**이 력 서**

**개인신상**

성 명 : 이준희 (한자 : 李俊熙 / 영문 : Lee Joon Hee )



생년월일 : 1991년 04월 16일

주 소 : 서울특별시 성동구 금호동 동호로 100 두산아파트 108동 1003호

핸 드 폰 : 010 – 4874 – 6269

E-Mail : dreamrunner416@gmail.com

**학력사항**

2010.03~2020.02 경영정보학과(MIS) (20년 2월 졸업예정) **충북대학교**2007.03~2019.02 자연계 **서울고등학교**

**Unity 프로젝트**

|  |  |
| --- | --- |
| **기간** | **내용** |
| 2019.05  ~  2019.10 | **–** PNC솔루션 센서를 활용한 초등학생용 스크린 스포츠 게임 (2019 KVRF 참가\_코엑스) |
| -Vive를 활용한 정찰 및 수송드론 운전 연습 시뮬레이션 VR |
| - Vive를 활용한 드론 운전 교육 VR |
| - 구글 데이드림과 Photon네트워크를 활용한 온라인 탁구 게임 어플 |
| - 구글 AR Core를 활용한 엔진 분해 및 수리 체험 어플 |
| - Unity를 활용한 Greedy Snake게임 앱스토어 출시 |

**수상경력 및 장학사항**

|  |  |  |  |
| --- | --- | --- | --- |
| **연월** | **이력** | **입상내역** | **주관기관** |
| 2014.11 | 창업아이템 경진대회 | **대상** | 중소기업청 |
| 2017.01 | Global Brain Park  스타트업 캠프 | **Excellence Prize** | Stone Wealth & Global Local Brain Park |
| 2017.05 | 충북 스타트업 아이템 경진대회 | **우수상** | 충북대학교 |
| 2017.11 | 캡스톤디자인 UCC  경진대회 | **최우수상** | 충북대학교 |
| 2017.12 | 충북대학교  창업경진대회 | **장려상** | 충북대학교 |

**자격사항**

2012. 07. 17. TOEIC(798) YBM

2013. 11. 03. TOEIC SPEAKING (LEVEL 7) YBM

2014. 09. 22. 비지니스번역(영어)자격증  **국제통번역협회**

**2019년 혁신성장 청년인재 집중양성사업 이수**(VR/AR/MR 융합 콘텐츠 개발 실무 프로젝트 과정)

|  |  |  |
| --- | --- | --- |
| **기간** | **내용** | **주관** |
| 2019.05.16  ~  2019.11.07  (종료예정) | 기획 워크숍(아델리)  - 프로그램 개발을 위한 기획 프로그램 | **메디치 교육센터**  **(가산디지털역)** |
| Unity를 위한 C# 프로그래밍(최문석 강사님) |
| Unity 기본 프로그래밍 (박승제 강사님) |
| Unity VR 프로그래밍 (이재현 강사님) |
| Unity AR 프로그래밍 (이재현 강사님) |
| Photon을 이용한 Unity 네트워크(봉신영 대표님) |
| HTC VIVE를 이용한 VR 프로그래밍 (이재현 강사님) |
| 기업연계 프로젝트 |

**인턴 및 경력사항(아르바이트 포함)**

|  |  |  |
| --- | --- | --- |
| **기간** | **직무** | **업체명** |
| 2018.01~2019.02 | 대표, 크래프터 | 막리공작소 |
|  | - 소품 및 가방 디자인  - 사업기획 및 사업기획서 작성  - 네이버 쇼핑몰 관리, 해외 쇼핑몰 관리(ETSY)  - 직원 관리  - 외부 출강  - 재무관리 |  |
| 2016.09 ~ 2017.12 | 팀장 | 푸디(FooDIY) |
|  | - 웹 플랫폼 서비스 기획  - UX/UI디자인 (카카오 오븐 활용) |  |

**Skills**

|  |  |  |  |
| --- | --- | --- | --- |
| **개발관련** | **숙련도** | **비개발** | **학교** |
| Unity | 중 | 포토샵 | 중 |
| C# | 중 | 프리미어프로 | 중 |
| Java | 하 | 일러스트 | 하 |
| GitHub | 중하 | 파워포인트, 워드, 한글 | 중상 |
| SVN | 하 | 엑셀 | 하 |

**교내. 외 동아리활동**

|  |  |  |
| --- | --- | --- |
| **기간** | **동아리명** | **학교** |
| 2016.09 ~ 2017.12 | 푸디 (집밥 에어비앤비 창업 동아리) | 충북대학교 |
|  | - 스타트업 동향분석  - 팀원관리  - 사업기획 및 사업 계획서 작성  - UI/UX 디자인  - 재무관리  - 창업 지원비 획득 및 관리 | 구성 : 경영정보학과(3인),  컴퓨터공학과(4인) |

**자기소개서**

**개발 직무 지원동기**

**<개발자 직무에 대한 적성을 발견 및 쿠키런 킹덤의 새프로젝트 참여 희망>**

28세가 될 때까지 소품디자인, 웹 앱 기획, 제품 판매 및 영업 등 다양한 경험을 하다가 올해 5월부터 정부지원 **직업훈련 교육으로 Unity게임 개발자 과정을 이수**하게 되었습니다. 기존에 해보았던 분야가 아니어서 처음 개발을 배울 때는 이해도 어렵고 적응도 어려웠지만 문제를 하나씩 해결하는 과정에서 즐거움을 느꼈습니다. 또한 새로운 지식을 이해하는 속도가 빨라짐에 따라 더 큰 즐거움을 느꼈습니다. 학원 내에서 본인들이 만든 게임을 서로 해보며 경쟁하고 새롭게 업데이트하거나 시간 가는 줄 모르고 버그를 수정하고 있는 제 모습을 보고 개발 직무가 저에게 맞는 다는 것을 확신했습니다.

마지막 프로젝트를 개발 하던 중 취업을 위해 채용정보를 알아보던 중 데브시스터즈에서 쿠키런의 새로운 프로젝트를 위해 Unity개발자 직무를 구인 하는 것을 보았습니다. 10년간 쿠키런으로 대중들에게 사랑 받던 게임이 SNG스타일을 가미해 새로운 스타일의 게임을 만드는 정보를 접하고 이 프로젝트에 참여하고 싶다는 마음이 생겼습니다. 유투브를 통해 쿠키런 킹덤 스틸영상을 보고 더욱 기대가 되어 본사에 방문을 하고 싶다는 이메일도 넣어 봤으나 거절되어 이렇게 이력서를 작성해 입사를 신청합니다.

**본인의 장단점**

**<장점 1 : 빠른 습득력 >**

본인은 소품 디자인, 기획, 창업, UX/UI디자인 등을 경험하고 수상경력이 있습니다. (이력서 뒤의 직접 작성한 연혁표를 참조해 주세요.) 짧은 기간 동안 많은 경험을 하게 된 이유는 어려서부터 하고 싶은 일을 해본 적이 없었기 때문입니다. 그래서 단기간에 남들보다 많은 경험을 해야 한다는 생각에, 빠르게 기술을 익히고 습득하려고 했습니다. 이러한 저의 과거는 현재의 저에게 영향을 주어 빠르게 익힐 수 있게 해주고 있다고 생각합니다. 19년 5월에 처음 접한 C# 과 Unity였지만 여러가지 프로젝트를 만들어볼 수 있게 된 것은 빠르게 습득하려는 제 성격이 일부분 도움을 줬다고 생각합니다.

**<장점 2 : 기간내에 문제를 해결하는 집중력과 끈기>**

Unity 게임 개발자 과정을 이수하면서 마지막 과제로 기업과 협업을 한 프로젝트를 진행할 때(19년 09월 11일)였습니다. 국방VR개발을 주력 상품으로 가진 PNC솔루션이라는 기업과 협업 하였는데, 이 업체는 초등학교 실내스포츠교실 시장에 진입하기 위해 스크린스포츠용 카메라 센서를 개발 중이었습니다. Unity를 이용해 스크린 축구 게임 콘텐츠를 개발해 달라는 요청으로 게임 개발계획서를 작성하고 개발에 착수했습니다. 특이사항으로 10월 02일부터 삼성 코엑스에서 열리는 KVRF에 전시가 될 수 있도록 요청하셔서 2주의 개발기간과 1주간 버그를 수정 후 출시하려고 했습니다. 그러나 두가지의 어려운 상황이 생겼습니다.

첫 번째는 해당 하드웨어의 개발언어와 Unity언어가 달라서 TCP통신으로 데이터를 받아야 해서 유니티용 Client를 개발하는 **네트워크 문제.** 두 번째는 하드웨어 개발 **딜레이**와 하드웨어 프레임 제작의 딜레이로 인한 **테스트 불가**.

**네트워크**에 대한 기본 상식이 없는 상태에서 클라이언트를 개발해야 했기에 처음에는 막막 하였으나 네트워크에 관련된 서적들을 찾아보고 블로그에 개념을 정리해 나갔습니다. 10일여동안 서버와 클라이언트를 만들어서 채팅창 구현을 해보기도 하면서 유니티에서 사용할 수 있는 클라이언트 코드를 개발했습니다. 네트워크개념에 대한 이해와 코드구현의 실패가 지속됨에 따라 포기하고 싶은 마음이 커졌었으나 그럴 때마다 마음을 다잡고 한번 더 구현해 보고 서치해 보면서 개발에 성공했습니다.

개발의 가장 큰 어려움은 두번째 문제인 **하드웨어 개발의 딜레이**였습니다. 외부 환경에 의한 개발의 어려움은 제어가 불가능한 요소였습니다. 이를 극복하기 위해 직접 강원도 원주에 있는 연구소에 내방해 게임 테스트를 해보기도 했습니다. 하지만 버그를 모두 잡기에는 시간이 부족하였고 KVRF를 진행할 때 실시간으로 버그를 수정하여 전시에 지장이 없도록 했습니다.

어려운 환경에서도 집중력을 발휘해 기간내에 문제를 해결하는 집중력과 끈기는 팀프로젝트를 진행할 때에 차질없이 완료할 수 있는 장점이라고 생각됩니다.

**<단점 : 다양한 개발 경험 부족>**

비록 교육기관의 Unity개발자과정에 국한된 경험이었지만, 크고 작은 개발 프로젝트들을 진행해 보면서 전반적으로 개발에 필요한 업무들을 두루 익혀볼 수 있었습니다. 이를 통해 정해진 일정에 맞춰 기획하고 구현하고 디버깅을 하는 과정이 얼마나 어려운지도 알게 되었습니다. 프로젝트를 진행 할 시 우선 발생할 수 있는 리스크들을 리스트업 하고 이것을 관리 할 수 있는 솔루션을 마련해 일정을 만들어야 함을 알게 되었습니다. 교육기관에서 진행한 프로젝트들로부터 얻은 교훈들은 앞으로의 실무 개발환경에도 좋은 밑거름이 될 수 있을 것이라 생각합니다.

**포트폴리오**

**기업연계 프로젝트 – PNC솔루션 센서를 활용한 초등학생용 스크린 스포츠 (2019 KVRF 참가\_코엑스)**

|  |  |  |
| --- | --- | --- |
| **개발 기간** | 2019.09.11 ~ 10.04(기획 20일, 개발 48일) | |
| **개발 환경** | Unity 1.10f / PNC솔루션 센서 / GitHub | |
| **개발 인원** | 2명 | |
| **요 약** | PNC솔루션에서 개발하고 있는 스크린센서와 위쪽 옆쪽의 카메라 센서를 활용해 초등학생들이 즐길 수 있는 스크린 실내 스포츠 컨텐츠. 2019 KVRF(코엑스 전시)를 위해 패널티킥 게임 1개를 개발함. | |
| **기능** | **1) TCP통신 클라이언트**  **2) 골 판정**  **3) 골, 노골 세레머니**  **4) 슛을 했을 때의 공 움직임**  **5) 스크린 좌표에서 3D UNITY좌표로 변환** | |
| 2019 kvrf에 대한 이미지 검색결과 | |  |
|  | |  |
|  | |  |
| **1. 개발 기획 (WBS활용해 팀원과 일정을 공유 했음.) Link : shorturl.at/eh127**    **1.1 PNC솔루션으로부터 요구사항 :** TCP통신 클라이언트를 활용해 센서로부터 값을 받을 것, 초등학생 저학년들이 할 수 있는 스크린 축구 게임을 개발할 것**, 10월02일 코엑스에 전시 할 것.**  **1.2 타사 아이템 분석**    삼성 코엑스에 열린 초등교육 박람회와 강남의 레전드 히어로즈를 방문해 스크린 스포츠 시장의 타사 아이템들을 경험해 보고 분석.  **차별점 : 이용자의 연령대를 고려해 난이도를 낮추고 기능을 간소화 하여 저학년들이 스포츠교실로 활용할 수 있도록 교육에 목적성을 둠.**  **1.3 마인드맵을 이용해 필요한 기능과 로직 분석.**      **1.4 BPMN을 활용한 사용자 시스템 분석 및 설계**      **2. Screen Soccer UML**    **2.1 ClientDeRecode1 - 센서의 값을 받을 TCP클라이언트 클래스**   |  |  | | --- | --- | | **Code** | **해설** | |  | **[서버 연결요청]**  1. 연결하고자 하는 서버의 IP주소와 Port번호를 전역변수에 지정하고 TCP프로토콜로 Connect요청을 한다.  2. 연결요청은 게임 시작 시 F11버튼을 눌러 요청한다.  3. 소켓의 Stream을 받아온다. | |  | **[스트림 안의 필요한 버퍼 받기]**  1. 스트림안의 데이터를 streamByte배열에 Read메소드로 저장한다.  2. streamByte배열의 값 중 16진수로 “70”인 버퍼를 찾기.  70이 아닌 경우 스트림 비우기.  3. byteToArray리스트에 버퍼 복사하기.  4. DecodeByte()로 해독하기. | |  | **[버퍼 해독하기]**  - 미리 약속된 규약대로 해독.  1. 70~77까지 카테고리 에 따른 데이터 해독.  2. 버퍼에서 카테고리 1바이트, 타입 1바이트를 제거하기.  3. 센서의 값이 있는 다음의 4바이트를 합쳐서 원래의 값으로 변환.  4. 다른 스크립트에서도 센서의 값을 사용할 수 있게 구조체로 초기화. PNCVariable()  5. 센서의 값이 들어왔음을 알리는 StartBall()로 GameManager 제어. |   **2.2 GameManager – 게임 제어 클래스**   |  |  | | --- | --- | | **Code** | **해설** | |  | **[인트로 씬 보여주기 및 환경설정 셋팅]**  1. Interface IStartIntro()실행.  -> 인트로씬 실행. | |  | **[IStartIntro()]**  2. 애니메이션이 들어 있는 OverViewCamera를 활성화시키기.  3. 애니메이션의 종료시간을 계산해 다음 프로세스를 StartCoroutine으로 실행. | |  | **[IPenaltyKickStart()]**  4. 카메라를 1인칭으로 돌아오게해 플레이를 할 수 있게 변경.  5. 센서로부터 값을 받을 수 있게 됨. (actionable =true;)  6. 1게임당 공차는 시간을 제약하는 카운트다운 기능을 넣고 싶다면 ICountDown()주석 제거. | |  | **[StartBall()]**  7. 센서로부터 값이 들어 오면 게임이 플레이 됨. (ClientReDecode1로부터 호출)  HJ\_SoccerPlayer의 StartShoot()실행. |   **2.3 HJ\_SoccerPlayer- 센서값을 이용한 슛팅 클래스**   |  |  | | --- | --- | | **Code** | **해설** | |  | **[StartShoot()]**  1. 센서값 UscreenX, UscreenY, Uvelocity를 private 전역변수에 저장하고 값을 보호해 사용. | |  | **[ScreenToUnity()]**  2. 하드웨어 업체에서 규약한 스크린 좌표값을 유니티로 넘어 왔을때 Unity좌표계로 변환. | |  | | |  | **[BallisReady()]**  7. 센서로부터 값이 들어오면 게임이 플레이 됨.  (ClientReDecode1로부터 호출) | |  | **[MovingBall()]**  8. 유니티 물리 엔진을 사용하는 AddForce메소드로 공을 목적지까지 힘을 가해 이동시킨다. 실제 공을 차는 느낌을 주기 위해 add\_LEFTx, add\_RIGHTx, addY변수를 이용해 게임상의 공의 움직임을 보간 한다.  특이사항 : 실제 센서를 이용했을 경우 PC로 테스트했을 때의 값이 달라서 다음과 같은 변경을 함.  1) 공의 목표지점이 카메라의 정 중앙을 기준으로 왼쪽과 오른쪽일 때의 값이 달라져서 add\_LEFTx와 add\_RIGHTx값으로 보정.  2) 카메라의 정중앙부터 X축으로 add\_RIGHTx까지의 값을 보정하기 위해 코드를 수정. | |  | **[GoalCheck()]**  1. 아래의 그림처럼 골대안의 Goall이 될 수 있는 지역(기즈모로 표시)을 Goal이 될 수 있는 구간으로 정함.  2. 해당 구간을 공이 생성되어 움직일 때 if문으로 순차적으로 판정하여 Goal여부를 판정. | |  | |   **기능 :**  **1) TCP통신 클라이언트**  해당업체에서 만든 **하드웨어의 언어와 유니티의 언어가 달라서 TCP통신 클라이언트를 개발함.** 센서의 값을 패킷으로 받아 필요한 값을 유니티에 맞게 변환시켜 게임을 진행하게 함. 패킷의 헤더 값을 검출하고 업체가 규약해준 표에 따라 각각의 바이너리 값 뒤의 데이터를 X값, Y값, 속도, 스핀 값 등으로 Decode하여 Unity에서 사용할 수 있게 함. 해당 값들은 구조체로 만들 줌. 새로운 값을 하드웨어로부터 받을 때, 구조체 초기화와 함께 Static 형으로 만든 StartBall() 메소드를 실행.  **2) 골 판정**  공이 골대에 들어갈 때 골대에 들어가지 않을 때 그리고 골대를 맞고 그라운드에 있을 때 3가지로 골 판정을 분류. 골대의 X, Y, Z축을 기준으로 삼아서 세가지의 경우의 수로 나눔. 축구공이 특정위치에서 골 판정을 할 수 있게 함.  **3) 골, 노골 세레머니**  골이 들어가거나 들어가지 않았을 때에 따라 특정 애니메이션 플레이가 됨.  **4) 슛을 했을 때의 공 움직임**  플레이어가 슛을 하면 센서에서 받아들인 공의 속도로 게임상에서도 날아 갈 수 있도록 물리엔진으로 공이 나갈 수 있도록 하여 실전과 비슷한 느낌이 들도록 함.  Addforce메소드로 공이 물리엔진 힘을 받을 수 있게 하고 Rigidbody Use Gravity로 공이 자연스럽게 중력을 받을 수 있게 함. 센서로부터 받은 스크린 좌표를 카메라의 좌표로 변경 후 Z값을 임의로 설정하여 목표지점을 만들어 줌. Z값을 공의 Velocity에 비례하여 만들어 줄 경우 실제로 공을 차는 플레이어가 가상게임과 거리감을 느꼈음. 그래서 오히려 임의의 Z 값을 줘서 본인이 스크린에 찬 위치에 비슷하게 궤적을 생성할 수 있도록 하는게 플레이어의 재미를 더 함.  **5) 스크린 좌표에서 3D UNITY좌표로 변환**  공이 스크린에 맞았을 때 해당 스크린의 좌표는 2D이고 유니티상의 좌표는 3D여서 좌표의 변환이 필요했음. 스크린 좌표계(왼쪽 상단이 (0,0), 우측 하단이 (1920, 1080))를 유니티 좌표계(왼쪽 상단이 (0, 1080), 우측 하단이(1920, 0))로 변환함. | | |
| **맡은 역할** | 1. 메뉴 UI만들기  2. TCP통신 클라이언트  3. 골 판정  4. 스크린 좌표에서 3D UNITY좌표로 변환  5. 게임매니저 | |
| **프로젝트를 진행하면서 어려웠던 점** | **1. 예상할 수 없는 외부 요인으로 인한 일정 딜레이.**  - 하드웨어 설치의 일정이 늦어 짐에 따라 개발 일정을 늦추거나 수정을 해야 하는 부분이 생겨 코엑스 KVRF전시의 출품에 어려움이 있었음. 전시회 전에 직접 강원도 원주의 연구소에 찾아가 테스트를 해보았으나 충분한 테스트 시간이 없었음. 하지만 전시회를 여는 당일 아침 일찍 도착하여 프로그램을 수정하고, 전시를 진행하며 시연을 원하시는 손님들이 시연했을 때 나온 오류를 바로 수정하여 업데이트함. 시연을 하는 손님들이 많아서 다양한 오류를 한번에 수정하기 용이했음.  **2. 해본 적 없는 기술을 익혀야 하는 점.**  - 요구사항에 따라 네트워크 기술로 하드웨어와 게임을 연결해야 했는데 이 기술은 한번도 해본 적 없는 기술이라 어려움이 있었음. 하지만 포기하지 않고 연구하고 개발을 시도한 결과 필요에 맞는 기능을 개발함. 이에 앞으로도 미지의 개발 영역을 공부하는 것에 자신이 생김. | |
| **깨달은 점** | **1. 프로그램 테스트 시에 오류가 날 만한 부분을 미리 체크하고 수정할 준비를 해야 함.**  **2. 외부의 환경으로 예상된 일정들이 변경 될 것을 감수 하고 개발 계획을 만들어야 함.**  **3. 협업을 위해 코드마다 자신이 어떤 의도로 어떤 기능의 코드를 만들었는지 주석으로 알릴 것.** | |

**국방 프로젝트 – PC 및 Vive를 활용한 정찰 및 수송드론 운전 VR국방 프로젝트**

|  |  |
| --- | --- |
| **개발 기간** | 2019.07.22 ~ 08.06(교육VR기획 및 개발 2일, 국방VR 기획 및 개발14일) |
| **개발 환경** | Unity 1.10f / VIVE / GitHub |
| **요 약** | 개발 인원 : 3명  4차산업혁명으로 통신 및 드론의 발달로 사람이 하기 어려운 정찰이나 수송을 군에서는 드론이 대신 작업을 한다고 함. 군에서는 이러한 사회변화에 드론조종병을 착출하기도 하고 훈련 하는 중. 이에 국방 융합 프로젝트로 드론으로 정찰과 수송을 연습할 수 있는 프로그램을 만들어보고자 함.  **기능 :**  **1) 적진 정찰**  드론 조종기로 드론의 카메라를 상하 좌우로 움직여 드론의 시야를 확보 할 수 있음. 확보한 시야는 조종기의 패널로 볼 수 있음. 또한 현재 위치를 알 수 있는 미니맵이 오른쪽 하단에 있음. 적진을 발견하게 되면(일정 거리 안에 적진이 포착되면) 위험 경고 소리가 조종기에서 들리고 미니맵에 적진이 나타남.  **2) 수송**  드론에 달린 줄로 수송할 물자를 연결하면 목적지까지 운송 할 수 있음. 수송 물자가 지정한 위치에 도착하면 미션 완료를 알 수 있는 이펙트 발동.  **3) 드론 콘트롤러를 잡았을 때 손의 애니메이션**  프로젝트에 사용된 드론 콘트롤러는 에셋 이었으므로 바이브에 입힌 손 애니메이션 중 맞는게 없었음. 그래서 콘트롤러에 맞게 손 애니메이션을 수정함. 콘트롤러를 인식하고 잡게 될 경우 자연스러운 모션으로 잡을 수 있게 함  **4) 튜토리얼 만들기**  수송 미션과 적진 정찰 미션을 쉽게 할 수 있도록 안내하는 튜토리얼을 제작. 바이브를 처음 사용해도 조작 할 수 있도록 바이브의 어떤 부분을 눌러야 하는지를 알려주고 해당 버튼을 클릭해야 다음 튜토리얼로 갈 수 있음. 튜터리얼을 위한 패널과 이미지 등은 ITween코드를 활용함. 보다 동적이게 보이게 하며, 보수를 쉽게 할 수 있도록 ITween코드로 패널의 동작과 사라짐 등등을 제어함.  **5) 바이브 인터페이스 환경 조성**  팀원들이 바이브의 버튼 기능을 바로 사용할 수 있도록 바이브의 버튼 조작 환경을 조성함.  **6) 텔레포트**  Unity 게임상에서 바이브 콘트롤러를 이용해 이동할 수 있는 텔레포트 구현. 콘트롤러에서 LineRenderer를 생성하여 Ray의 위치를 알 수 있고, 바닥 콜라이더에 맞아 Aim이 생기고 콘트롤러의 패드를 클릭하면 해당 위치로 이동할 수 있음. |
| |  |  | | --- | --- | |  |  | |  |  | |  |  | |  |  | | |
| **맡은 역할** | 1. 메뉴 UI만들기  2. 적진 정찰 튜토리얼 만들기  3. 수송 튜토리얼 만들기  4. 바이브 핸드애니메이션  5. 바이브 인터페이스 환경 조성  6. 텔레포트 |

**구글VR디바이스 - 구글 데이드림과 Photon네트워크를 활용한 온라인 탁구 어플리케이션**

|  |  |
| --- | --- |
| **개발 기간** | 2019.07.04 ~ 07.10 (기획 및 개발 7일) |
| **개발 환경** | Unity 1.10f / 데이드림/ GitHub / Galaxy S9 |
| **요 약** | **개발 인원** : **2명**  데이드림과 Photon Pun을 활용한 온라인 탁구 게임. 데이드림의 콘트롤러의 모션 인식으로 유니티상의 탁구채를 휘둘러 공을 튀길 수 있고 온라인으로 탁구 게임을 할 수 있음.  **기능 :**  **1) Photon Network**  Photon사의 Pun을 활용하여 20명까지 무료로 온라인 접속할 수 있는 게임 환경 조성. 온라인으로 접속한 플레이어는 탁구 게임을 할 수 있음. 플레이어의 움직임을 Photon View로 동기화하고 공의 움직임은 Photon Serialized View로 직접 코드를 입력하여 실시간 동기화 하였음. Serialized View코드를 이용하지 않고 PunRpc 코드나 Photon View를 인스펙터로 추가할 경우, 상대방 움직임에 딜레이가 발생함. 그렇기 때문에 동기화가 빨리 잘 되어야 하는 탁구 게임의 공 같은 경우는 Serialized View코드를 사용하였음.  **2) 공의 바운스**  공에 Bounce속성을 줘서 현실과 비슷한 탁구공의 탄성을 구현.  **3) 탁구채**  탁구채에 휘두르는 속도에 따라 공에 힘을 가하여 속도를 낼 수 있음. 빠르게 탁구공을 가격할 경우 실제 스매시와 비슷한 효과를 볼 수 있음.  **4) 탁구게임 점수 판정**  탁구 공이 테이블을 맞고 플레이어 뒤의 보이지 않는 콜라이더에 맞을 경우 이기거나 지는 판정을 할 수 있게 함. |
| |  | | --- | | G:\GithubUnity\TCPStudy\포폴사진\탁구\20191013_200838.png | |  | | |
| **맡은 역할** | 1. Photon Network 탁구공 동기화.  2. 공의 바운스 조정.  3. 탁구채 |

**NavMeshAgent(길찾기 알고리즘)을 이용한 “Musa” PC 게임**

|  |  |
| --- | --- |
| **개발 기간** | 2019.06.28 ~ 07.05 (기획 및 개발 8일) |
| **개발 환경** | Unity 1.10f |
| **요 약** | 개발 인원 : 1명  NavMeshAgent를 활용한 캐릭터의 장애물 피하기, 공격, 특정지형에서 점프를 할 수 있는 움직임 만들기.  **기능 :**  **1) 장애물피하기 or 캐릭터간 이동시 간섭 피하기**  유니티에서 제공하는 NavMeshAgent기능으로 맵을 Baking 할 시 맵 위에 있는 Object에 특정 간격을 부여해 캐릭터가 이 Object를 지나갈 시 간격만큼 저절로 피해갈 수 있게 해 줌.  **2) Enemy가 공격하기**  플레이어와 특정 거리 이하로 가까워지면 공격애니메이션 발동.  **3) Enemy 정찰 및 플레이어캐릭터 따라다니기**  Enemy가 정찰을 할 수 있도록 빈게임오브젝트를 정찰할 지역에 위치시키기. 해당 지역을 돌아다니게 하기. 플레이어와 특정 거리 이하로 가까워질 시(Vector3.Distance 활용) NavMeshAgent의 SetDestination메소드로 플레이어캐릭터 추적. 플레이어캐릭터와 멀어질 시 다시 해당 지역을 돌아다니게 하기. |
| |  |  | | --- | --- | |  |  | |  |  | | |

**구글 AR Core – 엔진 분해 및 수리 체험 AR 어플리케이션**

|  |  |
| --- | --- |
| **개발 기간** | 2019.06.21 ~ 06.27 (기획 및 개발 7일) |
| **개발 환경** | Unity 1.15f / Android 7.0 Nougat / AR Core / Galaxy S9 |
| **요 약** | 개발 인원 : 2명  구글에서 제공하는 AR Core를 활용하여 고장난 엔진의 부품을 찾고 수리를 체험할 수 있는 어플리케이션 만들기. 4가지의 엔진을 선택할 수 있고, 각각의 엔진은 각각의 고장 시나리오를 가짐.  **기능 :**  **1) 바닥 메시 생성 및 중단**  구글 AR Core에서 제공하는 바닥 메시 생성 스크립트 DetectedPlaneGenerator 스크립트에서 매시생성을 중단시키고 눈에 보이지 않게 만들기. “Stop Mesh Making”버튼에 DetectedPlaneGenerator에 isBool값을 false로 바꿔 메시 생성을 중단. 이와 동시에 AR Core에서 제공하는 PointCloud 스크립트를 SetActive False하여 보이지 않게 하기.  **2) 엔진 상하 좌우로 회전 및 엔진 분해**  핸드폰 터치를 하여 상하 좌우 회전할 수 있게 하기. AR Core의 Frame.Raycast()메소드를 활용해 터치한 스크린의 좌표를 가상의 좌표로 전환하여 엔진을 생성 및 회전 시킬 수 있다. 고장난엔진의 부품 4곳을 미리 정한다. (“Engine Block, Gear Box, Exhausted Manifold, Fly Wheel”.) 엔진을 Select버튼으로 UI창에서 선택할 수 있다. 엔진 본체를 터치하면 고장부품 주위의 부품들이 상하, 좌우로 분해되고, 한번 더 더치하면 다시 조립됨.  **3) 고장 난 부품 표시하기**  각각의 고장 시나리오에 따라 고장 부품의 Renderer컴포넌트를 불러옴. Material의 SetColor중 \_EmissionColor를 빨갛게 하여 위험해 보이게 함.  (brokenPiece.material.SetColor(“\_EmissionColor”, color);  **4) 인벤토리에서 교체할 부품 찾기**  팀원들이 바이브의 버튼 기능을 바로 사용할 수 있도록 바이브의 버튼 조작 환경을 조성함.  **5) 엔진 그림자 생성**  Plane GameObject에 알파 값을 가지고 있는 원(PNG파일 이미지를 Sprite로 변환한 파일)을 엔진의 사이즈에 맞게 조정하여 엔진의 Position에서 0.1f 값 밑으로 생성하여 엔진이 공중에 뜬 것 같은 효과를 만듦. |
| |  |  | | --- | --- | |  |  | |  |  | |  |  | |  |  | | |
| **맡은 역할** | 1. 바닥 메시 생성 및 중단.  2. 고장 난 부품 표시하기  3. 엔진 그림자 생성 |

**삼성VR디바이스 – 디바이스 기초기능 VR어플리케이션**

|  |  |
| --- | --- |
| **개발 기간** | 2019.06.11 ~ 06.13 (3일) |
| **개발 환경** | Unity 1.10f / VIVE / GitHub / Gear VR /Galaxy S9 |
| **요 약** | 개발 인원 : 1명  가벼운 기능들을 추가하여 기어VR로 간단한 기능 구현해 보기.  **기능 :**  **1) 텔레포트**  기어VR은 오른손 콘트롤러만 있기 때문에 오른쪽 콘트롤러에서만 LineRenderer를 생성하여 RayCast를 표현함. 바닥 콜라이더에 맞으면 Aim이 생성되고 콘트롤러의 트랙패드를 클릭하여 이동할 수 있음.  **2) GearVR 헤드 움직임 Raycast 시각화**  OVRCameraRig를 이용하여 이용자의 움직임(헤드를 상하 좌우로 회전하는 움직임)을 가능하게 함. LineRenderer를 이용해 텔레포트를 위한 시각화 구현. |
|  | |