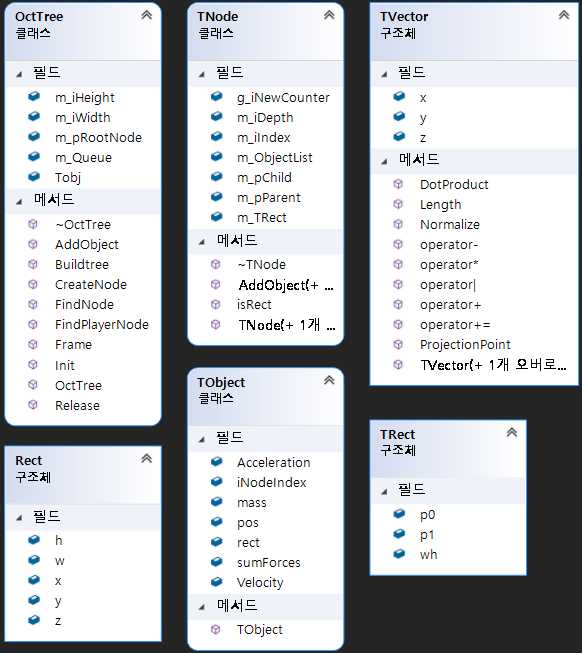
게임 알고리즘 – Check\_7 – 평가일자: 2021년 09월 16일

이름 : 최지원

선택1. Octree가 적용된 지형에서 정적 및 동적 개체의 제외 및 이동 처리를 구현한다.

1. 프로젝트 설계 및 세부적 기능 단위로 다이어그램을 기술한다.

A-1. Octree의 클래스 다이어그램



|  |  |
| --- | --- |
| OctTree- 클래스 | Octree, 팔진트리의 생성, 연결, 탐색, 이동처리, 소멸을 담당 |
| TNode- 클래스 | 노드의 오브젝트 추가, 생성, 소멸을 담당 |
| TObject- 클래스 | 동적 오브젝트의 위치, 질량, 가속을 담당 |
| TVector- 구조체 | X,Y,Z 세 축에 대한 연산자 오버로딩, 벡터의 내적, 투영, 정규화 등 담당 |
| TRect, Rect- 구조체 | 육면체를 표현, x,y,z,w,h / p0, p1, wh |

A-2. Octree의 시퀀스 다이어그램

텍스트이(가) 표시된 사진

자동 생성된 설명A

|  |  |
| --- | --- |
| 1. Init | 전체 팔진트리의 크기 즉, 루트노드의 크기를 지정한다. |
| 1. CreateNode | 부모노드를 받아 노드를 생성한다. |
| 1. Buildtree | 재귀호출을 통해서 전위방식으로 자식노드와 연결한다. |
| 1. AddObject | rand()함수로 xyz에 난수 값을 받아와 오브젝트를 트리에 추가 |
| 1. FindNode | stl 큐를 이용해 레벨오더 탐색으로 해당 노드를 반환한다. |
| 1. AddObject | FindNode로 반환된 오브젝트를 노드 안에 추가 |
| 1. Frame | Window라이브러리 사용함으로 동적인 이동을 구현한다 추후에F=ma 가속도 구현이 가능하다. |
| 1. FindPlayerNode | 동적으로 이동하는 노드 위치를 반환함 |
| 1. Release | 동적 메모리 해제 |

1. 프로젝트의 구현된 주요 기술 단위로 분석 및 설계 문서를 기술한다.
2. Octree의 개요

공간 분할 알고리즘은 실시간으로 렌더링이 될 데이터를 검출하거나 실시간 충돌 데이터를 검출하는 데에 쓰인다. 하나의 게임에 한가지 공간 분할 알고리즘이 적용되는 것이 아니며 여러 알고리즘이 공간에 특성에 따라 적합한 알고리즘이 있다.

공간 분할 알고리즘에는 쿼드트리, 옥트리, BSP트리, K-D트리가 있는데, 그중에서 넓은 공간( 야외 )에서 높은 빌딩들이나 심지어 공중을 날 수 있는 3차원 공간에서는 Octree가 적합하다. 왜냐하면 Octree는 상하로도 객체를 분할하기 때문이다.

2. 주요 변수 설계

<입력, 내부 및 출력 데이터에 대응하는 변수에 대한 구체적인 사항 결정>

|  |  |
| --- | --- |
| Struct TVector  float x,y,z | X, Y, Z 맴버변수, 벡터 변수로 두어 여러 연산자 오버로딩을 통해 구조체끼리 계산이 가능하게 하고, 벡터의 내적, 투영, 정규화등을 계산 할 수 있다.  백터의 내적은 제 2코사인 법칙에 연계되어 a.x\*b.x\*c.x+a.y\*b.y\*c.y+ a.z\*b.z\*c.z이 성립된다. 벡터의 투영은 X와 X 내적분의 Y와 X내적 곱하기 X이다. 정규화는 단위백터를 구하는 과정으로 피타고라스 정리에 의해 나온 길이 V로 나눠주는 것이다. |
| TRect  TVector p0, p1, wh | 박스의 버텍스 변수, 박스의 밑 앞점과 뒷 끝점로 높이와 변의 길의를 통해 한 박스의 형태를 표현 할 수 있다. 정육면체를 전제로 하였다. |
| TObject  Float mass  TVector Velocity  TVector Acceleration  TVector sumForces | 오브젝트 변수, (F=ma) 힘은 가속도 곱하기 질량이다. 속도에 시간을 계속해서 곱함으로써 동적인 움직임을 구현할 수 있다. 추후에 선형보간을 구현 할 수 있다. |
| TNode  Static int g\_iNewCounter | 전위방식으로 받은 동적 할당 카운터 구조체 변수 |
| Octree  TNode\* m\_pRootNode  Int m\_iWidth, m\_Height  Queue<TNode\*>m\_Queue | 루트노드 맴버변수,  초기 루트노드 설정할 너비 높이 변수, 정육면체 전제로 구현하여 두개의 인자.  트리를 레벨 노드 탐색하기 위해 선언한 자료구조 큐. |

1. 주요 함수 설계

<시스템 구성도에서 제시한 모듈에 포함된 클래스나 함수에 대한 구체적인 설계>

TNode\* CreateNode(TNode\* p, float x, float y, float z, float w, float h)

* 부모노드와 위치, 크기를 받아서 새로운 노드, 자식노드를 생성한다. 단, 루트노드는 부모가 없으므로 nullptr 받는다.

Buildtree(TNode\* pNode)

* Buildtree 함수를 여러 번 재귀 호출하여 트리의 전위순회로 노드를 연결, 추가해준다. 2개의 쿼드트리를 붙여 놓은 모양으로 첫 번째 자식 노드의 생성 점을 제일 왼쪽 아래에 두어 반시계방향으로 4개(쿼드 트리), 그리고 다음 자식 노드는 z축으로 p.1의 절반의 값을 넣어서 반시계방향으로 4개(쿼드 트리)를 만들어 4개로 8개 정육면체로 분할이 된 옥 트리를 구현한다.

FindNode(TNode\* pNode, TVector pos)

* 개체를 추가하거나 탐색할 때 쓰는 함수이다. 레벨 순회기법들은 큐를 이용한다. 즉 First-in First-out 속성을 이용하여 먼저 들어온 데이터를 먼저 출력하게끔 하기 위해 큐를 쓴다. 먼저, 큐에 있는 노드를 꺼내어 방문하고 그 노드의 자식 노드를 큐에 삽입하는 작업을 큐에 더 이상의 노드가 없을 때까지 반복하게 된다. (루트 노드 큐에 삽입-> 큐가 empty 하지 않을 때까지 꺼내고 출력하고 자식 노드를 큐에 삽입하는 연산 반복)

Frame(float time)

* 기본 라이브러리Window의 Sleep()함수로 실시간으로 개체의 이동처리를 구현한다. 각 축에 속도를 부여해서 시간 인자값을 받아 계속해서 위치가 변경된다. F=ma, 힘은 가속도 곱하기 질량이다. 가속도 \* 시간 = 속도를 통해서 추후에 선형보간 기능도 추가 할 수 있다.

C. 범용성/유연성/확장성/간결성을 고려하여 추가/변경된 내용을 기술한다.

공간 분할 알고리즘은 주로 실시간 렌더링 데이터 검출과 실시간 충돌 데이터 검출을 위해 사용한다. 그 중, Octree는 Quadtree에서 높이에 대한 분할까지 시도하는 것이다. 따라서, 최적화를 위해서 넓은 공간(야외)이고 날 수 있는 환경에서는 Octree를 적용하고, 평면 하나밖에 없는 환경에서는 Quadtree를 사용하도록, 각 환경에 적합한 알고리즘은 골라 적용하게 병행하는 알고리즘을 추가 할 수 있다.