

Final Report

[Covid-Story]

1. Introduction

team title : 의쏘공

software title : Covid-Story

team members & member contribution



2015147562 이의동 컴퓨터과학과

역할 : 텐 제 전략 게임 구상 및 구현
게임 로직 구현, 카메라워크 구현, 맵 구현, 이벤트 구현



2015147533 유현석 컴퓨터과학과

역할 : 텐 제 전략 게임 구상 및 구현
행동 애니메이션 구현, 배경 음악 및 효과음 구현



2015147563 류동철 컴퓨터과학과

역할 : 메인 시나리오 구상 및 구현
애니메이션과 bgm, 효과음 구현,
몬스터 & 플레이어 행동 구현



2015147565 박성현 컴퓨터과학과

역할 : 메인 시나리오 구상 및 구현
카메라, 맵 세팅, 이벤트, npc 구현,
플레이어 행동 구현



2016147538 임재훈 컴퓨터과학과

역할 : 리듬 게임 구상 및 구현

2. Development purpose

코로나로 인한 외부활동 저하되고 사람들이 집에서 보내는 시간이 늘어남에 따라 제한된 외부활동 대신에 실내에서 할 수 있는 게임에 소비하는 시간이 늘어났다. 게임에 대해서 생각해보면 예전에는 주로 술게임, 아니면 보드게임이나 실제로 친구들과 만나서 어울리면서 했던 게임들이 생각났는데 지금은 그러한 평범한 일상들이 불가능해졌다.

그에 따른 대체로 온라인에서 누구나 쉽게 즐겁게 할 수 있는 게임을 기획하게 되었다. 즉, 스트레스를 해소하기 위해 게임을 소비하는 시간이 커지고 또한, 코로나로 인해 피해를 입고 있는 사람들에게 게임을 통해 그들의 스트레스 해소에 도움을 주기 위한 재미있는 게임을

개발하고 싶었다. 또한 외부 활동을 대체할 수 있는 흥미로운 컨텐츠가 있다면 집에서 머물면서 병역수칙 보다 잘 준수해 코로나를 보다 일찍 종속 시키는데 일조할 수 있을 것 같다.

따라서 스토리 라인을 구성할 때도 이러한 배경을 드러낼 수 있게 구성해서 사람들의 흥미를 유발했다.

3. Target software introduction

전반적인 개발은 아케이드 형식을 기반으로 한 모바일 게임으로 진행될 계획이다. 크게 메인 시나리오 게임과 2개의 서브 미니 게임으로 전체적인 게임 진행 방식을 정했다.

메인 시나리오 게임으론 '슈퍼마리오'와 같은 2D 횡스크롤 게임으로 진행될 예정이고 이 게임에 전반적인 목표는 코로나 치료제를 구하러 가는 것이다. 2개의 서브 미니 게임은 신나는 흥미 요소를 포함한 리듬 게임과 스트레스 해소에 직방인 슈팅이 바탕인 텐 제 전략게임으로 개발 중이다.

메인 게임과 2개의 서브 미니게임 모두 메이플스토리의 캐릭터 및 배경 등을 오마주했다. 메인 게임을 보다 자세히 설명하자면 백신을 가는 길에 몬스터와 여러 가지 이벤트들이 존재하고 이벤트들과 스토리의 진행에 따라 2개의 미니게임으로 연결된다.

리듬게임은 배경 노래에 맞춰 나오는 몬스터를 타격박스에 정확한 타이밍에 눌러야 하고 텐제 전략게임은 아군을 움직이면서 자동으로 움직이는 적군을 먼저 죽지 않고 제거하는 방식으로 이루어질 것이다. 이 모든 메인 게임과 서브 게임들을 이겨내면 백신을 구하는 엔딩을 볼 수 있.

4. Software requirements

1) User requirements

- 게임에 코로나를 이겨내자는 목표
 - 아케이드 기반의 게임 형식
 - 캐릭터의 단일화
 - 재미 추구
 - 리듬게임 형식의 미니게임
 - 슈팅게임과 시뮬레이션 형식의 미니게임
-
- + 게임을 신나게 해주는 배경음악
 - + 게임을 다채롭게 하는 효과음
 - + 시각적인 다채로움

2) Functional requirements

- 횡스크롤 형식으로 맵을 이동하면서 배경을 기준으로 키보드에 따른 캐릭터의 이동 가능
- 몬스터를 통한 난이도 조절
- 이벤트를 통한 미니게임으로 이동 기능
- 리듬게임 중 음악에 맞춰 나오는 퀘스트 기능

- 시뮬레이션 게임, npc를 구성하기 위한 인공지능 기능
- 맵과 게임에 따라 다른 배경음악이 나올 수 있는 기능
- 키보드 입력에 따른 캐릭터의 애니메이션의 변화 기능

- + 행동에 따른 효과음 기능
- + 상황에 따라서 달라지는 카메라의 뷰
- + 맵을 이동할 수 있는 포탈 시스템 기능
- + 상황에 따라서 나오는 대화창
- + 캐릭터의 상태를 알려주는 체력 바의 기능
- + 랜덤으로 설정되는 몬스터의 위치 기능
- + 몬스터가 재생성되는 기능
- + 플레이어의 공격과 몬스터의 피격 기능
- + 플레이어가 피격당할 시 무적 기능

3) System requirements

- 안드로이드 플랫폼에서 작동
- Unity를 활용한 물리법칙, 이벤트 및 게임 오브젝트 구현 및 외부 BGM, Image source 등을 소프트웨어에 접목
- 메인 게임과 미니게임들의 로직 구현
- 메이플스토리 게임에서 에셋을 추출할 수 있는 프로그램(위컴알)

- + 배경 음악 및 효과음을 추출할 수 있는 프로그램
- + 게임을 실행 시킬 수 있는 여러 구성요소

4) Non-functional requirements

- 캐릭터와 맵 디자인의 독창성
- 모든 안드로이드 기기에서 호환이 잘 되게끔 구현
- 보안적으로 안전, 쉽게 실행될 수 있는 접근성, 용량이 너무 크지 않게 잘 실행될 수 있는 공간의 효율
- 윤리적이나 사회적으로 문제가 되지 않아야 하는 규범성
- 나중에 피드백이 들어왔을 때의 유연성

- + 미니게임과 연결될 수 있는 스토리상의 적절한 흐름
- + 완성 후 게임을 했을 때 사람들이 즐거울 수 있는 흥미
- + 예전에 메이플스토리를 했을 때의 추억

5. System modeling and Architecture design

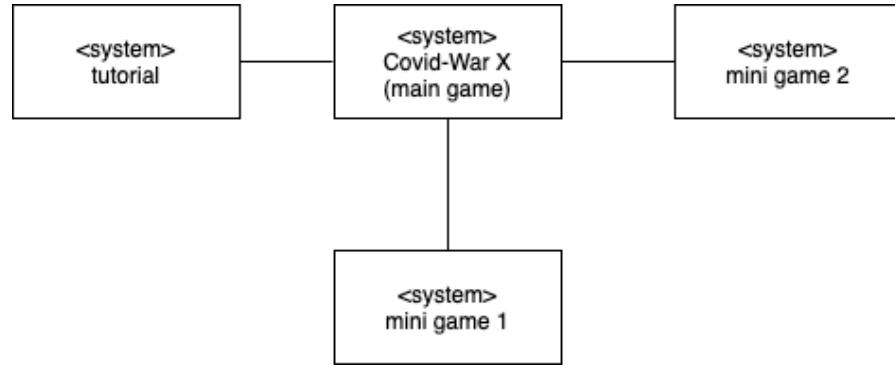
우리가 만들고자 하는 것은 모바일 게임 어플리케이션이다. 즉, 시스템 모델링을 하는데 있어서

게임 외부의 다른 시스템을 사용하지 않는다. 따라서 메인 게임 이외의 다른 게임들을 서브 시스템으로 잡고 모델링을 해보았다.

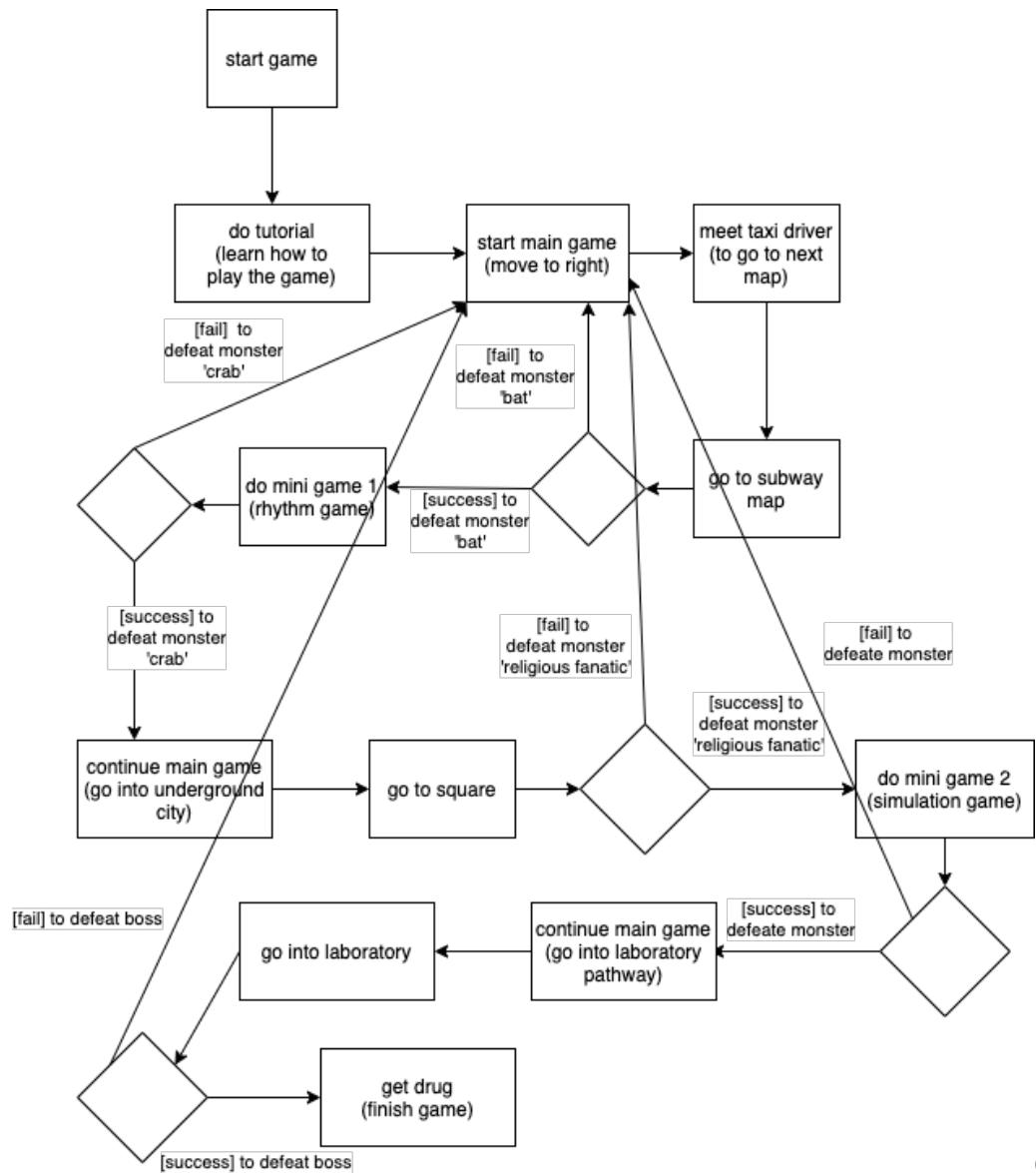
우선 context model 이다.(그림 1) 위에 언급한 대로 메인 게임을 하나의 큰 시스템으로 보고 나머지 미니 게임들과 튜토리얼을 서브 시스템으로 잡아서 구성을 했다.

다음은 process model이다.(그림 2) 게임에서 어떠한 것이 언제 이용되는지 좀 더 구체적으로 나타냈다.

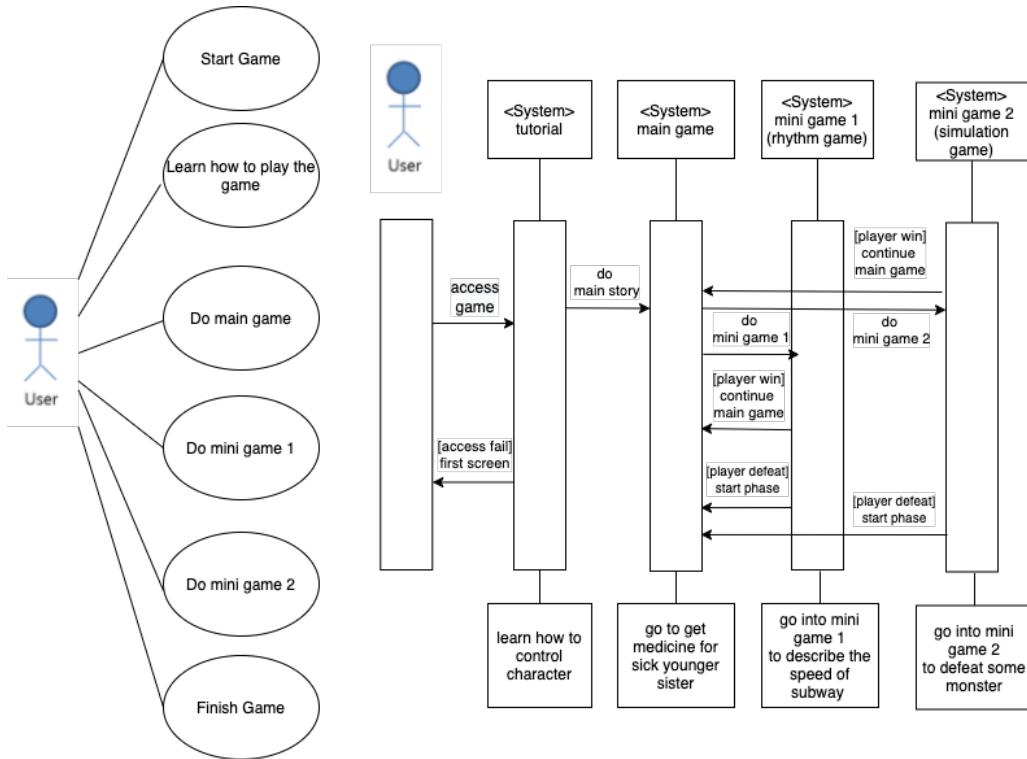
interaction model은 Use case model과 게임 흐름의 전체 sequence model 을 사용해서 그려주었다.(그림 3)



(그림 1)

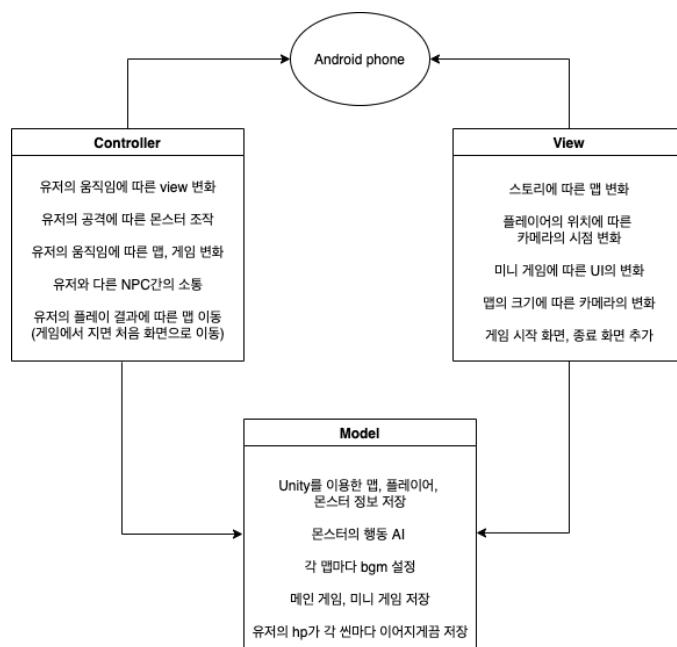


(그림 2)



(그림 3)

우리의 목표는 게임을 만드는 것이기 때문에 시스템을 layer로 나눌 수 없었고 client-server 간 통신 역시 필요하지 않았다. 또한 어떠한 데이터의 흐름에 따라서 게임이 진행되는 것이 아니고 게임에 스토리에 맞게 게임이 흘러가기 때문에 pipe and filter architecture도 맞지 않다고 생각했다. 따라서 MVP 모델에 우리의 소프트웨어를 맞추어 architecture design을 구성해보았다. 기존의 디자인에서 개발이 진행되는 과정에서 몇가지가 추가되었다.



6. Implementation description

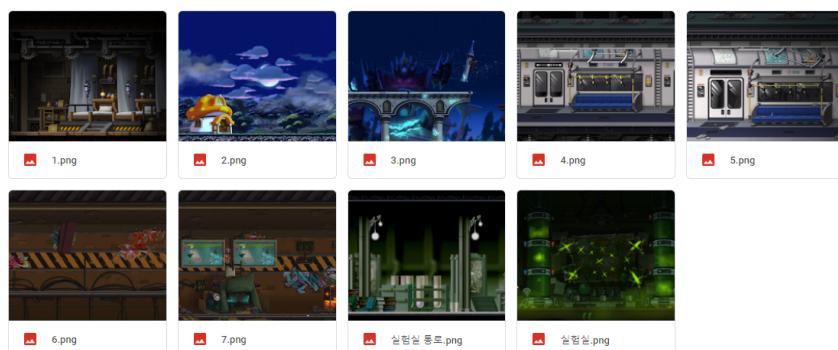
1) Main Game

메인 시나리오를 가지고 횡스트롤 2D 게임으로 구현했다. 백신을 찾아 떠나는 일정이다. 게임에 흥미를 유발하기 위해 현재 사회의 최대의 문제인 코로나 사태를 반영할 수 있게끔 스토리를 만들었다. 또한 게임의 각 씬이 스토리와 잘 맞게끔 구성했으며 스토리에 어울리는 긴박한 bgm으로 플레이에 집중할 수 있게끔 해주었다.

시나리오의 맵 이동 구성은 크게 다음과 같다.

집 -> 집 앞 -> 지하철 역 앞 -> 지하철 -> 부서진 지하철 -> 미니 게임(리듬게임) -> 선로 -> 피난처 -> 미니 게임(턴제 전략 게임) -> 피난처 -> 연구실 복도 -> 연구실

지하철과 광장을 중간에 선택한 이유는 코로나 사태에 국내에 발생했던 사건들을 모티브하기 위해서다. 지하철의 슬리퍼 사건, 광화문 집회 사건들을 오마주해 기획했다.



몬스터들도 코로나 사태에서 공감을 얻을 수 있게 에셋들을 선별했다.

코로나가 중국의 박쥐에서 창궐했다는 이야기를 모티브한 박쥐 몬스터, 바이러스를 연상시키는 몬스터, 광화문 사태에 영감을 얻어 종교인들을 연상시키는 사제 몬스터가 있다. 등장하는 몬스터들은 다음과 같다. 각 몬스터 별로 움직이는 애니메이션과 맞는 애니메이션을 넣어주었다.





여동생과 택시기사, 피난처를 지키는 사람, 총 세 명의 NPC가 존재하고 이에 따른 이벤트가 있다.



맵 간의 이동은 포탈이나 특수한 이벤트로 이동하게 했다.

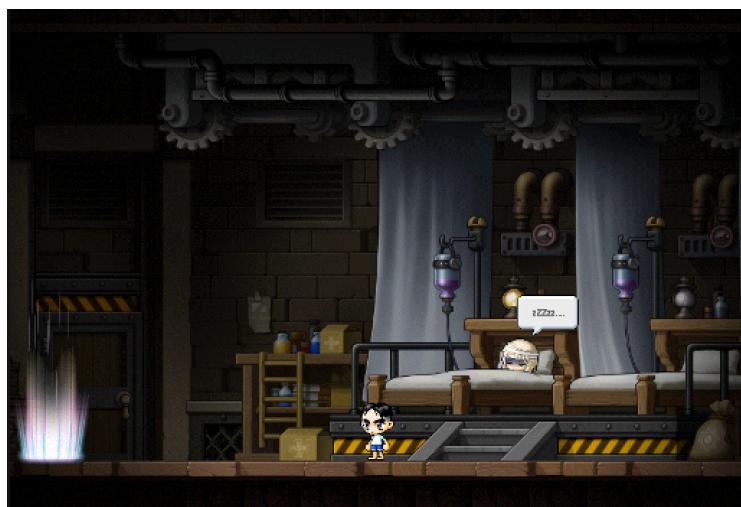


플레이어의 모습이다. 맨 처음 집 안의 씬에서는 마스크와 몽둥이가 없는 상태로 등장하다가 집 밖에 나오면 전염병으로 전염이 되지 않기 위해 마스크를 착용한다. 플레이어는 공격과 점프가 가능하다. 플레이어가 공격할 때, 몬스터가 공격을 당할 때 각각 상황에 맞는 효과음을 넣어줘서 좀 더 실감나는 플레이가 가능하게끔 해주었다.

씬에는 타이틀 화면과 게임 종료 화면을 추가해주었다. 우선은 타이틀 화면이다. 메이플스토리라는 컨셉에 맞추어 메이플 스토리의 모바일 배경화면을 이용했다. 게임 개발을 하는 도중 COVID-war x라는 이름은 아기자기한 우리 게임과 이미지가 맞지 않다고 생각하여 COVID-story로 이름을 바꾸어주었다.



화면을 클릭하면 게임이 본격적으로 시작된다. 가장 먼저 나오는 화면은 '집' 씬이다. '집' 씬에서는 여동생이 병세가 악화되고 있고 멀리 위치한 연구실에 백신이 보관되어 있다는 소문을 들은 주인공이 백신을 구하러 가기로 결심하고 집 밖으로 나가며 끝난다.



(처음 시작 맵 '집')

위는 실제 게임 플레이 화면의 캡쳐이고 처음에 동생과 대화를 하는 도중에는 게임이 멈춰 있다가 대화가 끝나면 포탈로 이동할 수 있게 된다. 분위기에 맞는 아련한 bgm을 채택하였다.

포탈로 나가면 '집 앞' 씬이 등장한다. '집 앞' 씬에서는 마스크와 몽둥이로 장비하고 동생의 병을 고치기 위한 여정을 떠난다. 왼쪽 상단 부분에 캐릭터의 hp를 표시해주는 UI를 하나 추가해주었다. 맵을 이동하면 택시기사를 볼 수 있고, 택시기사와 플레이어 간의 거리가 일정 거리 이하로 좁혀지게 되면 이벤트가 발생한다. 택시기사가 플레이어를 다음 씬까지 이동시켜주는 설정이다. 따라서 이 씬에는 포탈이 존재하지 않는다. 택시기사는 플레이어를 다음 씬인 '지하철 입구'로 이동시킨다.



(집 앞 씬)

다음 씬인 '지하철 입구'션이다. 이 씬으로 들어가자 마자 이벤트가 발생하여 택시기사가 여기까지 밖에 데려다 줄 수 없어서 미안하다는 대사를 남기고 이동할 수 있게 된다. 이 씬에서부터 몬스터가 등장하게 되는데, 여기에서는 광신도들이 등장한다. 광신도들은 기본적으로 느리고 체력도 많지 않다. 광신도들을 무찌르면서 맵을 나아가다 보면 지하철 입구와 포탈이 등장한다. 이 포탈로 이동할 경우 '지하철' 씬이 등장하게 된다. 몬스터들을 죽이면 랜덤한 위치에서 계속 재생성되며 이는 몬스터가 그 씬에서 설정되어 있는 최대 개수에 도달할 때까지 생성된다.



('지하철 입구'션)



'지하철 입구'션에서 지하철 입구가 등장하기 전에는 광신도 주교가 등장한다. 광신도보다 살짝 빠르고 체력이 더 많다는 특징이 있다. 이 몬스터는 한 마리만 등장한다.

몬스터는 플레이어가 일정 거리 안에 들어오면 플레이어를 추적해서 따라가는데, 플레이어가 몬스터에게 맞을 경우 플레이어는 일정 시간(1초) 동안 무적이 되고 살짝 투명하게 바뀐다. 이 시간 동안은 플레이어는 몬스터에게 맞지 않는다. 아래의 사진은 이 무적 시간이 발동된 모습과 몬스터에게 맞아서 hp가 감소된 모습이다.



다음은 '지하철' 씬이다. 이 씬에서는 박쥐 몬스터가 등장하게 된다. 박쥐 몬스터는 광신도보다 움직임이 빠르지만 체력이 약하다는 특징이 있다. 그 후에 '지하철 끝' 씬으로 연결되게 된다.



(‘지하철’ 씬)

몬스터는 피격 때 기존의 움직이는 모션이 아닌 새로운 애니메이션이 나오게 된다. 마찬가지로 플레이어도 공격 시 공격 모션이 존재한다. 몬스터가 죽을 때 역시 다른 애니메이션이 존재한다.



미니 게임을 끝마치고 나면 ‘지하철 끝’ 씬으로 이동하게 되고 반이 갈라진 지하철이 보인다. 마찬가지로 박쥐가 등장하고 이 지하철에서 떨어지게 되면 함정 이벤트가 발동하고 미니 게임(리듬 게임)을 플레이하게 된다.



(‘지하철 끝’ 씬)



(떨어지는 모습)

미니 게임을 끝내면 ‘철로’씬이 나오는데 ‘철로’ 씬에서는 광신도들과 박쥐들이 함께 등장하고, 많은 몬스터들을 통하여 플레이어는 다음 씬인 ‘피난처’로 이동하게 된다.



(‘철로’ 씬)

처음 ‘피난처’ 씬에 들어가면 NPC가 하나 존재하는데 이 NPC 근처에 다가가면 이벤트가 발생한다. 감염자들이 몰려오고 있다는 대사와 함께 미니게임(턴제 전략 게임)으로 들어가게 되고 이를

클리어하면 NPC가 사라지고 다음 씬으로 갈 수 있는 포탈이 생긴다.



(NPC와 이벤트)



(미니 게임이 끝나고 피난처로 돌아온 모습)

미니 게임이 끝나고 난 뒤에는 약이 있는 실험실로 가기 위한 '실험실 복도' 씬이 나온다. 여기에는 바이러스 형 몬스터들이 가득하다. 바이러스 형 몬스터들은 체력이 높고 움직임이 빠르다. 몬스터들의 크기 역시 상당히 크기 때문에 가장 힘든 씬이다. 아래 플레이 화면은 플레이어가 자신보다 훨씬 큰 몬스터들에게 둘러 쌓여진 모습이다. 맵을 통과하여 이동하게 되면 최종 보스가 있는 '실험실'로 가게 된다.



(‘실험실 복도’ 씬)

‘실험실’ 씬이다. 메인 게임의 마지막 씬이며 보스 몬스터가 등장한다. 보스 몬스터를 처치하게 되면 씬이 페이드 아웃 하면서 약을 구했다는 메세지가 나오며 게임이 끝나게 된다.



축하드립니다.
동생의 약을 구했습니다.

(‘실험실’ 씬)

플레이어의 hp가 다 달게 되면 Game Over가 되며 화면을 클릭할 경우 다시 맨 처음 씬인 ‘집’ 씬으로 돌아가서 처음부터 게임을 플레이할 수 있게 된다.



2) Sub mini Game(리듬게임)

Implementation description



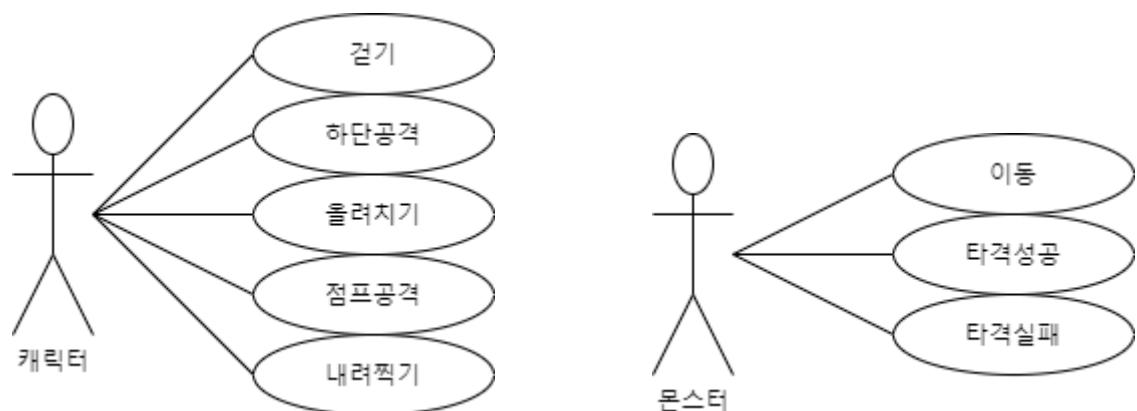
메인 게임에서 호출되는 미니 게임으로 리듬 게임을 구현하였다.

위 사진은 리듬게임의 화면을 구성하는 요소들을 나타낸 것이다.

게임은 기본적으로 리듬 게임의 형식을 따르며, 음악에 맞게 몬스터가 기존의 리듬게임의 '노트' 대신 나오는 형식으로 몬스터가 타격박스에 위치하는 타이밍에 맞게 캐릭터가 공격을 하는 방식으로 진행되는 리듬 게임이다.

캐릭터, 배경, 몬스터 등은 메인 게임과의 일관성을 유지하기 위함과, 캐릭터 sprite를 직접 제작하기에는 어려움이 있기에 기존에 존재하고 있는 게임 '메이플스토리'의 sprite를 활용하였다.

또한 캐릭터와 몬스터의 조건에 따라 발생하는 동작들을 USE CASE diagram으로 도식화 해보면 아래와 같다.



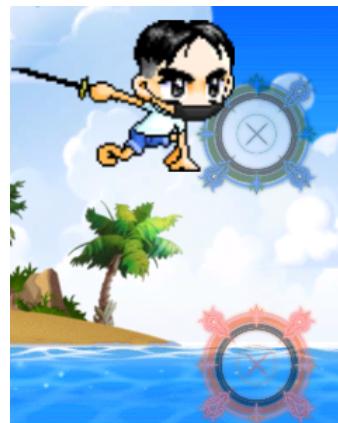
캐릭터의 동작은 크게 다음과 같은 **5가지** 동작으로 구성된다.



기본 상태 '걷기'



'하단 공격'



'점프 공격'



'올려치기'



'내려찍기'

위의 동작들은 다음과 같은 상황에서 동작한다.

캐릭터의 기본 상태인 '**걷기**'

'**걷기**' 상태에서 하단 공격을 시도하는 '**하단 공격**'

캐릭터가 점프 중인 상태에서 상단 공격을 시도하는 '**점프 공격**'

캐릭터가 '**걷기**' 상태에서 상단 공격을 시도하는 '**올려치기**'

캐릭터가 점프 중인 상태에서 하단 공격을 시도하는 '**내려찍기**'로 구성된다.

몬스터의 동작은 크게 다음과 같은 **3가지** 동작으로 구성된다.



기본 상태 '이동'



'타격 성공'



'타격 실패'

몬스터가 등장 후에 이동하는 '이동'

캐릭터가 타격에 성공했을 경우 회전하면서 점점 흐려지며 사라지는 '타격 성공'

캐릭터가 타격에 실패했을 경우 모션을 취하며 지나치는 '타격 실패'

몬스터의 종류는 **2가지** 종류로 구성된다.



'몬스터1'



'몬스터2'

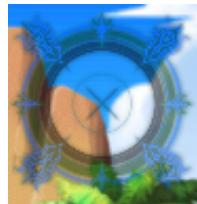
화면의 하단에 등장하는 '몬스터1'

화면의 상단에 등장하는 '몬스터2'

타격박스 또한 **2가지** 종류로 구성된다.



'하단 타격박스'



'상단 타격박스'

'몬스터 1'의 색상과 비슷한 색상으로 설정한 하단 타격을 위한 '하단 타격박스'

'몬스터 2'의 색상과 비슷한 색상으로 설정한 상단 타격을 위한 '상단 타격박스'

UI의 경우 아래와 같이 확인할 수 있다.



몬스터를 타격하는데 성공하면 콤보가 1 증가하고 체력바가 유지된다.

몬스터를 타격하는데 실패하면 콤보가 0으로 초기화되고 체력바가 한칸 깎인다.

BGM 및 타격 효과음은 보고서로 확인할 수 없지만 시연 영상을 통해 확인할 수 있다.



이러한 내용들을 구현한 플레이 화면은 다음과 같이 확인할 수 있다. 게임을 마치면 페이드 아웃이 되며 메인 게임으로 돌아가게 된다.



Issue and solution

Q. 캐릭터와 몬스터의 동작을 어떻게 자연스럽게 전환할 것인가?

A. 적절한 동작의 sprite를 animator를 이용하여 state를 전환시켜 보여주는 방식으로 구현하며, state 전환에 여러 변수를 도입하여 조건에 맞게 state가 전환되도록 한다.

Q. 어떤 방식으로 user의 재미를 확보할 것인가?

A. B 급 감성을 살릴 수 있는 소재들을 활용하여 user 의 흥미를 유발하며, 리듬게임 자체로서의 흥미를 유발하기 위해 노트의 sync 를 정확하게 조정하고, 기존의 일반적인 리듬 게임과는 달리 타격감을 강조하기 위하여 타격 효과음을 삽입하여 이를 두드러지게 구현한다.

Q. 노트의 경우 어떻게 생성할 것인가?

A. 노트의 생성을 담당하는 SpawnManager를 사용하여 게임 실행 시 temporal하게 노트가 object로 생성되게 한다.

Q. 노트의 sync가 안 맞을 경우 어떻게 조정할 것인가?

A. sync를 조정할 수 있도록 기능을 구현하였다.

```
void insert_Note(float a, float b){  
    //a += 1f; revise  
    if (b == -0.8f)  
        positions[top++] = new Vector2(a, b);  
    else  
        positions2[top2++] = new Vector2(a, b);  
}
```

중간보고서 대비 변경된 사항

상단, 하단 타격박스 색상 및 투명도 조절

몬스터를 노트로 박자에 맞게 등장하도록 구현

몬스터의 종류를 2가지로 구현(상단, 하단에 따라 다르게)

체력바, 콤보 UI 추가

캐릭터 동작 변경 및 추가

BGM, 타격효과음 추가

3) mini Game(턴 제 전략 게임)

턴 제 전략 게임(TBT) 형식의 미니게임으로, 플레이어 팀과 적 팀이 턴을 번갈아 사용하며 상대방에게 총알을 발사하는 게임이다. 비슷한 예시로는 Worms 시리즈, 포트리스 등이 있다.



[Worms 시리즈]



[포트리스]

Functional Requirements

이 게임이 제공해야 할 functional requirements는 다음과 같다.

1. 게임 관리

- 플레이어 팀 턴과 적 팀 턴이 번갈아 주어진다.
- 일정 시간이 지나거나 공격권을 사용하면 턴이 종료된다.
- 플레이어와 적은 체력바가 존재하며, 폭탄에 맞아 0이 되면 사망한다.
- 상대팀의 모든 인원을 제거한 쪽이 승리한다.

2. 플레이어 턴

- 마우스로 클릭 및 드래그하여 맵을 탐색할 수 있다.
- '이동' 상태일 때, 좌우로 움직이거나 점프하여 맵을 돌아다닐 수 있다.
- '공격' 상태일 때, 마우스를 이용해 방향과 파워를 조절하여 폭탄을 발사할 수 있다.

3. 적 턴

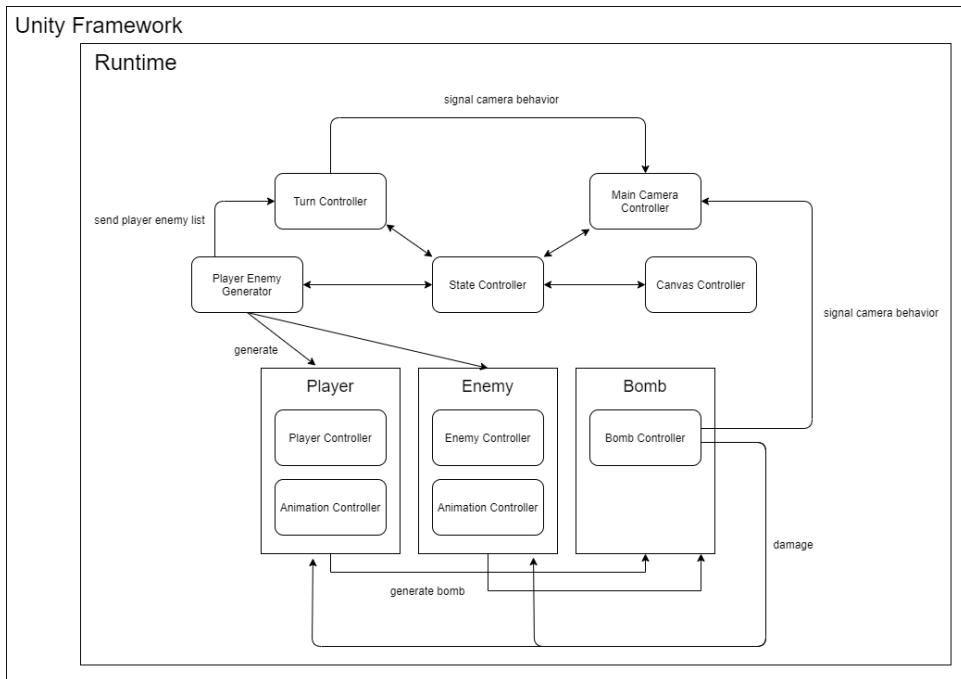
- 플레이어와 유사한 행동을 하는 인공지능 알고리즘이 존재한다.

4. 맵

- 폭탄에 의해 지형이 파괴되거나 변경될 수 있다.

System model

시스템 모델을 state diagram으로 표현하면 다음과 같다.



모듈 이름	기능
State Controller	게임의 전체적인 state를 관리하고 각 모듈들에게 전파한다.
Player Enemy Generator	플레이어와 적 인스턴스를 관리, 맵에 배치한다.
Turn Controller	턴을 관리하고 플레이어에게 권한을 부여/제거한다.
Main Camera Controller	유니티의 카메라 행동을 관리한다.
Canvas Controller	플레이어의 행동 로직을 관리한다.
Enemy Controller	인공지능으로 구현되는 적 행동 로직을 관리한다.
Animation Controller	캐릭터 애니메이션, 그래픽 등을 관리한다.

이전 개발 사항

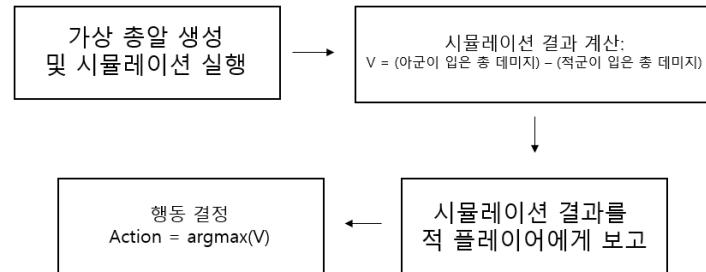
이전 개발 사항들은 다음과 같다.

	이전 개발 사항
1	State Controller 구현
2	Turn Controller 구현
3	Main Camera Controller 구현
4	Canvas Controller 구현
5	Player Controller 구현
6	맵(테레인) 구현
7	캐릭터 디자인(그래픽) 구현

추가 기능 개발 및 수정 사항

1) Enemy Controller 구현

인공지능으로 동작하는 적 플레이어 로직을 구현하였다. 구체적인 행동 로직은 다음과 같다.



적이 턴 권한을 획득하면, 우선 가상 총알들을 여러 방향과 파워로 생성한 후 시뮬레이션을 실행 한다. 생성된 각각의 가상 총알들은 $V = (\text{아군이 입은 총 데미지}) - (\text{적군이 입은 총 데미지})$ 의 식으로 V 를 계산한 후, 적 플레이어에게 보고한다. 마지막으로, 가장 큰 V 를 창출한 방향과 파워를 선택한 뒤 행동을 결정한다. 만약 $V = 0$ 이라면, 적군은 시간 제한동안 아무 행동도 취하지 않는다.

2) Animation Controller 구현

캐릭터의 상태에 따라 변화하는 애니메이션을 추가하였다.



[아군 정지]

[아군 점프]

[아군 이동]

[아군 사망]



[적군 정지]

[적군 사망]

3) BGM 및 오디오 소스 추가

게임의 몰입도를 높일 수 있는 음향들을 추가하였다.

4) Main Camera Controller 기능 추가

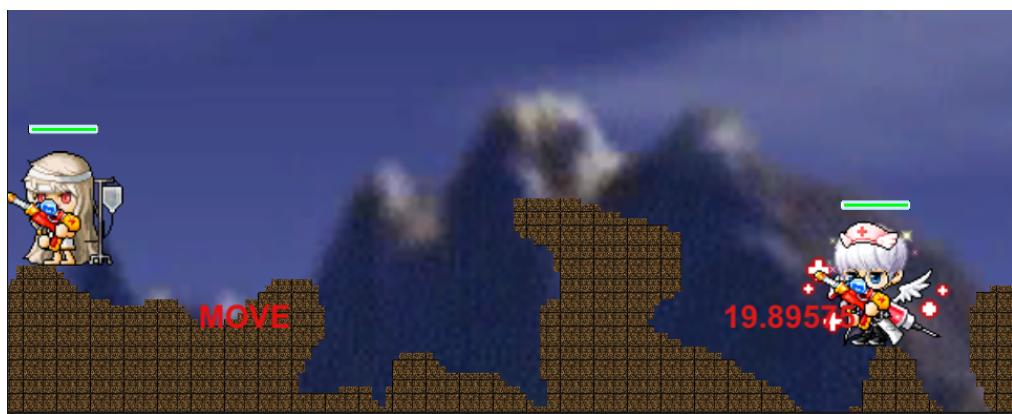
유저의 눈을 즐겁게 하고 게임 흐름을 매끄럽게 하기 위하여, 몇 가지 추가 기능들을 구현하였다.

기능 이름	기능 설명
게임 시작 시 카메라워크	카메라를 천천히 내리면서 넓은 화면에서 전체 맵을 보여준다. 캐릭터 배치가 완료되면, 첫번째 플레이어 위치로 카메라가 이동하며 줌인한다.
턴 이동 시 카메라워크	플레이어와 적의 턴이 전환될 때, 다음 캐릭터 위치로 카메라가 이동하며 줌인한다.
폭탄 발사 시 카메라워크	폭탄을 발사할 때, 카메라는 폭탄의 위치를 중심점으로 따라가며, 맵의 전체적인 상황을 보여주기 위해 줌아웃한다.
플레이어 사망 시 카메라워크	폭탄을 발사할 때마다 현재 턴에 죽은 캐릭터들을 계산한 후, 한명씩 줌인하며 사망 애니메이션을 보여준다.
게임 종료 시 카메라워크	플레이어가 전원 사망할 시 LOSE, 적이 전원 사망할 시 WIN, 양 진영 모두 사망할 시 DRAW 메시지를 띄우며 더 이상 카메라는 움직이지 않는다.

플레이 스크린샷



< 맵 + 플레이어 초기 랜덤 배치 >



< 플레이어 턴 – “이동” 선택>



< 적 턴 – 플레이어 권한 제거 >



< 플레이어 턴 – “공격” 선택 후 폭탄으로 적을 맞춤 >



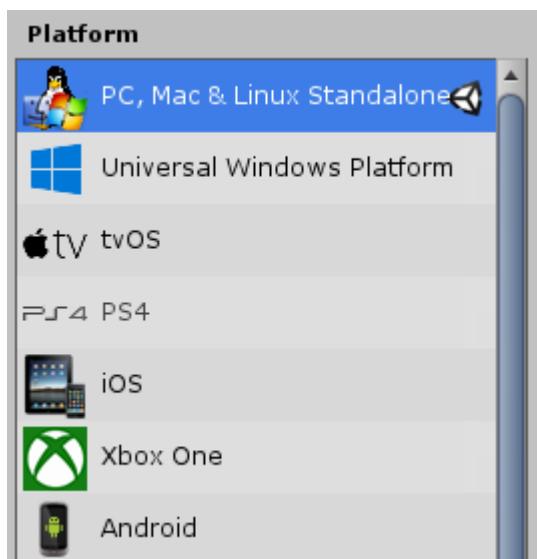
< 폭탄 발사로 맵 지형이 변경된 모습 >

Implementation and Testing

Git 을 사용한 Configuration Management

📁 .vs/의소공/v16	v1.05
📁 CovidStory	v1.06
📄 .gitattributes	Proto
📄 README.md	Update README.md
📄 의소공.sln	Proto

Unity 빌드를 이용한 여러 환경에서 실행가능하도록 하는 Host-target Management



Unity 환경을 이용하여 개발과 동시에 테스트를 수행하는 Development Testing

