CSED451 Team Project Proposal Draft

GOLDBERG MACINE

Team: \_\_\_

CSE 20202728 김진수 fusion4268

CSE 20202405 문민재 csmmj4594

CSE 20150221 이화윤 hwayoon2

**Introduction**

이번 프로젝트에서는 골드버그 머신을 Unity를 사용하여 Computer Graphics로 구현한다. 다양한 물체를 이용하여 골드버그 머신을 구현해 봄으로써 물체의 충돌, texture에 대해 이해하고 physics-based animation에 대해 이해할 수 있다.

**Background & Motivation**

1. 골드버그 머신

골드버그 머신은 아주 간단한 일을 하기 위해서 지나치게 복잡하고 간접적인 방법을 하는 머신을 말한다. 관련 없는 물체들의 각각의 동작이 그 다음 동작을 실행시키는 방식으로 구성된다. 따라서 골드버그 머신을 설계하기 위해서는 사용 될 물체의 특성에 대한 이해와 물리 법칙에 대한 이해가 필요하다. 이번 프로젝트에서는 Unity를 사용하여 골드버그 머신을 구현함으로써 physics-based animation과 물리 엔진에 대해 이해할 수 있다.

1. 물체의 충돌

골드버그 머신에서는 주로 물체끼리의 충돌을 통해 한 장치의 동작에서 다음 장치의 동작으로 넘어간다. 충돌을 감지하는 방식에는 AABB, OBB 크게 두 가지가 존재한다. AABB란 Axis Aligned Bounding Box의 약자로 Bounding Box가 축에 나란하게 만들어진다. 따라서 구현이 쉽고 많은 연산이 필요하지 않다. 하지만 실제 물체와 Bounding Box의 차이가 클 수 있다는 단점이 있다. OBB는 Oriented Bounding Box의 약자로 Bonding Box가 축에 나란하지 않을 수 있다. 따라서 충돌을 확인하는 것이 더 복잡하다. 하지만 AABB보다 물체와 Bounding Box의 차이가 크지 않다는 장점이 있다. 이번 프로젝트에서 골드버그 머신을 구현해봄으로써 다양한 Collision Detection 방식에 대해 이해할 수 있다.

1. 물리 엔진

Computer Graphics로 사실적인 골드버그 머신을 구현하기 위해서는 물체가 현실에서 가지는 물리적 특성을 프로그램 안에서 재현해야한다. 물체가 물리적 특성을 가지게 만들기 위해서 Unity에서는 RigidBody를 사용한다. RigidBody가 포함된 물체는 사실적인 움직임을 가질 수 있다. 이번 프로젝트에서 골드버그 머신을 구성하는 물체들이 현실에서 가지는 특성을 반영할 수 있도록 RigidBody의 물리적 요소를 조절해봄으로써 물리 엔진에 대해 이해할 수 있다.

**Project Description**

1. 여러 가지 물리적 case 구현

골드버그 머신은 처음 작동시킬 때 이외에는 설계자 또는 사용자가 별다른 조작을 가하지 않아도 스스로 작동한다는 특징이 있다. 이렇듯 첫 조작 이외에 모든 동작이 스스로 작동하기 위해서는 중력, 마찰력, 관성, 장력 등의 여러 물리적 요인을 고려하여 세심하게 설계해야 하고, 특별한 경우에는 바람, 물 등을 이용한 유체역학이 필요한 경우나 자력을 이용하는 경우도 있다. 이번 그래픽스 프로젝트에서는 최대한 다양한 종류의 물리적 현상을 이용하여 골드버그 머신을 만들어내고자 한다.

1. 다양한 종류의 object 사용

나무, 고무, 플라스틱, 철 등 우리 주변에서 볼 수 있는 다양한 재질로 이루어진 물체들은 서로 다른 물리적 특성(무게, 탄성, 밀도 등)을 지닌다. 그리고, 오브젝트의 모양(구, 원기둥, 정육면체 등)에 따라 구현할 수 있는 액션이 달라질 수 있다.

따라서 다양한 종류의 재료로 이루어진 다양한 모양의 오브젝트를 사용하여 골드버그 머신을 구성하면 머신의 작동 과정에서 더욱 다채로운 모습을 보여줄 수 있을 것으로 기대되기에, 이번 그래픽스 프로젝트에서는 다양한 종류의 material을 이용한 여러 오브젝트들로 골드버그 머신을 구성하고자 한다.

1. 다양한 카메라 액션 구현

골드버그 머신을 구현한 영상들을 살펴보면, 골드버그 머신의 작동 모습을 보여주기 위해 다양한 시점을 사용한다. 머신 전체가 하나의 평면 위에서만 작동하여 하나의 평면만을 비추는 시점이 사용되는 경우도 있고, 반대로 입체적 공간 속에서의 작동 모습을 보여주기 위한 시점이 사용되는 경우도 있다.

카메라의 움직임의 경우에도, 카메라 앵글에 하나의 씬을 담고 카메라의 움직임 없이 씬의 전환만을 통해 골드버그 머신의 진행을 보여주는 경우도 있고, 장치의 역동성을 강조하기 위하여 카메라를 빠르게 이동시키거나 특정 상황을 강조하기 위해 줌인/아웃 기능을 사용하는 등 역동적인 카메라 움직임을 보여주는 경우도 있다.

이번 그래픽스 프로젝트에서는 parallel, perspective projection 등 여러 가지 종류의 프로젝션 기법을 사용하여 다양한 종류의 시점을 구현하고, 정적/역동적 카메라 움직임을 모두 이용한 골드버그 머신을 만들고자 한다.

**Development Plan**

research directions, problem analysis, potential difficulties and possible solutions, references, and so on

**Development Environment**graphics library, UI package, and so on

**Schedule**

including detailed roadmap and milestones

**Role of Team Members**

김진수

문민재

이화윤

**Discussion**

이번 그래픽스 프로젝트에서 Unity를 이용한 골드버그 머신을 구현해 봄으로써 여러 가지 오브젝트들의 collision detection, 오브젝트간의 충돌에서 오브젝트의 재질 및 형태의 차이로 인해 발생할 수 있는 물리적 현상, 그리고 골드버그 장치를 계속해서 작동시키기 위한 물리 기반의 애니메이션 등 현실 세계의 물리적 특징을 반영한 그래픽스를 공부할 수 있을 것으로 기대된다. 또한, 역동적/정적 카메라 액션을 통해 3D 공간에서의 카메라 배치 및 시점의 구현에 대해서도 더 자세히 공부할 수 있는 계기가 될 것으로 기대된다.