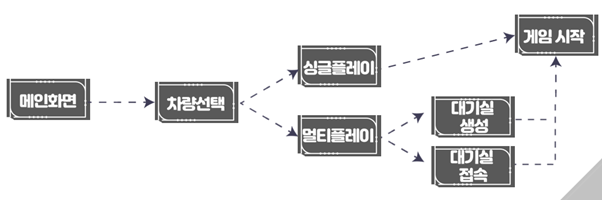
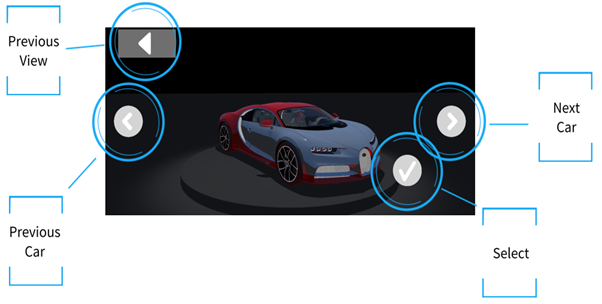


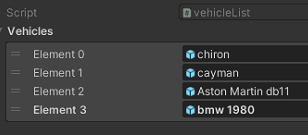
| **프로젝트명** | | 비트레이싱 |
| --- | --- | --- |
| **본인 역할** | | 팀원( 카트 제작 & 미니 맵 제작 ) |
| **개발 개요** | | Unity를 사용해 처음 개발하는 프로젝트 |
| 프로젝트를 통한 Unity 개발 능력 향상 |
| 간단하게 할 수 있는 레이싱 게임 |
| 싱글 플레이와 Phothon Server를 이용해 다른 사람과 함께 멀티 플레이 가능 |
| **개발환경** | 사용언어 | C# |
| 개발 도구 | Photon Server |
| 사용 툴 | Unity, Visual Studio 2019 |
| **프로젝트**  **진행 순서** | 주제 및 개발 목표 선정 | |
| 차량 속성 연구 | |
| 포톤 서버 학습 및 연결 | |
| 프로젝트에 사용 할 Asset 자료 수집 | |
| 싱글 플레이 모드 구현 ( Asset + Car ) | |
| Photon 서버를 이용한 멀티 플레이 모드 구현 | |
| 에러 수정 및 미비점 보완 | |

**<Flow Chart>**

**<Single Play Mode>**

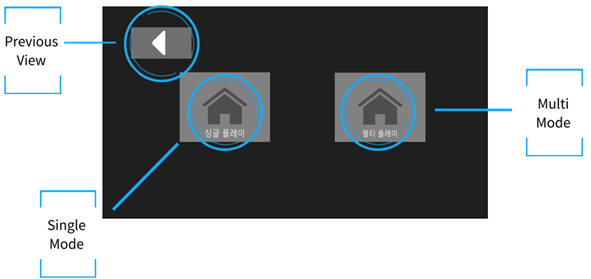
****

* 차량 선택화면
* Car List에 4개의 Car를 추가



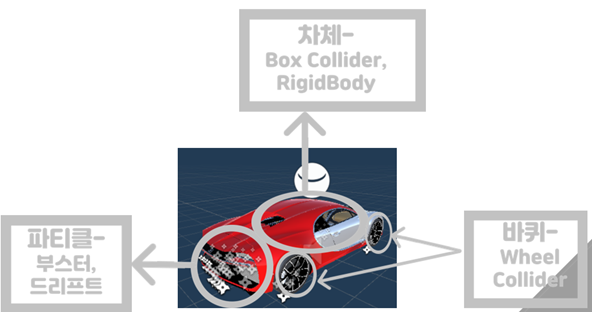
* 원하는 차량 선택 후 Select
* 프로젝트 순서도

**<Select Play Mode>**

****

* Play Mode Choice - Single / Multi

**<Car Function>**

****

* Car body
  + BoxCollider : 충돌감지를 위해 추가
  + RigidBody : 물리효과를 주기 위해 추가
* Particles
  + Booster : 부스터 사용시 이펙트 효과를 넣어주기 위함
  + Drift : 드리프트 사용시 이펙트 효과를 넣어주기 위함

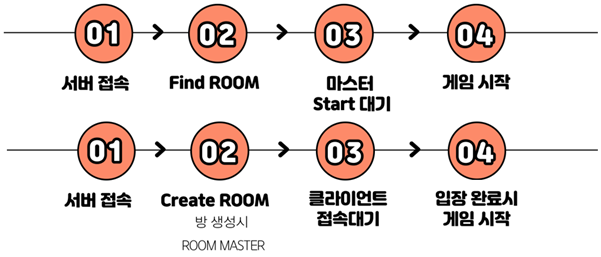
* Wheel Collider - Wheel Collider는 육상 차량용 특수 콜라이더입니다. 내장 충돌 검출, 휠 물리 특성, 슬립 기반 타이어 마찰 모델을 포함합니다. 휠 이외의 오브젝트에도 사용할 수 있지만 특히 바퀴가 있는 차량을 위해 설계되어 있습니다.

**<View Point & In Game Scene>**

****

* Camera
  + 3인칭 시점
  + 속도에 따른 시점과 차량 변화 ( 주행, 부스터, 역주행 )

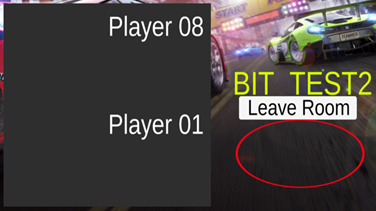
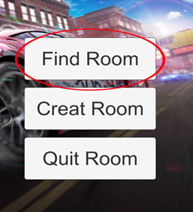
**<Multi Play Mode>**

****

* Photon Server Process

서버에 입장 시 방 생성 ( 생성한 사람이 방의 주인이 된다. )

다른 사용자가 방 찾기를 이용해 생성한 방에 입장을 하게 된다.

****

**<프로젝트 마무리>**

**시행 착오**

1. 카트를 직접 만들려고 하였으나 Wheel Collider에 대한 이해도 부족으로 인해 어려움을 겪음

튜토리얼 및 참고서를 보며 해당 문제 해결.

2. 카트라이더를 생각하며 디자인을 위해 low poly를 쓸려고 했으나 관련된 에셋이 없어

일반 무료 에셋을 참고, 기존 계획과는 달랐지만 해당 에셋에 담긴 내용을 이해하기 위해 시간이 다소 소요

3. PUN에 대해 수업을 토대로한 기본적인 이해도만 가지고 카트를 불러오는데 어려움을 겪음,

강사님의 도움과 참고서를 활용하여 해당 문제 해결.

**느낀 점**

프로젝트 선정 당시 전공자와 비전공자가 같이 진행하는 프로젝트 였기 때문에 당시에 쉽고 간단한 카트 레이싱 주제를 선정했다. 하지만 핵심 기능인 Cart의 구현이 복잡하여 타이어를 구현하는 것에 많은 어려움이 있었다. 1분 30초 가량의 길이와 코스 그리고 이벤트 기능을 고려하는 부분도 있었다. 기획 할 때는 그렇게 어려움이 없을 것이라고 생각했지만, 직접 찾아보고 코딩을 시작하니 디테일한 부분이 너무 많다는 것을 알았다. 사용자와 개발자의 시점은 확실히 다르다는 것을 느낄 수 있는 계기가 되었다.