

Доклад на защиту Ознакомительной практики по теме:

“Разработка серверной части программного продукта”

1 Pygame

1.1 Pygame - определение

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

- Графика и анимация
- Звук (включая музыку)
- Управление/обработка событий от пользователя (мышь, клавиатура, геймпад и так далее)

1.2 Принцип работы

В ходе разработки серверной части продукта данная библиотека использовалась для обеспечения наглядности тех или иных функций, добавляемых в код программы, в процессе его написания и для простоты визуализации работы программы.

Таким образом, принцип работы следующий:

- Подключение библиотеки
- Инициализация игры и игрового окна
- В основном игровом цикле на каждой итерации контроль FPS (frames per second)
- В основном игровом цикле обработка событий (пользовательского ввода) на каждой итерации при необходимости. В случае работы серверной части в качестве источника изменения данных об игровом поле служат данные, получаемые сервером от клиентов, поэтому пользовательского ввода в коде серверной части не присутствует, как правило
- В основном игровом цикле обновление данных игрового поля на каждой итерации
- В основном игровом цикле отрисовка графики/рендеринг на каждой итерации

2 Модуль socket

2.1 Сокет - определение

Сокет - название программного интерфейса для обеспечения обмена данными между процессами. Процессы при таком обмене могут выполняться как на одной ЭВМ, так и на различных ЭВМ, связанных между собой сетью. Сокет - абстрактный объект, представляющий конечную точку соединения.

Следует различать **клиентские** и **серверные сокеты**. Клиентские сокеты грубо можно сравнить с конечными аппаратами телефонной сети, а серверные - с коммутаторами. Клиентское приложение (например, браузер) использует только клиентские сокеты, а серверное (например, веб-сервер, которому браузер посылает запросы) - как клиентские, так и серверные сокеты.

Для взаимодействия между машинами с помощью стека протоколов TCP/IP используются адреса и порты. Адрес представляет собой 32-битную структуру для протокола IPv4, 128-битную для IPv6. Номер порта — целое число в диапазоне от 0 до 65535 (для протокола TCP).

Эта пара определяет сокет («гнездо», соответствующее адресу и порту).

В процессе обмена, как правило, используется два сокета — сокет отправителя и сокет получателя. Например, при обращении к серверу на HTTP-порт сокет будет выглядеть так: 194.106.118.30:80, а ответ будет поступать на mmm.nnn.ppp.ddd:xxxxx.

Каждый процесс может создать «слушающий» сокет (серверный сокет) и *привязать* его к какому-нибудь порту операционной системы (в UNIX непривилегированные процессы не могут использовать порты меньше 1024).

Слушающий процесс обычно находится в цикле ожидания, то есть просыпается при появлении нового соединения. При этом сохраняется возможность проверить наличие соединений на данный момент, установить тайм-аут для операции и т. д.

Каждый сокет имеет свой адрес. ОС семейства UNIX могут поддерживать много типов адресов, но обязательными являются INET-адрес и UNIX-адрес. Если привязать сокет к UNIX-адресу, то будет создан специальный файл (файл сокета) по заданному пути, через который смогут общаться любые локальные процессы путём чтения/записи из него. Сокеты типа INET доступны из сети и требуют выделения номера порта.

Обычно клиент явно «подсоединяется» к слушателю, после чего любое чтение или запись через его файловый дескриптор будут передавать данные между ним и сервером.

2.2 Сокеты в python

Модуль `socket` обеспечивает доступ к интерфейсу сокета BSD. Он доступен на всех современных системах Unix, Windows, macOS и, возможно, на дополнительных платформах.

Интерфейс Python представляет собой простую транслитерацию интерфейса системного вызова и библиотеки Unix для сокетов в объектно-ориентированный стиль Python: функция «`socket()`» возвращает объект сокета, методы которого реализуют различные системные вызовы сокетов, функция «`bind("IPv4address", port)`» привязывает сокет сервера к порту указанного адреса протокола IPv4. Типы параметров несколько более высокого уровня, чем в интерфейсе C: как и в случае операций «`read()`» и «`write()`» в файлах Python, выделение буфера при операциях приема происходит автоматически, а длина буфера неявно указывается при операциях отправки.