3D게임1 과제2 설명 문서

2018182025 이승준

1. 과제에 대한 목표 및 가정

본 과제는 주어진 자동차 모델을 활용하여 트랙을 달리는 레이싱 게임을 만드는 과제이다. 레이싱 게임인 만큼 기존의 롤러코스터 과제, *LabProject14*에서 다룬 *TerrainPlayer,* *SampleProject*에서 다룬 *AirplanePlayer*등과는 비슷하면서도 다른 점이 있다.

기존 수업에서 다룬 모델은 상하좌우 방향키를 통해 플레이어를 이동시킨다. 그러나 이는 레이싱 게임에 그대로 적용시키기에 부적절하다. 왜냐하면 자동차는 왼쪽 또는 오른쪽으로 이동할 수 없기 때문이다. 정확히는 핸들을 틀어 차체가 향하는 방향을 왼쪽 또는 오른쪽으로 향하게만 하고, 직진 또는 후진을 하는 것에 가깝다. 이는 (정확히 1대1로 일치하지는 않으나) 차체의 *Look*벡터를 왼쪽 또는 오른쪽으로 향하게 하고 직진 또는 후진만을 가능하게 하는 것으로 비슷하게 구현할 수 있겠다고 생각했다.

따라서 기존 모델의 좌우 이동을 그대로 적용시킬 수는 없다. 대신 기존 모델에서 *Look*벡터의 이동은 마우스 조작으로 이뤄지기 때문에 마우스를 왼쪽 또는 오른쪽으로 회전시켰을 때의 작용을 방향키 왼쪽 또는 오른쪽을 눌렀을 때의 작용으로 설정하여 자연스러운 차체의 이동을 구현해 보고자 하였다.

다음으로 플레이어가 이동하기 위한 트랙을 만들어야 한다. 트랙 생성에는 필연적으로 하드 코딩이 동원되기 쉽다. 어쨌든 레일의 배치를 디자인해야 하기 때문이다. 그러나 단순히 코드 상에서 레일을 구현한다면 이후 수정 및 관리가 굉장히 어려워지기 때문에 이번 과제에서는 파일 입출력을 이용하여 txt 파일에 레일의 모양을 정해진 규칙에 따라 설정하고, 프로그램이 파일을 읽어 적절하게 오브젝트를 배치할 수 있도록 구현해 보기로 하였다.

레일이나 장애물에는 충돌 체크가 구현되어야 한다. 수업시간에 다룬 바운딩 박스를 활용하여 플레이어와 오브젝트의 충돌을 체크하고, 플레이어 이동 후의 프레임에서 충돌이 검출된다면 이동시키지 않는 것으로 충돌 체크를 구현하였다.

단순히 레일 위를 달리기만 한다면 플레이가 지루할 것이기 때문에, 차량의 급선회를 가능케 하는 드리프트 기능과, 플레이어의 속도를 증가시키는 부스터 기능도 추가 구현해 보기로 하였다. 더불어 부스터를 사용하는 중이라면 장애물을 만났을 때 충돌하지 않고 장애물을 파괴해 버리는 기믹을 추가한다면 재밌을 것이다.

1. 실행 결과와 조작법

다채로운이(가) 표시된 사진

자동 생성된 설명

*(과제 실행 화면)*

플레이어의 이동은 **상하좌우 방향키**를 이용하여 이뤄진다. 상, 하 방향키를 통해 플레이어를 전진, 후진시킬 수 있으며, 좌, 우 방향키를 통해 플레이어가 향할 방향을 정할 수 있다.

***Space***를 누르면서 이동하면 부스터가 발동된다. 부스터 상태에서는 카메라의 시점이 좀 더 앞으로 이동하여 박진감을 더하며, 플레이어의 이동속도가 증가하고, 장애물을 만나면 가로막히지 않고 파괴한다. 돌이나 나무 같은 벽까지 파괴하지는 않으며, 선인장 장애물만을 파괴한다.

***Shift***을 누르면서 회전 시 급선회한다. 이를 통해 드리프트를 구현하였고 트랙 내에 구현된 지름길로 빠르게 접근하는 과정에서 활용될 수 있다.

기타 마우스 조작 등은 사용되지 않으며, 카메라 시점 변경 기능 또한 제거하였다. F5 버튼을 통해 시작 위치로 이동할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

*(플레이어 이동)*

플레이어의 이동시에 이동 방향으로 바퀴가 회전하게 하였고, 부스터 사용 시에는 바퀴의 회전 속도가 빨라진다.

텍스트이(가) 표시된 사진

자동 생성된 설명

*(장애물 파괴)*

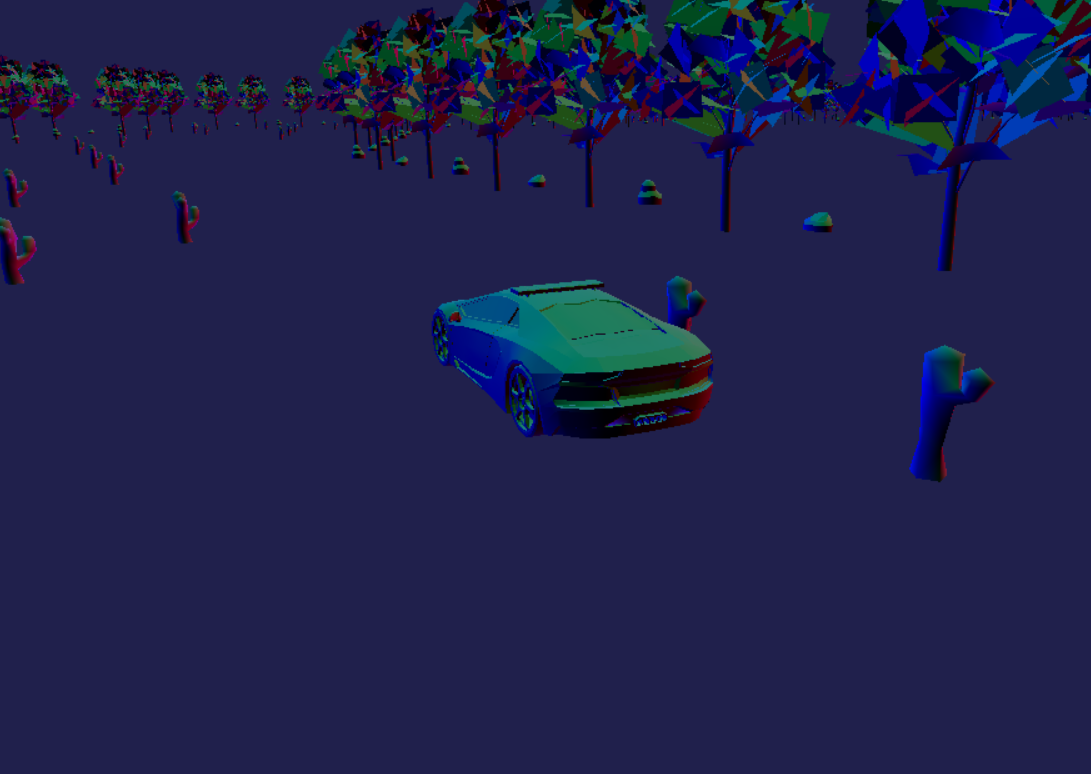
부스터를 사용해 장애물을 파괴하면 파괴 이펙트를 출력하여 자연스러운 효과를 주었다.

텍스트, 다채로운이(가) 표시된 사진

자동 생성된 설명

*(지름길 주파)*

트랙 상의 지름길 구현을 통해 재미 요소를 추가하였다. 이는 파일 입출력을 통해 구현되었으므로, txt 파일 수정을 통해 언제든 바꿀 수 있다.



*(드리프트)*

*Shift* 키를 누르며 선회하면 급선회가 가능하다.

1. 구현 내용

텍스트이(가) 표시된 사진

자동 생성된 설명

*(파일 입출력을 통한 맵 생성)*

맵의 구현은 *‘ObjectData.txt’*를 읽어 이뤄진다. 따라서 맵의 수정 및 관리 등에서 상대적으로 유리하다. 생성된 맵의 정보는 *벡터(vector)*를 통해 관리한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

*(충돌 처리)*

바운딩 박스를 통해 충돌 여부를 확인한다. 이때 충돌하는 오브젝트가 파괴 가능한 지 여부를 체크하며, 파괴 가능하다면 파괴한다. 그렇지 않다면 플레이어가 지나갈 수 없도록 막는다.

텍스트이(가) 표시된 사진

자동 생성된 설명

(폭발 이펙트 구현)

오브젝트가 파괴 가능한 오브젝트일 경우, 오브젝트 내부에 *ExplosionMesh*를 추가로 생성하여 폭파 이펙트 출력에 사용한다.

텍스트, 모니터, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

*(바퀴 회전)*

바퀴의 회전에 플레이어가 향하는 방향을 이용하여 회전 방향을 정하고, 플레이어의 속도를 이용하여 회전 속도를 결정한다.

1. 결론

파일 입출력을 통한 맵 생성이나 부스터 사용시 객체 파괴, 충돌 처리 등 이번 과제를 위해 처음 계획했던 요소에 대해 어느 정도 구현에 성공했다. 그러나 이번 과제를 위해 가장 오래 시간을 할애한 인스턴싱을 통한 렌더링 구현에 실패하여 같은 메쉬를 수십 개 렌더링하는 과정에서 프로그램 실행 시간이 굉장히 늦어지는 문제가 발생한다. 사실상 가장 필요한 기능 중 하나였는데 구현하지 못해 아쉽고, 잘 공부하여 다음 과제부터는 반드시 구현하여야 하겠다.