Yujung Kim Portfolio

---

1. ⚙️YJ Engine, Custom 2D Engine

- 개발 환경 : Visual Studio 2022, Windows 10
- 사용 기술 : C / C++
- 개발 기간 : 2024.10 ~ 2024.11 (1개월)
- **git** : YJ_Engine.git

- 엔진 설명 : C++로 직접 구현한 컴포넌트 기반의 2D 게임 엔진입니다.

- 개발 내용

    1. 컴포넌트 기반 엔진의 핵심 컴포넌트 구현

        OpenGL로 텍스처 출력을 하는 Sprite 컴포넌트, 오브젝트의 위치와 크기를 계산하는 Transform 컴포넌트를 GLM 라이브러리를 사용하여 구현했습니다.

        Singleton Pattern을 사용하여 게임 루프와 게임의 상태를 관리하는 Game State Manager을 구현하였습니다.

        게임 엔진을 설계 및 개발하며 컴포넌트 기반 구조를 이해했으며 쉐이더, 텍스처 매핑 등 그래픽스의 이해도 또한 높였습니다.

        Json을 사용하여 각 컴포넌트의 상태를 저장하는 Serializer를 구현하여 게임의 저장과 로드를 간편하게 처리하였습니다.

    2. 에디터 모드 구현

        ImGui를 활용하여 직관적이고 편리한 GUI를 구현했고 그 결과 오브젝트의 생성과 삭제, 저장과 로드를 쉽게 사용할 수 있었습니다.

    3. 충돌 체크 개선

        기존에는 모든 오브젝트 간의 충돌을 확인했기 때문에 연산 횟수가 많았습니다. 이를 개선하고자 Collision Layer Mask를 구현했습니다. 설정된

레이어 사이 충돌만 확인함으로써 실행 시간 중 연산 횟수를 줄이고 게임 성능을 향상 시켰습니다.

---

2. 👾Digipen Invaders (Action Arcade Game)

- 개발 환경 : Visual Studio 2022, Windows 10
- 사용 기술 : C / C++, Custom Engine
- 개발 기간 : 2024.09 ~ 2024.10 (1개월)
- **git** : alpha_project.git

- 게임 설명 : Space Invaders 모작

- 개발 내용

1. 컴포넌트 기반 엔진 기능 구현

   이전에 아무런 설계 과정 없이 개발 했던 프로젝트에서 오브젝트 간의 코드 중복이 많고 코드 간의 의존도가 높았습니다. 그러다 보니 협업 과정에서 서로의 코드가 많이 얽혀있어 진행에 어려움이 있었습니다.

   그래서 이 프로젝트에서는 코드의 중복을 줄이고 코드의 재사용성을 높이기 위해 컴포넌트 기반으로 개발을 진행했습니다.

   오브젝트들의 특성은 컴포넌트로 만들어 필요한 오브젝트에 추가하여 사용했습니다. 그 결과 코드의 가독성과 재사용성이 높아졌고 디버깅 과정도 훨씬 수월했습니다.

2. 최고 기록 / 랭킹 시스템 기능 구현

   json library를 사용하여 게임 데이터를 관리했습니다. 데이터의 구조가 직관적이라 원하는 데이터를 용도에 맞게 저장하고 불러오기 쉬웠습니다. 다양한 데이터 타입을 지원하고 데이터 파싱이 쉽다는 점 또한 유용하여 json을 선택했습니다.

데이터의 저장을 위해 사용자 환경에서 값을 받아 디렉토리를 생성했습니다. 이 때, 윈도우 시스템에서 C 드라이브 아래에 파일 쓰기 권한 문제가 있었습니다. 그래서 관리자 권한이 필요하지 않은 사용자 문서 폴더 아래 새로운 디렉토리를 생성하여 저장할 수 있도록 하였습니다.

- 실행 영상

https://youtu.be/sQGeNWUNJAU

---

3. 🎮비주얼 노벨 게임

- 개발 환경 : Visual Studio 2022, Windows 10
- 사용 기술 : C / C++, WIN32 API, GDI+
- 개발 기간 : 2024.02 ~ 2024.03 (1개월)

리소스는 리그오브레전드의 일러스트를 직접 가공하여 사용했습니다.

- 설명 : 선택지에 따라 결과가 달라지는 비주얼 노벨 게임입니다.
- 개발 내용

1. Win32 / GDI+

Windows 환경에서 그래픽을 처리하기 위해 GDI+를 사용했습니다. 이미지와 텍스트를 출력하는 함수를 구현하고 레퍼런스 카운팅 방법으로 직접 리소스 관리를 관리했습니다. 이 과정에서 버퍼에 이미지가 그려지는 원리를 이해하고 백버퍼를 활용하는 더블 버퍼링 기법으로 이미지를 출력하는 방식을 배웠습니다. 직접 버퍼를 관리하는 일이 어려움과 동시에, 섬세한 이미지 처리가 가능하다는 점에서 장점을 느꼈습니다.

2. 인코딩

   텍스트 기반 게임이기 때문에, 문자열 처리가 핵심이었습니다. 윈도우 운영체제에서 인코딩 방식으로 UTF-16 방식을 이용하고, 한글은 UTF-8을 사용합니다. 정상적인 문자열 처리를 위해 한글의 인코딩을 UFT-8에서 UTF-16으로 변환 해주는 과정이 필요했습니다. 그래서 직접 입력한 한글을 UTF-16으로 변환해주는 함수를 구현하여 윈도우 API 함수를 사용 할 수 있었습니다.

3. 문자열 처리

   비주얼 노벨이라는 장르 특성 상, 텍스트 연출이 주는 효과가 중요합니다. 대사를 글자 하나하나 출력하거나 Fade-In / Out을 주는 등 직접 텍스트 효과를 구현하며 텍스트를 다루는 데 익숙해졌습니다.

- 실행 영상

https://youtu.be/n9OlChvo8Q0

언어

TOEIC 880

연락처

📧 **email** : ujungg3@gmail.com

📑 **git** : https://github.com/Yujxxng

🗒️ **linked-in** : www.linkedin.com/in/yujung-kim