

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет програмної інженерії та бізнесу

Кафедра інженерії програмного забезпечення

Розрахункова робота

з дисципліни «Програмування віртуальної реальності»

(назва дисципліни)

на тему: «Зв'язок віртуального та реального світу(Взаємодія з датчиками руху/положення у просторі)»

Виконав: студент 4 курсу групи № 545В

напряму підготовки (спеціальності)

123 - Комп'ютерна інженерія

(шифр і назва напряму підготовки /спеціальності)

Ткаченко І. Д.

(прізвище й ініціали студента)

Прийняв: доцент Лучшев П. О.

(посада, науковий ступінь, прізвище й ініціали)

Національна шкала: _____

Кількість балів: _____

Оцінка ECTS: _____

1. Постановка задачі

1.1. Загальне завдання

Головною метою є здійснення всебічного аналізу ефективності, виявлення чинників, що обмежують результативність, з урахуванням попередніх досліджень у межах тем з 1 по 4.

2. Вступ

Для виконання поставленої задачі я зроблю огляд коду, та виконаю аналіз продуктивності коду. Фактори обмежуючі ефективність.

3. Генерація ландшафту

Основна мета цього коду - розробка програми для вивчення генерації ландшафту за допомогою OpenGL. Програма має вбудований користувацький інтерфейс, де користувач може налаштовувати параметри генерації ландшафту.

3.1. Організація коду:

Код розділений на функції та функціональні блоки для забезпечення читабельності та обслуговуваності. Це дозволяє легше масштабувати проект у майбутньому. На жаль, у коді занадто багато глобальних змінних, що може створювати проблеми великих проектах. Рекомендується обмежувати використання глобальних змінних і передавати параметри функціям за необхідності..

3.2. OpenGL та Графічний код:

Код використовує застарілий OpenGL API, що може обмежувати функціональність та продуктивність. Рекомендується розглянути перехід на сучасніші версії OpenGL або використання сучасних бібліотек, таких як GLFW або SDL, для поліпшення сумісності та розширення можливостей. Також в коді є неактивний код, який слід видалити або розширити для підтримки чистоти та зрозумілості проекту. Застосування застарілої функції gluPerspective для установки перспективи також може бути замінено сучасними матричними операціями для поліпшення читабельності та сумісності.

3.3. Продуктивність:

Велика кількість операцій з графікою виконується кожен кадр, що може призвести до значного падіння продуктивності. Рекомендується оптимізувати ці операції, наприклад, шляхом генерації шуму та обчислення кольорів на GPU або виконання їх лише раз під час ініціалізації. Генерація геометрії на кожному кадрі також може негативно впливати на продуктивність, і рекомендується розглянути можливість генерації геометрії лише один раз та її подальше використання, що буде доцільним для статичних об'єктів, які не змінюються впродовж гри.

3.4. Методи оптимізації:

3.4.1. Оптимізація графічних операцій:

- Розгляньте можливість використання сучасних версій OpenGL або бібліотек, таких як GLFW або SDL, для покращення продуктивності та функціональності.
- Перегляньте неактивний код та видаліть його або доповніть для забезпечення чистоти та зрозумілості проекту.
- Замініть застарілу функцію `gluPerspective` сучасними матричними операціями для поліпшення читабельності та сумісності зі сучасними версіями OpenGL.

3.4.2. Оптимізація генерації ландшафту:

- Розгляньте використання GPU для генерації шуму та обчислення колірів для поліпшення продуктивності.
- Виконайте генерацію шуму та колірів лише раз під час ініціалізації, а не на кожному кадрі.
- Розгляньте можливість оптимізації алгоритмів генерації геометрії та використання результуючої геометрії лише один раз.

3.4.3. Оптимізація користувацького інтерфейсу:

- Обмежте використання глобальних змінних та передавайте параметри функціям за необхідності.
- Розгляньте можливість використання кешування результатів обчислень та асинхронного завантаження ресурсів для поліпшення швидкодії інтерфейсу.

3.4.4. Оптимізація роботи з пам'яттю:

- Розгляньте можливість оптимізації роботи з пам'яттю, зокрема управлінням масивами та їх розміром.
- Використовуйте ефективні структури даних для зберігання та обробки інформації.

3.4.5. Оптимізація вводу користувача:

- Оптимізуйте обробку вводу користувача, зменшуючи кількість опитувань клавіш та оптимізуючи код для керування камерою.

3.4.6. Профілювання та аналіз продуктивності:

- Використовуйте інструменти для профілювання коду та аналізу продуктивності для виявлення та усунення буттялеків.

3.4.7. Оптимізація рендерингу:

- Розгляньте можливість використання більш ефективних методів рендерингу, таких як VBO (Vertex Buffer Objects) та шейдери.
- Вивчіть можливість використання більш ефективних алгоритмів рендерингу для поліпшення швидкодії програми.

4. Камера

4.1. Організація коду:

Код розгорнуто на функції та функціональні блоки, щоб полегшити читання та підтримку. Це дозволяє легше масштабувати проект в майбутньому. Також було виправлено деякі коментарі, які можуть вважатися зайвими або очевидними. Використані статичні змінні, але їх можна замінити константами або використовувати більш безпечні підходи.

4.2. Продуктивність:

Велика кількість операцій з графікою виконується на кожен кадр, включаючи обчислення напрямків та руху, що може призвести до значного зниження продуктивності. Метод Look() та інші функції можна оптимізувати для зменшення обчислювального завантаження. Але код максимально продуктивний.

4.3. Способи Оптимізації:

Оптимізації можливо лише при використанні Асемблер вставок в математичних розрахунках. Можливо краще буде використовувати алгоритм кватерніонів, який дасть кращу взаємодію з роботою камери

5. Моделі

5.1. Організація коду:

Хоча структура коду ефективно організована на функції та функціональні блоки, існує потенціал для подальшого поліпшення читабельності та обслуговуваності. Рекомендується розглянути можливість розділити деякі функціональні блоки на окремі функції або класи, що сприятиме легшому розумінню та обслуговуванню коду. Додаткові коментарі можуть бути використані для пояснення ключових частин коду або дій.

5.2. OpenGL та Графічний код:

Застосування застарілого OpenGL API може бути ризикованим у майбутньому. Рекомендується розглянути перехід на більш сучасні версії OpenGL або використання бібліотек, таких як GLFW або SDL, для покращення сумісності та продуктивності. Ініціалізація та робота із світлом реалізовані, проте можливо, буде

корисною додаткова робота над освітленням для основного об'єкта. Також є можливість оптимізації шляхом винесення функцій ініціалізації та обробки світла у відокремлені методи.

5.3. Продуктивність:

Код містить значну кількість обчислень матриць та трансформацій на кожен кадр, що може впливати на продуктивність. Рекомендується оптимізувати та кешувати результати обчислень для зменшення навантаження на процесор. Також можна розглянути використання менш ресурсоємних функцій та ресурсів-вимірників, які не змінюються кожен кадр.

5.4. Способи Оптимізації:

Для полегшення обслуговування та розуміння коду можливо варто розглянути використання окремого класу або структури для керування камерою та світлом. Також рекомендується подальша оптимізація шляхом використання менш ресурсоємних функцій та ресурсів-вимірників, які не змінюються кожен кадр, для підвищення продуктивності.

6. Стереопара

6.1. Код:

Код Стереопари достатньо простий та тривіальний. Его суть це створення пари точок зору, з різними кутами по осі X.

6.2. Покращення коду:

Для покращення коду, я би краще використовував більш інноваційні методи створення ефекту Стереопари. Наприклад використовуючи шейдер для створення кадру, та змінював не тільки кут зору, а й приламував картинку щоб була можливість одягнути окуляри с лінзами та получить 3x вимірну картинку для зору людини.

7. Висновки

У підсумку цієї роботи я отримав значний досвід у створенні проектів, пов'язаних із віртуальною реальністю. Проаналізувавши свої роботи та враховуючи вчинені помилки, я зрозумів, як їх виправити і покращити свої навички. Цей досвід став для мене важливим кроком у вдосконаленні умінь та знань у цій області. Зараз я впевнений, що зможу успішно використовувати здобуті навички у майбутніх проектах, де виникає потреба у віртуальній реальності.