

Лабораторная работа № 07

Тема: Асинхронное программирование

Цель:

- Знакомство с асинхронным программированием;
- Получение практических навыков в параллельной обработке данных;
- Получение практических навыков в синхронизации потоков;

Порядок выполнения работы

- Ознакомиться с теоретическим материалом.
- Получить у преподавателя вариант задания.
- Реализовать задание своего варианта в соответствии с поставленными требованиями.
- Подготовить тестовые наборы данных.
- Создать репозиторий на GitHub.
- Отправить файлы лабораторной работы в репозиторий.
- Отчитаться по выполненной работе путём демонстрации работающей программы на тестовых наборах данных (как подготовленных самостоятельно, так и предложенных преподавателем) и ответов на вопросы преподавателя (как из числа контрольных, так и по реализации программы).

Требования к программе

Модифицируйте вашу лабораторную работу №6 следующим образом:

- Должно быть реализовано три потока:
 - **Поток**, который осуществляет передвижение NPC на определенное расстояние (см. таблицу), а также определяет, что два NPC находятся на расстоянии убийства (см. таблицу). Если два NPC вступили в «бой», то он создает задачу для потока, осуществляющего бои.
 - **Поток**, который осуществляет бои. Если один прс может убивать другой, то каждый прс «кидает 6-гранный кубик» определяя силу атаки и силу защиты (соответственно). Если сила атаки больше, чем сила защиты – то происходит убийство.
 - Мертвые прс не передвигаются (у нас тут без некромантов).
 - Живые прс не могут покинуть карту (размер задается через константы, например 100 x 100)
 - **Основной поток** раз в секунду печатает карту. Мертвые прс на карте не отображаются.
- Осуществить контроль доступа к разделяемым ресурсам с помощью `std::shared_lock` и `std::lock_guard`
- Осуществить контроль к потоку вывода `std::cout` через `std::lock_guard`
- Потоки могут запускаться как `std::thread` с использованием лямбда функций или функторов (классов с перегруженным оператором `operator()`)
- Вначале игры должно создаваться 50 прс в случайных локациях.
- Игра должна останавливаться через 30 секунд и выводить на экран список выживших

Усложненный вариант (+10 очков к максимальному балу):

- Реализовать перемещение и сражение NPC в одном потоке, используя `coroutine`.
- Основной поток, так же должен будет выводить данные о карте на экран

Таблица убиваемости:

| Кто нагоняет | Расстояние хода | Расстояние убийства |
|----------------------|-----------------|---------------------|
| Орк | 20 | 10 |
| Белка | 5 | 5 |
| Друид | 10 | 10 |
| Странствующий рыцарь | 30 | 10 |
| Эльф | 10 | 50 |
| Дракон | 50 | 30 |
| Медведь | 5 | 10 |
| Разбойник | 10 | 10 |
| Оборотень | 40 | 5 |
| Принцесса | 1 | 1 |
| Жаба | 1 | 10 |
| Работорговец | 10 | 10 |
| Пегас | 30 | 10 |
| Выпь | 50 | 10 |
| Выхухоль | 5 | 20 |
| Бык | 30 | 10 |

Отчет

1. Код программы на языке C++.
2. Ссылка на репозиторий на GitHub.