

캡스톤디자인(Capstone Design) 결과 보고서

1. 과제 현황

과제명	국문	컴포넌트 기반 설계로 제작한 C++ 헬리콥터 게임 캡스톤 디자인			
	영문	C++ Helicopter Game Capstone Design made by Component-Based Design			
연도-학기	2022- 1	교과목명	게임캡스톤디자인	학부(과)	게임소프트웨어전공
지도교수	소속	게임소프트웨어전공		성명	김혜영

2. 제작자

구분	학부(과)	학번	이름	휴대전화	E-Mail
대표	게임소프트웨어전공	B677011	방찬웅	010-5171-8541	tkfkdcksdnd@daum.net

3. 과제 결과

작품 유형	<input type="checkbox"/> 시작품
	<input type="checkbox"/> 아이디어 <input checked="" type="checkbox"/> 소프트웨어 <input type="checkbox"/> 논문
과제 결과 및 주요 성과	컴포넌트 기반 설계를 사용한 C++ 콘솔 게임 제작 게임에 대한 상세 클래스 다이어그램, 시퀀스 다이어그램 제작
개선방안	1. 컴포넌트간의 의존성을 낮추기 위해 메시지 통신 도입 2. Modern C++을 사용하여 더욱 효율적인 코드 작성 (RAII, Lambda, Algorithms)

캡스톤디자인 결과 보고서

작품명

컴포넌트 기반 설계로 제작한 C++ 헬리콥터 게임

1. 작품 배경, 개요 및 필요성

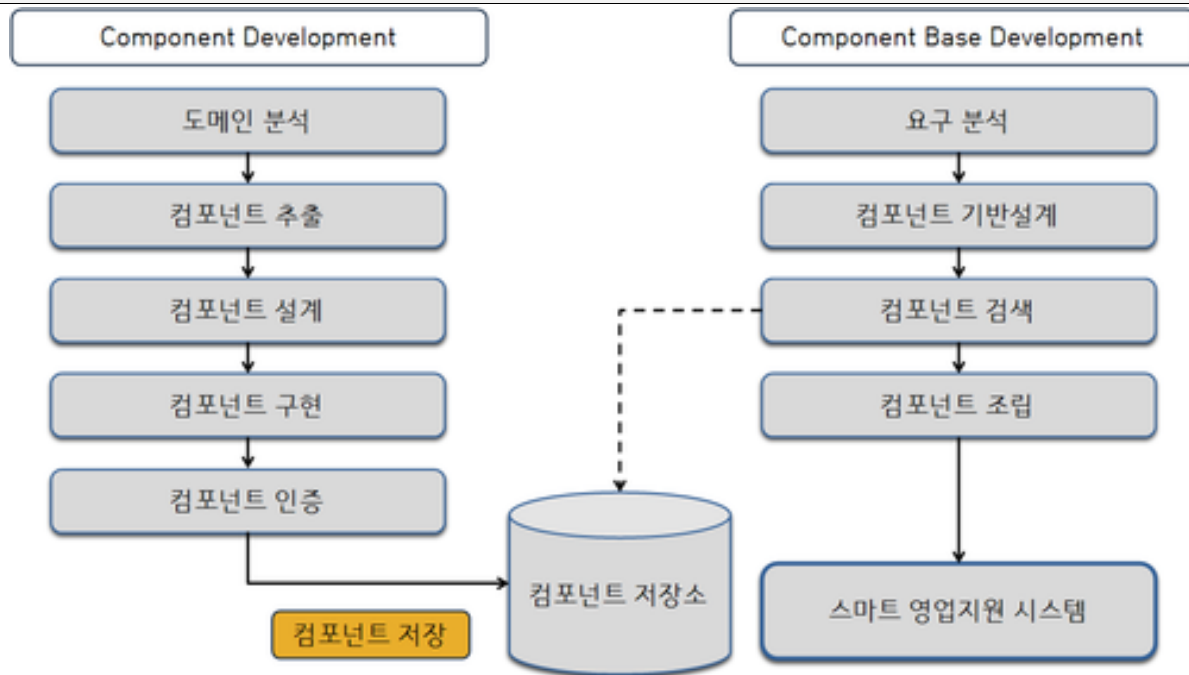
- 현재 상용 게임들은 대부분 게임 엔진을 통해 제작되며, 인디 게임 개발자에게도 게임 엔진은 게임을 개발하는데 있어 굉장히 중요한 툴로 자리잡음.
- 경쟁력 있는 프로그래머가 되기 위해선 게임 엔진을 잘 다룰 뿐만 아니라 게임 엔진의 구조에 대한 깊은 이해 필요.
- 널리 사용되고 있는 Unity, Unreal 엔진 등은 컴포넌트 기반으로 게임 제작이 이루어지므로, 컴포넌트 기반 설계에 대한 이해 필요.
- 개발 전 충분한 설계로 개발 도중 생기는 문제에 대해 더욱 빠르게 파악하고 해결할 수 있도록 함.

2. 작품의 이론, 기술, 디자인(그래픽) 현황

(캡스톤 디자인 과제의 이론적, 기술적 요소, 알고리즘 등)

2-1 컴포넌트 기반 개발 (CBD : Component-Based Development)

캡스톤디자인 결과 보고서



● 재사용 가능한 컴포넌트를 기반으로 소프트웨어를 개발하는 방법론

● 컴포넌트 : 특정한 기능을 수행하기 위해 독립적으로 개발되고, 잘 정의된 인터페이스를 가지며, 다른 부품과 조립되어 응용 시스템을 구축하기 위해 사용되는 소프트웨어 부품

● 장점

1. 계층 구조의 복잡한 설계와 클래스 비대화를 완화할 수 있다.
2. 컴포넌트의 높은 재사용성으로 시간, 노력, 비용이 절감되며, 개발 생산성이 향상된다.
3. 새로운 기능을 추가하는 것이 간단하여 확장성이 보장된다.

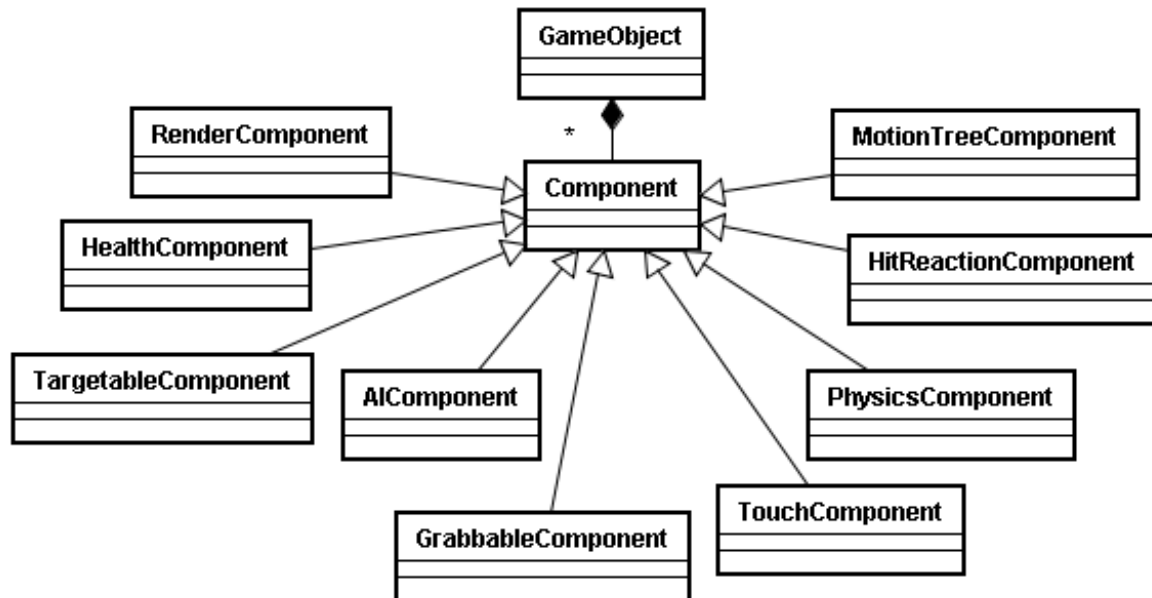
2-2 Unity Engine에서의 Component

● 게임 오브젝트 : 게임에 존재하는 모든 오브젝트이며 기능을 구현하는 컴포넌트의 컨테이너. 기본적으로 Transform(Position, Rotation, Scale) 컴포넌트를 포함하고 있다.

● 컴포넌트 : 게임에서 오브젝트와 동작에 관한 기본 구성 요소. 모든 게임 오브젝트의 작동과 관련한 부품이다.

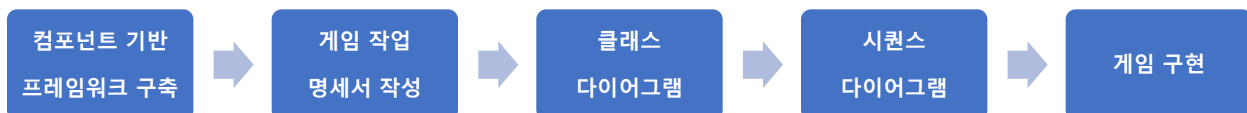
캡스톤디자인 결과 보고서

- 컴포넌트 활용 : 게임 오브젝트가 해야 할 기능들을 각각 별도의 컴포넌트 객체로 생성하여 게임 오브젝트에 연결하는 방식



3. 작품의 개발 방법 및 과정

● 제작 순서



3-1 프레임워크 구축

캡스톤디자인 결과 보고서

소스 파일

- ▶ Behaviour.cpp
- ▶ Borland.cpp
- ▶ Component.cpp
- ▶ GameObject.cpp
- ▶ Input.cpp
- ▶ main.cpp
- ▶ Position.cpp
- ▶ Renderer.cpp
- ▶ Scene.cpp
- ▶ Screen.cpp
- ▶ Transform.cpp

헤더 파일

- ▶ Behaviour.h
- ▶ Borland.h
- ▶ Component.h
- ▶ GameObject.h
- ▶ Input.h
- ▶ Position.h
- ▶ Renderer.h
- ▶ Scene.h
- ▶ Screen.h
- ▶ Transform.h

Main : Game main loop

Screen : 게임 창에 대한 클래스 (Singleton 패턴)

Input : 키보드, 마우스 입력에 대한 클래스 (Singleton 패턴)

Position : 좌표 정보를 가지고 있는 구조체

Borland : 콘솔 창의 커서 정보에 대한 클래스

GameObject : 게임에 존재할 Object 클래스.

Scene : GameObject들이 배치되어 구성되는 GameObject

Component : GameObject에 필요한 기능을 가지고 있으며,
GameObject에 탈부착이 가능한 컴포넌트

Transform : GameObject의 Transform 정보를 가지고 있는 컴포넌트

Renderer : GameObject가 Screen에 그려질 모양, 크기 정보를 가지고
있는 컴포넌트

Behaviour : Input, Renderer 컴포넌트를 상속해주는 컴포넌트

3-2 게임 작업 명세서 작성

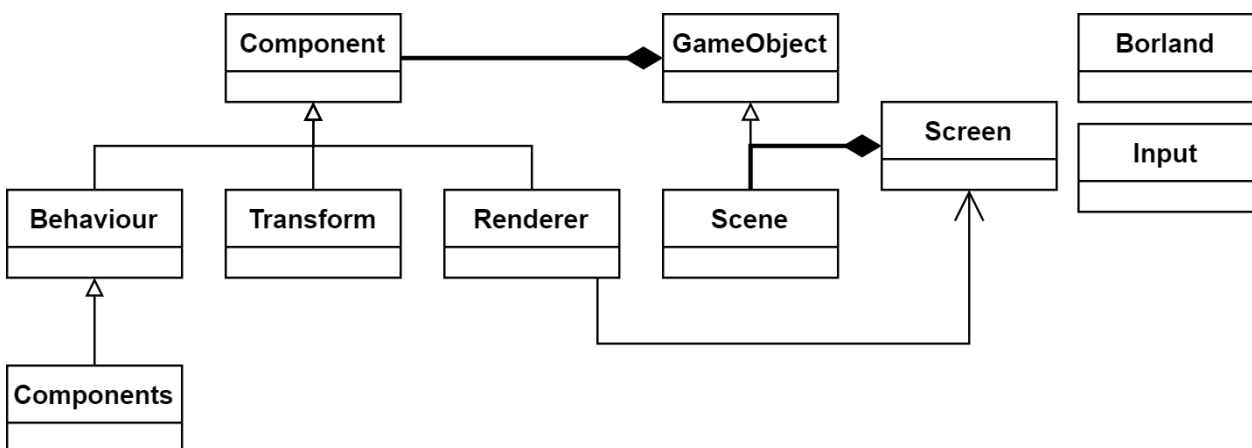
게임 / 작업 명세서

컴포넌트 기반 설계를 이용한 게임 엔진 프레임 워크를 구축한 뒤 만든 프레임 워크를 이용하여 콘솔 헬리콥터 게임을 만든다.

헬리콥터 게임

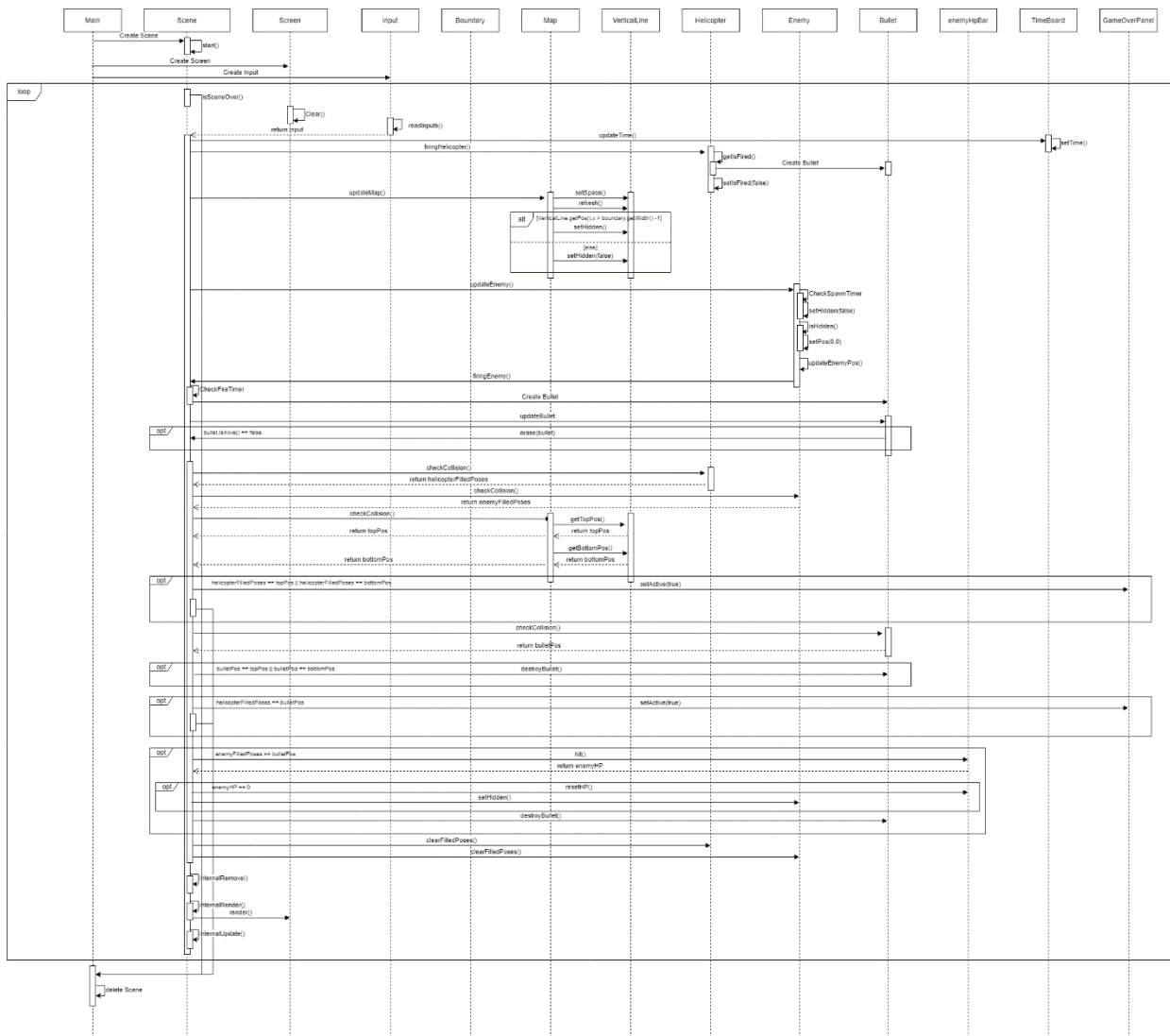
1. 헬리콥터는 WASD키로 상하좌우 이동이 가능하며, Space키로 총알을 발사한다.
2. 헬리콥터는 날개가 돌아가는 애니메이션을 보여준다.
3. 맵은 오른쪽에서 왼쪽으로 고정된 속도로 자동생성되며, 세 종류의 다른 타일로 구성되어있다.
4. 맵은 헬리콥터가 이동할 수 있는 간격이 좁아지거나 넓어지면서 생성된다.
5. 적은 일정 시간 간격으로 헬리콥터 반대편에서 최대 하나의 적이 나타난다.
6. 적은 HP를 가지고 있으며, 헬리콥터가 쏘는 총알에 맞으면 HP가 감소한다.
7. 적의 HP가 0이 되면 적은 사라지고 일정 시간뒤에 다시 생성된다.
8. 적은 움직일 수 있는 공간에서 상하로 계속 이동하며, 일정 간격으로 총알을 헬리콥터 쪽으로 발사한다.
9. 적이 총알을 쏘는 애니메이션을 보여준다.
10. 헬리콥터는 맵에 충돌하거나, 적의 총알에 맞으면 게임오버된다.
11. 게임 화면 가운데 아래 쪽에는 플레이 타임이 상시 표시된다.

3-3 대략 클래스 다이어그램



3-4 전체 시퀀스 다이어그램

캡스톤디자인 결과 보고서



3-5 게임 구현

1. Helicopter Control

WASD 로 헬리콥터 이동, 맵 경계선 밖을 나가지 못하게 입력 예외 처리

맵이 자동 생성되어 왼쪽으로 움직이기 때문에, 왼쪽 이동키를 눌렀을 땐 속도가 증가되어 이동

2. Helicopter Animation

image를 만든 뒤 Vector Container 안에 넣고 한 개씩 순회하여 이미지를 출력하여 헬리콥터의 날이 돌아가는 듯한 애니메이션 구현

3. Helicopter Shooting

Space 키를 누르면 fire 트리거가 활성화되어 Bullet GameObject를 새로 생성하여, bullets

캡스톤디자인 결과 보고서

Vector Container에 저장하여 연사 기능 추가.

4. Bullet

오른쪽으로 계속 움직임, 맵 경계선 밖으로 나가지 못하게 예외 처리
사용이 끝난 총알은 bullets Vector Container에서 제거

5. Map Generator

여러 개의 수직 선(Vertical Line)이 모여 하나의 맵을 구성하는 방식
수직선에 pivot(중심)과 space(빈 공간)를 설정하여 pivot에서 space 만큼의 빈 공간을 위
아래로 생성하며, 그 외는 terrain으로 채움.

Terrain은 총 3가지가 있으며, 바깥쪽은 진하게 안쪽일수록 연한 terrain으로 채움.

수직선은 이전 수직선의 정보를 받아 자신의 정보를 세팅하여 끊임없이 자연스러운 맵
생성.

수직선은 입력한 개수만큼 생성되며, 계속해서 순환되어 재사용 됨 (메모리 단편화 현상
방지)

6. Enemy

적은 20 frame에 한 번씩 **출현하며**, map의 pivot값에 따라 위아래 움직임
helicopter에 연사 기능이 있으므로 Enemy가 자주 생성되도록 난이도 조절
20 frame 마다 총알 발사, 플레이어 총알에 맞으면 없어짐
총알 발사 시, 포신이 줄어들었다가 다시 되돌아오는 애니메이션 추가

7. 충돌 처리

1) 헬리콥터 - 맵

헬리콥터 shape의 채워져 있는 칸의 Position을 모두 저장한 Vector Container를 만듦.

맵의 수직선에 빈 공간 바로 위와 아래 채워져 있는 부분의 Position을 가져옴.

헬리콥터의 채워져 있는 부분과 수직선의 경계 부분의 Position이 서로 같다면 충돌 처리

2) 총알 - 맵

각각의 총알의 Position과 수직선 경계 부분의 Position이 같다면 충돌, 충돌 후에 총알은
사라짐

총알은 오른쪽, 맵은 왼쪽으로 서로 이동하고 있으므로, 뚫고 지나가는 버그가 발생하여
다중 충돌 체크

3) 헬리콥터 - 총알

헬리콥터 shape의 채워져 있는 칸의 position이 총알과 같아진다면 충돌 처리

4) 적 - 총알

적 shape의 채워져 있는 칸의 position이 총알과 같아진다면 충돌 처리

캡스톤디자인 결과 보고서

적의 HP가 0이 되면 없어지고, 일정 시간 뒤에 다시 출현

8. UI

1) time UI

현재 게임 로직 상 Sleep() 함수를 사용하고 있으므로 위의 방법을 사용한다 하더라도 정확한 시간 측정이 어려울 것으로 판단되어, Sleep() 함수의 특성을 이용하여 매 프레임을 0.1초로 보고 프레임마다 시간을 측정

2) HPBar

원하는 크기의 HPBar를 원하는 위치에 세팅하고 HP가 감소하는 기능과, reset 할 수 있는 기능 구현

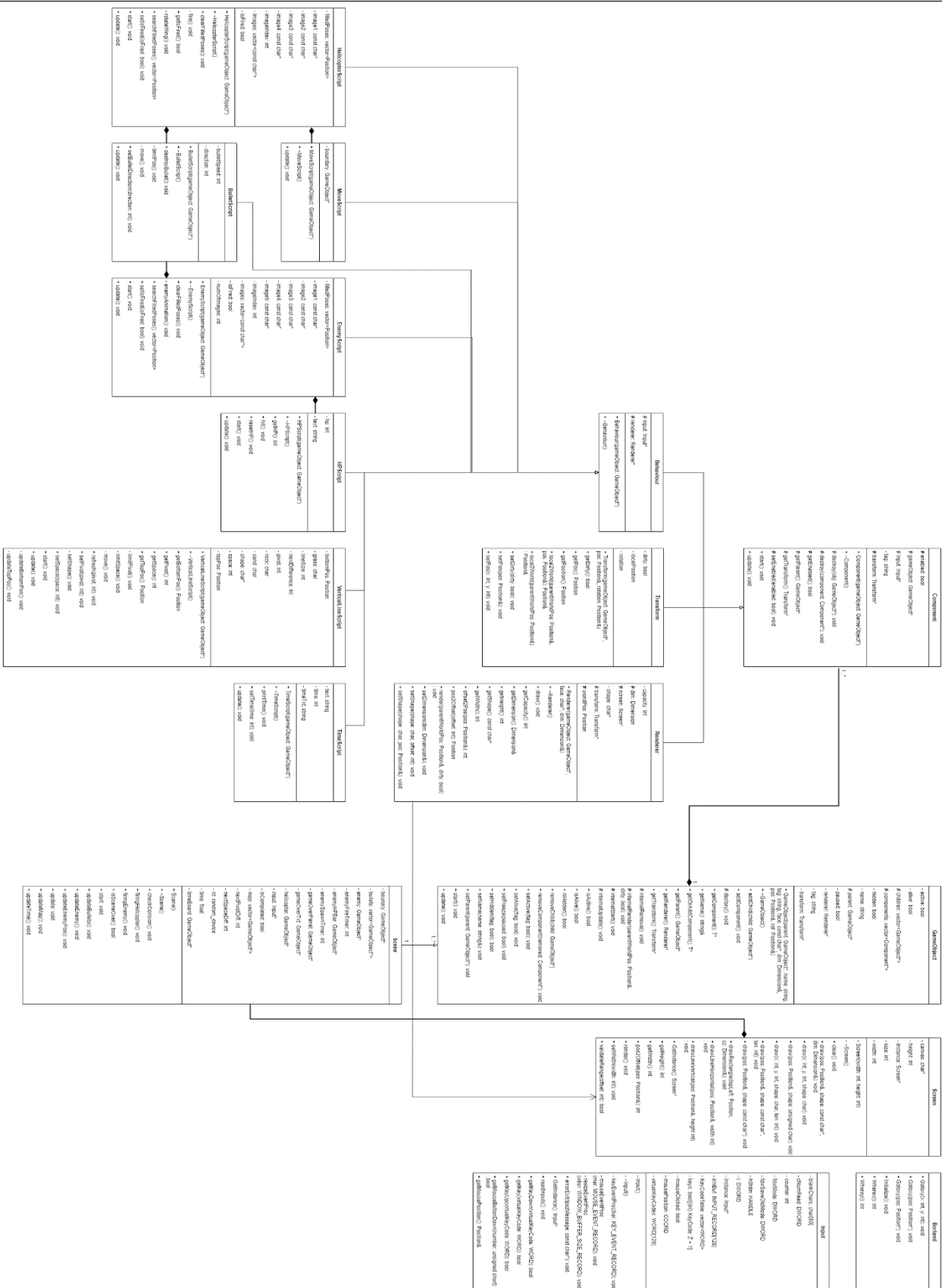
3)Game Over

게임 오버 시 게임이 종료되었음을 알려주는 게임오버 UI 표시

4. 작품 구조도(작품설계, 제작도, 디자인 컨셉 등)

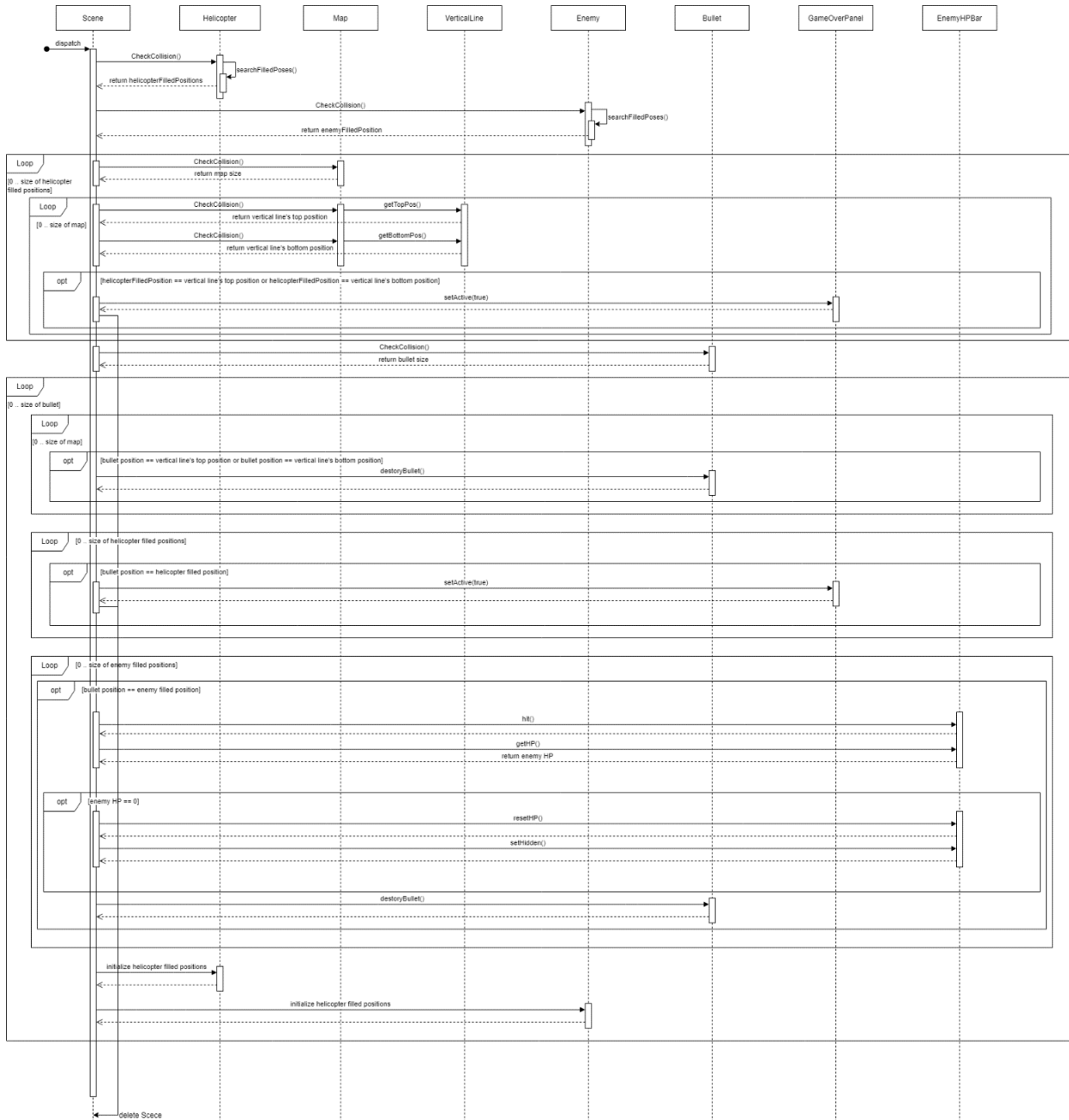
4-1 상세 클래스 다이어그램

캡스톤디자인 결과 보고서



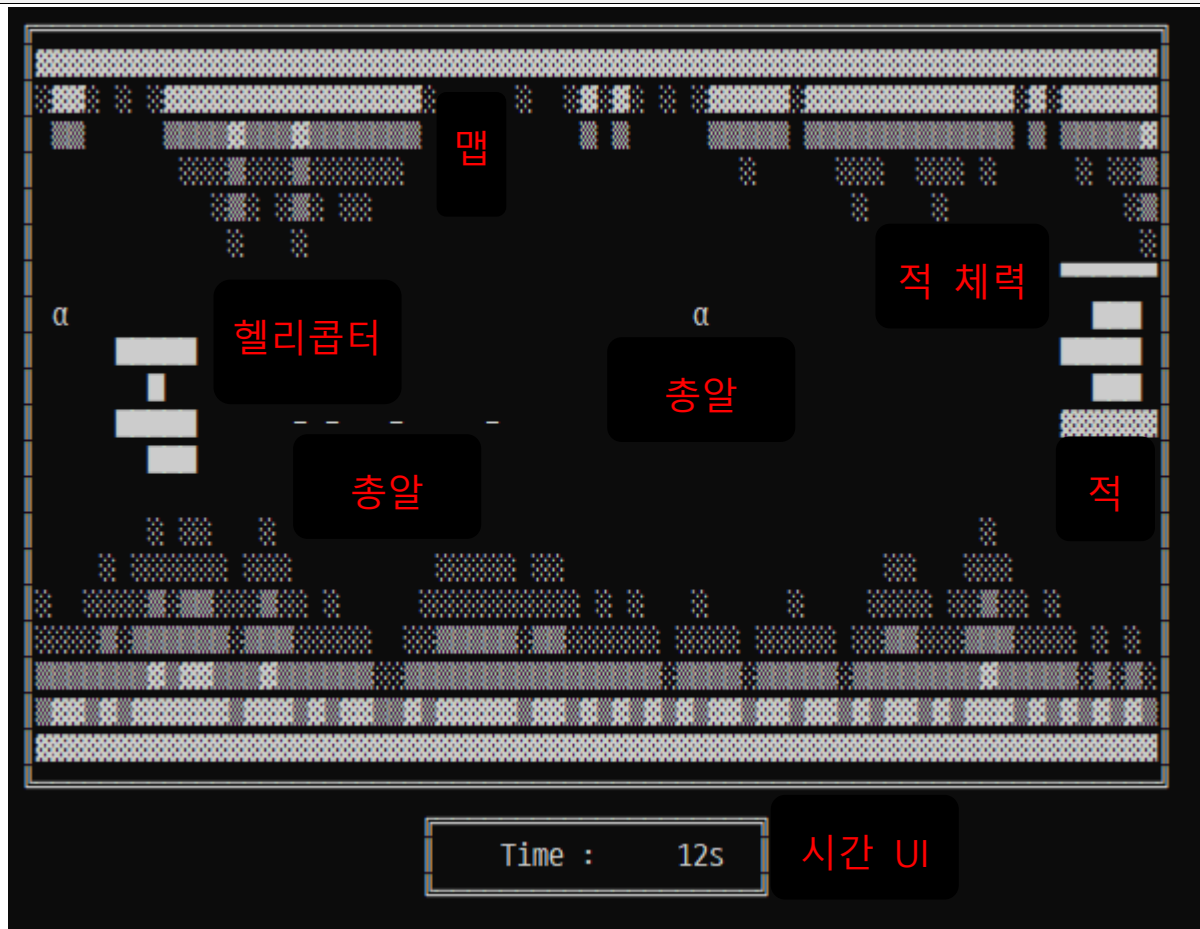
캡스톤디자인 결과 보고서

4-2 충돌 시퀀스 다이어그램



4-3 게임 화면

캡스톤디자인 결과 보고서



5. 기대 효과 및 활용 방안

- 게임 엔진 프레임 워크를 구축하여 게임을 제작하여 게임 엔진에 대한 더 깊은 이해와 활용을 할 수 있게 됨.
- C++ 언어의 숙련도 향상
- 클래스 다이어그램, 시퀀스 다이어그램을 통한 설계로 소프트웨어 개발 효율 및 속도 향상
- 컴포넌트 기반 개발(CBD) 방법 숙달

캡스톤디자인 결과 보고서

2022. 6. 13.

대표학생 방찬웅 (인)

지도교수 (인)