шаблон проектирования MVC](#)

5 [Шаблон проектирования практических задач "Фасад"](#)

[Заключение](#)

[Список использованных источников](#)

[Приложения](#)

Введение

Ни для кого не секрет, что видео игры прочно заняли свою позицию в современной индустрии развлечений. Существуют попытки выделить компьютерные игры как отдельную область искусства, наряду с театром, кино и т.п. Разработка игр может оказаться не только увлекательным, но и прибыльным делом, примеров этому предостаточно в истории. Первые примитивные компьютерные и видео игры были разработаны в 1950-х и 1960-х годах. Они работали на таких платформах, как осциллографы, университетские мейнфреймы и компьютеры EDSAC. Самой первой компьютерной игрой стал симулятор ракеты, созданный в 1942 году Томасом Голдсмитом Младшим (англ. Thomas T. Goldsmith Jr.) и Истл Рей Менном (англ. Estle Ray Mann). Позже, в 1952 году, появилась программа "ОХО", имитирующая игру "крестики-нолики", созданная А.С. Дугласом как часть его докторской диссертации в Кембриджском Университете. Игра работала на большом университетском компьютере, известном как EDSAC (Electronic Delay Storage Automatic Calculator). В настоящее время, разработка игры - это многомиллионный процесс, в котором задействована целая команда разработчиков, сложные современные технологии и даже маркетинговые ходы. Так как при написании курсовой работы в начале 2 курса мне пришлось познакомиться с таким языком программирования как Python и несколькими библиотеками, я решил для себя, что хочу попробовать себя в написании игры на Python (заранее понимал, что это достаточно сложная задача, т.к. до этого Python я не изучал).

1 Описание проекта

Перед тем, как начать составлять описание проекта мной был более подробно изучен язык Python, а именно библиотека pygame.

Что такое Pygame?

Pygame — набор модулей (библиотек) языка программирования Python, предназначенный для написания компьютерных игр и мультимедиа-приложений. Pygame базируется на мультимедийной библиотеке SDL.

Изначально Pygame был написан Питом Шиннерсом (Pete Shinnners). Начиная примерно с 2004/2005 года поддерживается и развивается

сообществом свободного программного обеспечения.

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.

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 Pygame – это фреймворк. Однако его сложно назвать "мощным инструментом". По своему объему и функционалу это скорее библиотека.

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.

В этом смысле Pygame можно сравнить с Tkinter, который через свои функции и классы предоставляет Питону доступ к графической библиотеке Tk.

Pygame — это «игровая библиотека», набор инструментов, помогающих программистам создавать игры. К ним относятся:

Графика и анимация;

Звук (включая музыку);

Управление (мышь, клавиатура, геймпад и так далее);

Игровой цикл.

В сердце каждой игры лежит цикл, который принято называть «игровым циклом». Он запускается снова и снова, делая все, чтобы работала игра. Каждый цикл в игре называется кадром.

В каждом кадре происходит масса вещей, но их можно разбить на три категории:

Обработка ввода (события).

Речь идет обо всем, что происходит вне игры — тех событиях, на которые она должна реагировать. Это могут быть нажатия клавиш на клавиатуре, клики мышью и так далее.

Обновление игры.

Изменение всего, что должно измениться в течение одного кадра. Если персонаж в воздухе, гравитация должна потянуть его вниз. Если два объекта встречаются на большой скорости, они должны взорваться.

Рендеринг (прорисовка).

В этом шаге всё выводится на экран: фоны, персонажи, меню. Все, что игрок должен видеть, появляется на экране в нужном месте.

Время.

Еще один важный аспект игрового цикла — скорость его работы. Многие наверняка знакомы с термином FPS, который расшифровывается как Frames Per Second (или кадры в секунду). Он указывает на то, сколько раз цикл должен повториться за одну секунду. Это важно, чтобы игра не была слишком медленной или быстрой. Важно и то, чтобы игра не работала с разной скоростью на разных ПК. Если персонажу необходимо 10 секунд на то, чтобы пересечь экран, эти 10 секунд должны быть неизменными для всех компьютеров.

Pygame-приложения могут работать под Android на телефонах и планшетах с использованием подмножества Pygame для Android (pgs4a). На этой платформе поддерживаются звук, вибрация, клавиатура, акселерометр. Это довольно важный аспект, т.к. это может быть одна из будущих целей моей игры (размещение ее на Google Play).

После изучения библиотеки rpygame, а также просмотра обучающих видеороликов я перешел к составлению описания проекта.

Проект заключается в следующем – первое, что видит пользователь – это меню игры, в котором после нажатия клавиши «Start game» запускается игра, а по нажатии клавиши «Quit» выход соответственно. Для более красивого отображения интерфейса реализована анимация бега Волка, а также изменения цвета кнопки при наведении мышкой. Звуковое сопровождение также присутствует.



После нажатия кнопки открывается окно игры. В левом верхнем углу отображается количество жизней и энергии. Снизу отображается Ваш текущий счет. Реализован прыжок по нажатии клавиши «Пробел» и стрельба Волка по нажатии клавиши «X» (энергия и является счетчиком возможных выстрелов). Также реализовано оповещение текущего кулдауна (время, через которое Волк снова может выстрелить). Создана случайная генерация 3 типов блоков (препятствий). Реализовано столкновение Волка с препятствием и отнимание жизни при их столкновении. Также присутствует случайный вылет Зайца и его стрельба по бегущему Волку (реализация отнимания жизни у Волка на данный момент не реализована), однако убийство Зайца при попадании выстрела Волка в него реализовано (однако Заяц вернётся, чтобы помешать Волку через определенный промежуток времени).

2 Обоснование выбора технологий

Так как основной задачей курсовой было написание проекта на объектно-ориентированном языке, то необходимо обосновать что такое

ООП и особенности Python, как объектно-ориентированного языка.

Объектно-ориентированное программирование (ООП) — это парадигма программирования, где различные компоненты компьютерной программы моделируются на основе реальных объектов. Объект — это что-либо, у чего есть какие-либо характеристики и то, что может выполнить какую-либо функцию.

Стоит обратить внимание на то, что объектно-ориентированное программирование — не зависящая от языка программирования концепция. Это общая концепция программирования и большинство современных языков, такие как Java, C#, C++ и Python поддерживают объектно-ориентированное программирование.

Объектно-ориентированный язык работает по следующим принципам:

Все данные представляются объектами;

Программа является набором взаимодействующих объектов, посылающих друг другу сообщения;

Каждый объект имеет собственную часть памяти и может иметь в составе другие объекты;

Каждый объект имеет тип;

Объекты одного типа могут принимать одни и те же сообщения (и выполнять одни и те же действия).

Идеи/принципы объектно-ориентированного программирования:

1. Наследование. Возможность выделять общие свойства и методы классов в один класс верхнего уровня (родительский). Классы, имеющие общего родителя, различаются между собой за счет включения в них различных дополнительных свойств и методов.
2. Инкапсуляция. Свойства и методы класса делятся на доступные из вне (опубликованные) и недоступные (защищенные). Защищенные атрибуты нельзя изменить, находясь вне класса. Опубликованные же атрибуты также называют интерфейсом объекта, т. к. с их помощью с объектом можно взаимодействовать. По идеи, инкапсуляция призвана обеспечить надежность программы, т.к. изменить существенные для существования объекта атрибуты становится невозможно.
3. Полиморфизм. Полиморфизм подразумевает замещение атрибутов, описанных ранее в других классах: имя атрибута остается прежним, а реализация уже другой. Полиморфизм позволяет специализировать (адаптировать) классы, оставляя при этом единый интерфейс взаимодействия.

Преимущества ООП.

В связи со своими особенностями объектно-ориентированное программирование имеет ряд преимуществ перед структурным (и др.) программированием. Выделим некоторые из них:

1. Использование одного и того же программного кода с разными данными. Классы позволяют создавать множество объектов, каждый из которых имеет собственные значения атрибутов. Нет потребности вводить множество переменных, т.к. объекты получают в свое распоряжение индивидуальные так называемые пространства имен. Пространство имен конкретного объекта формируется на основе класса, от которого он был создан, а также от всех родительских классов данного класса. Объект можно представить как некую упаковку данных.
2. Наследование и полиморфизм позволяют не писать новый код, а настраивать уже существующий, за счет добавления и переопределения атрибутов. Это ведет к сокращению объема исходного кода.

Особенность ООП.

ООП позволяет сократить время на написание исходного кода, однако ООП всегда предполагает большую роль предварительного анализа предметной области, предварительного проектирования. От правильности решений на этом предварительном этапе зависит куда больше, чем от непосредственного написания исходного кода.

Особенности ООП в Python.

По сравнению с другими распространенными языками программирования у Python можно выделить следующие особенности, связанные с объектно-ориентированным программированием:

1. Любое данное (значение) - это объект. Число, строка, список, массив и др. - все является объектом. Бывают объекты встроенных классов (как те, что перечисленные в предыдущем предложении), а бывают объекты пользовательских классов (тех, что создает программист). Для единого механизма взаимодействия предусмотрены методы перегрузки операторов.
2. Класс - это тоже объект с собственным пространством имен. Это нигде не было указано в данном цикле уроков. Однако это так. Поэтому правильнее было употреблять вместо слова «объект», слово «экземпляр». И говорить «экземпляр объекта», подразумевая под этим созданный на основе класса именно объект, и «экземпляр класса», имея в виду сам класс как объект.
3. Инкапсуляции в Python не уделяется особого внимания. В других языках программирования обычно нельзя получить напрямую доступ к свойству, описанному в классе. Для его изменения может быть предусмотрен специальный метод. В Python же это легко сделать, просто обратившись к свойству класса из вне. Несмотря на это в Python все-таки предусмотрены специальные способы ограничения доступа к переменным в классе.

3 Инструментарий

Обоснование используемых инструментов.

PyCharm предоставляет умную проверку кода, быстрое выявление ошибок и оперативное исправление. Вместе с автоматическим рефакторингом кода и богатыми возможностями в навигации. Кроме этого, скачать PyCharm не составит никакого труда.

Умный редактор кода.

Редактор кода PyCharm предоставляет первоклассную поддержку для Python, JavaScript, CoffeeScript, TypeScript, CSS, популярных языковых и многого другого. Оцените преимущества возможности поддержки нескольких языков, обнаружения ошибок и их исправления!

Умная навигация.

Воспользуйтесь умным поиском для перехода к любому классу, файлу, символу, или даже событию IDE и окну инструментов. Всего один клик, и вы переходите к супер-методу, тестированию, использованию, реализации, декларации и это как минимум!

Быстрый и безопасный рефакторинг.

С PyCharm вам открывается разумное обновление кода с безопасным удалением и переименованием, методом извлечения, вводной переменной, встроенной переменной или методом, и другими видами рефакторинга. Ориентированные на фреймворк и язык программирования рефакторинги помогут выполнить любое изменение в рамках целого проекта.

Встроенные Инструменты Разработки.

Большая коллекция инструментов PyCharm включает в себя интегрированный отладчик и запуск тестирования

- профайлер Python;
- встроенный терминал;
- интеграцию с большими VCS и встроенными инструментами баз данных;
- возможность удаленной разработки с удаленными интерпретаторами;
- интегрированный терминал ssh;
- интеграция с Docker и Vagrant.

Отладка, Тестирование и Профилирование

Используйте мощный отладчик с графическим интерфейсом под Python и JavaScript. Создавайте и проводите тестирования с поддержкой кода и драйвером текстов на основе графического пользовательского интерфейса. Получите полный контроль над своим кодом благодаря интеграции Python Profiler.

VCS, Развертывание и Удаленная Разработка

Сохраните свое время благодаря унифицированному пользовательскому интерфейсу для работы с Git, SVN, Mercurial и другими системами контроля версий. Запускайте и проводите отладку своего приложения в удаленном режиме. Вам доступна простая конфигурация автоматического развертывания для удаленного хоста или VM и управление вашей инфраструктурой при помощи Vagrant и Docker.

Инструменты Базы Данных

- Access Oracle
- SQL Server
- PostgreSQL
- MySQL

И другие базы данных прямо из IDE. Вы можете рассчитывать на помощь PyCharm при редактировании кода SQL, выполнении запросов, просмотре данных и внесении изменений в схемы.

Веб разработка.

В дополнении к Python, PyCharm предоставляет первую поддержку различных фреймворков веб разработки от Python, отдельных языков, JavaScript, CoffeeScript, TypeScript, HTML/CSS, AngularJS, Node.js и других.

Веб фреймворки Python.

PyCharm предоставляет отличную поддержку отдельных фреймворков для современных фреймворков веб разработки, таких как Django, Flask, Google App Engine, Pyramid, и web2py, включая отладчик шаблонов Django, инструменты manage.py и appcfg.py, автозаполнение и навигацию.

JavaScript и HTML.

PyCharm предоставляет первую поддержку для JavaScript, CoffeeScript, TypeScript, HTML и CSS, а также их современных преемников. Отладчик JavaScript также включен в PyCharm и является интегрированным с конфигурацией запуска сервера Django.

Live Edit.

Live Editing Preview позволяет вам открывать страницу в редакторе и браузере для поиска и мгновенного нахождения недавно внесенных изменений в браузере. PyCharm сохраняет ваши изменения в автоматическом режиме, в то время как браузер сам обновляет страницу, показывая результаты изменений.

Научные инструменты.

PyCharm интегрируется с IPython Notebook, имеет интерактивную консоль Python и поддерживает как Anaconda, так и множество других пакетов, включая Matplotlib и NumPy.

Интеграция IPython Notebook.

PyCharm интегрируется с IPython Notebook и обеспечивает решение, которое комбинирует возможности IPython Notebook с дополнительными преимуществами, которые может предложить большая часть интеллектуальной среды разработки Python, включая автоматическое завершение, навигацию, проверку ошибок, и так далее.

Интерактивная консоль Python.

Еще одна причина [скачать PyCharm](#) заключается в том, что вы можете запустить консоль Python REPL в PyCharm, что дает массу преимуществ: мгновенная проверка синтаксиса с дополнительными проверками, сопоставления скобок и кавычек, и, конечно, завершением кода.

Поддержка научного стека.

PyCharm предоставляет специальную поддержку научных библиотек. Он поддерживает Anaconda, Numpy, Matplotlib и другие научные библиотеки, предоставляя пользователю глубокое понимание кода, интерактивные графики, просмотр массивов и многое другое.

Настраиваемая и кроссплатформенная IDE.

Вы можете скачать PyCharm и установить на Windows, Mac OS и Linux используя один лицензионный ключ. Наслаждайтесь отлично настроенным рабочим пространством с настраиваемыми цветовыми схемами и горячими клавишами, с доступной эмуляцией VIM.

Настраиваемый пользовательский интерфейс.

Существует ли хотя бы один разработчик, который не любит настраивать свои инструменты? Мы с таким еще не сталкивались, так что к вашим услугам – простая настройка UI от PyCharm. Наслаждайтесь удобным рабочим пространством с настраиваемыми цветовыми схемами и горячими клавишами.

Кроссплатформенная среда разработки.

Вы можете скачать PyCharm и установить его на Windows, Mac OS или Linux. Кроме этого, вы можете установить его на такое количество компьютеров, которое сами пожелаете, пользуясь той же средой и функционалом на любом компьютере.

4 Архитектурный шаблон проектирования MVC

MVC — это фундаментальный паттерн, который нашел применение во многих технологиях, дал развитие новым технологиям и каждый день облегчает жизнь разработчикам.

Впервые паттерн MVC появился в языке SmallTalk. Разработчики должны были придумать архитектурное решение, которое позволяло бы отделить графический интерфейс от бизнес логики, а бизнес логику от данных. Таким образом, в классическом варианте, MVC состоит из трех частей, которые и дали ему название. Рассмотрим их:

Модель.

Под Моделью, обычно понимается часть содержащая в себе функциональную бизнес-логику приложения. Модель должна быть полностью независима от остальных частей продукта. Модельный слой ничего не должен знать об элементах дизайна, и каким образом он будет отображаться. Постигается результат, позволяющий менять представление данных, то как они отображаются, не трогая саму Модель.

5 Шаблон проектирования практических задач "Фасад"

Решение

Определить одну точку взаимодействия с подсистемой — фасадный объект, обеспечивающий общий интерфейс с подсистемой, и возложить на него обязанность по взаимодействию с её компонентами. Фасад — это внешний объект, обеспечивающий единственную точку входа для служб подсистемы. Реализация других компонентов подсистемы закрыта и не видна внешним компонентам. Фасадный объект обеспечивает реализацию GRASP паттерна [Устойчивый к изменениям](#) (Protected Variations) с точки зрения защиты от изменений в реализации подсистемы.

Заключение

Ни для кого не секрет, что видео игры прочно заняли свою позицию в современной индустрии развлечений. Существуют попытки выделить компьютерные игры как отдельную область искусства, наряду с театром, кино и т.п. Разработка игр может оказаться не только увлекательным, но и прибыльным делом, примеров этому предостаточно в истории. Первые примитивные компьютерные и видео игры были разработаны в 1950-х и 1960-х годах. Они работали на таких платформах, как осциллографы, университетские мейнфреймы и компьютеры EDSAC. Самой первой компьютерной игрой стал симулятор ракеты, созданный в 1942 году Томасом Голдсмитом Младшим (англ. Thomas T. Goldsmith Jr.) и Истл Рей Менном (англ. Estle Ray Mann). Позже, в 1952 году, появилась программа "OXO", имитирующая игру "крестики-нолики", созданная А.С. Дугласом как часть его докторской диссертации в Кембриджском Университете. Игра работала на большом университетском компьютере, известном как EDSAC (Electronic Delay Storage Automatic Calculator). В настоящее время, разработка игры - это многомиллионный процесс, в котором задействована целая команда разработчиков, сложные современные технологии и даже маркетинговые ходы. Так как при написании курсовой работы в начале 2 курса мне пришлось познакомиться с таким языком программирования как Python и несколькими библиотеками, я решил для себя, что хочу попробовать себя в написании игры на Python (заранее понимал, что это достаточно сложная задача, т.к. до этого Python я не изучал).

Список использованных источников

1. [url] **Laravel документация** <http://laravel.com>

Приложения

1. [электронный документ] [5ece8776bb090_shablon_kr.docx](#)