**게임개발자전문가과정**

**과 목 명 : 게임일반프로그래밍**

**능력단위 : 게임알고리즘**

**제출일자 : 2022년 08월 30일**

**포트폴리오 : 옥트리와 충돌판정, 수학, 물리를 이용한 유도탄 구현**

**작 성 자 : 권성호**

**<제출내역>**

1. **게임 알고리즘 프로그램 구현 프로젝트**
2. **게임 알고리즘 프로그램 구현 분석 및 세부 문서**

|  |
| --- |
|  |

**< Contents >**

1. **프로젝트 소개 및 개요**
2. 프로젝트 소개
3. 프로젝트 주요 기술
4. **프로젝트 설계 및 다이어그램**
5. 클래스 다이어그램
   1. 게임 알고리즘 프로젝트 클래스 다이어그램

1-1-1) 게임 알고리즘 프로젝트 클래스 개요

1. 시퀀스 다이어그램
2. **최종 결과 및 추가 내용**
3. 범용성/유연성/확장성/간결성 고려하여 추가된 내용
   1. 현재 프로젝트 설계(구현 내용)
   2. 향후 개발 내용
4. 최종 결과
5. **프로젝트 소개 및 개요**
6. **프로젝트 소개**

본 프로젝트는 공간분할 알고리즘 중 하나인 옥트리 알고리즘을 이용하여 기초적인 수학과 물리를 이용하여 간단한 유도탄 발사체를 제작하였다. 본 프로젝트를 개발하면서 학습한 내용을 복기 와 적용에 초점을 두었다.

앞으로 학습하게 될 수학과 물리를 적용과 게임 애플리케이션의 구조를 제작하면서 앞으로의 학습 방향과 향후 게임 프로젝트의 구조를 다시 한 -번 정리 할 수 있는 기회가 되었다.

1. **프로젝트 주요 기술**

* **공간 분할**

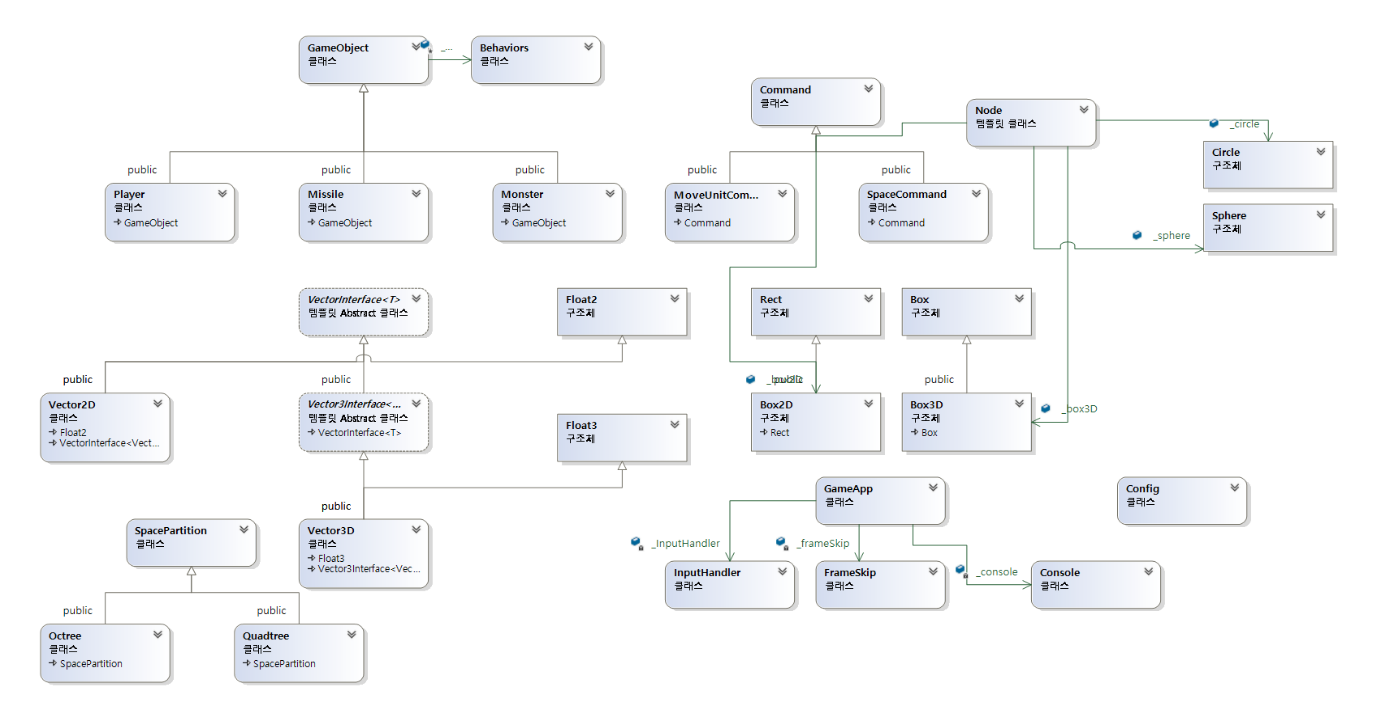
특정한 영역으로 오브젝트를 효과적으로 찾기 위해 공간 분할 알고리즘 중 하나인 옥트리 알고리즘을 사용하였다. 특정 영역을 기준으로 찾고자 하는 오브젝트를 게임 내 오브젝트를 모두 순회하지 않고 해당 영역에서만 오브젝트를 선별하여 효율적으로 추릴 수 있는 기능을 제공한다.

* **유도탄 구현**

수학과 물리를 이용하여 적을 추적하는 유도탄을 구현하였다. 벡터를 직접 코드로 구현하고, 해당 벡터를 이용하여 유도탄 알고리즘을 구현하였다. 해당 프로젝트에서만 이 아닌 많은 게임에서 유용하게 사용할 것으로 보인다.

* **게임 시간의 동기화**

하드웨어의 성능마다 게임 루프가 디바이스 별로 일치하지 않는다. 디바이스 별로 게임 루프의 순회가 다르면 게임의 전체적인 속도가 천차만별일 것이다. 이 점을 해결하고자 게임의 시간을 동기화 하는 프레임 동기화 클래스를 제작하였다.

1. **프로젝트 설계 및 다이어그램**
2. **클래스 다이어그램**
   1. **게임알고리즘 프로젝트 클래스 다이어그램**

[그림 1-1] 게임 알고리즘 프로젝트 클래스 다이어그램

1-1-1) 게임 알고리즘 프로젝트 클래스 개요

|  |  |  |
| --- | --- | --- |
| **부모클래스/인터페이스** | **클래스** | **내용** |
|  | **GameObject** | 여러 게임 오브젝트들의 기본 클래스 공통적인 콜리전, 속도, 방향, 스피드, 위치 등을 가진다. |
| **GameObject** | **Player/Missile/Monster** | 각 게임클래스마다 고유의 게임알고리즘을 가지고 실행 한다. |
|  | **Behaviors** | 오브젝트의 행동을 조정하기 위한 클래스  각 연산들을 조절하여 Force를 구한다. |
| **VectorInterface/ Vector3Interface** | **Vector2D,Vector3D** | 수학 벡터를 이용하기 위한 클래스 |
|  | **SpacePartion** | 공통적인 공간분할 알고리즘 클래스의 메소드를 추상화한 클래스 |
| **SpacePartition** | **Octree,Quadtree** | 높이의 유무에 따라 Octree, Quadtree로 나뉘어짐 |
|  | **Command** | 오브젝트에 명령을 위한 커맨드 클래스 |
| **Command** | **MoveUnitCommand/**  **SpaceCommand** | 방향키와 스페이스키마다 물체의 커맨드를 입력하기 위한 클래스 |
|  | **Node** | 공간 분할 알고리즘에서 이용하는 노드 클래스 |
|  | **Circle/Sphere/Box2D/Box3D** | 충돌처리를 위한 클래스 |
|  | **GameApp** | 게임 어플리케이션 클래스 |
|  | **InputHandler** | 키보드 입력을 위한 클래스 |
|  | **FrameSkip** | 프레임 동기화를 위한 클래스 |
|  | **Console** | 콘솔 출력을 위한 클래스 |
|  | **Config** | 게임의 설정을 위한 클래스 |

1. **시퀀스 다이어그램**

**최종 결과 및 추가 내용**

1. **범용성/유연성/확장성/간결성 고려하여 추가된 내용**
   1. **현재 프로젝트 설계(구현 내용)**
2. 프레임워크 구현으로 프로그램의 확장과 유연함을 확보
3. 하나의 앱 객체에서 모든 객체를 관리가 되어 객체 간의 통신이 가능하게 구현
4. 앱 전체 혹은 씬 별로 계층적으로 객체를 관리하여 객체들의 활성화, 비활성화, 생성, 소멸을 사용자가 주도적으로 제어할 수 있게 구현
5. 파일 입출력을 라이브러리화 하여 다른 프로젝트에서도 사용할 수 있도록 제작
6. 학생관리 프로그램이 별도의 유니크한 키를 레코드별로 발급하여 값을 원자적으로 관리가 가능
   1. **향후 개발 내용**
7. 중복되는 씬 의 기능들이 많아 씬 을 더 추상화 하여 씬의 일반적인 기능별로 부모 클래스 구현
8. 현재 전체 데이터 쓰기, 읽기 구조로 되어 있어 레코드별로 파일에 offset을 저장하여 레코드 별로 값이 저장될 수 있게 구현
9. 특정 씬 의 Update에서 block함수 사용으로 Update가 멈추게 되어있음 이점 개선하여 InputManager와 ConsoleManager를 개선하여 실시간으로 사용자의 입력과 화면을 갱신할 수 있게 구현
10. 현재 각 싱글톤 객체에 접근하여 프로세스를 제어, 더욱 유저 편의적이게 프록시 객체를 생성하여 프록시 객체 만으로 애플리케이션을 제어할 수 있게 구현
11. **최종 결과**

텍스트이(가) 표시된 사진

자동 생성된 설명