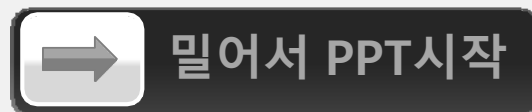


테디 디펜스 (프로그래머)



테디 디펜스 제작 정보



제작 인원 : 2명 (코드 작성) , (2d 컨텐츠 제작) 상황 : 제작중...

게임 디자인 측면 쪽의 소스 출처들 :

<https://assetstore.unity.com/packages/3d/characters/toon-reaper-62209>

<https://assetstore.unity.com/packages/3d/characters/humanoids/funny-bear-84649>

<https://assetstore.unity.com/packages/3d/environments/fantasy/green-forest-pack-super-bloom-series-95350>

테디 디펜스 제작 정보



프로그래밍 관련 사용 도구 :

서버 : playfab, photon(chat 기능 포함)

게임 엔진 : 유니티 1.8f 버전

기반 언어 : c#

타겟 플랫폼 : ios , android

테디 디펜스 로비 제작 현황



1. 로그인
2. 상점
3. 인벤토리
4. 인형 관리(캐릭터 목록)
5. 친구
6. 게임 플레이

테디 디펜스 로비 로직

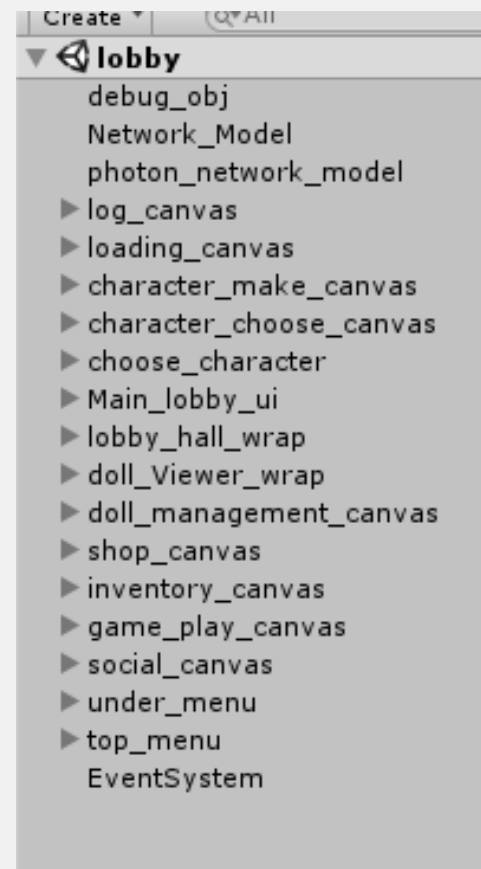


로비의 게임 오브젝트의 목록입니다.

각각 게임오브젝트별로 제작한
컴포넌트를 달고 서로
상호작용하면서 작동합니다.

network_model은 playfab의 서버
모델을 제작한것이고

photon_network_model은 포톤의
서버 모델을 제작한것입니다.



테디 디펜스 로비 (로그인)



게임이 시작할때 자동으로
google_login()을 호출하여 구글
sdk를 사용하여 인증과 구글플레이에
로그인을 합니다.

로그인에 성공을 하면
OnLoginSucc을 호출하여 다음
작업을 진행하고 로그인 정보를
game_System_network클래스에
정적변수로 저장하고 그 정보를
기반으로 playfab에 유저 정보를
작성하여 실제 게임에서 사용할
아이디를 생성합니다.

그후 playfab의 식별번호로 photon
서버와 채팅 서버에 연결을 합니다.

테디 디펜스 메인 로비



로비의 핵심 로직에는 playfab 서버와 연결 해제 및 서버와 통신과 관련된 기능을 가지고 있습니다.

캐릭터나 아이템 뽑기를 하여 정보가 추가되거나 변경되면 서버에서 다시 정보를 불러 와서 모든 로비에 뿌려 각각의 기능을 수행할 수 있게 설계되었습니다.

테디 디펜스 playfab 데이터 구조



playfab고유의 인벤토리라는 스키마와 유사한 개념을 가진 자료구조를 사용합니다.

Status	Item ID	↓ Create d	Expires	Uses left	Actions
Active	equitment_list 부속품 보유 리스트입니다. Instance ID: 7D91149F2A95D8E6	October 28, 2018 7:24 AM			Revo ke
Active	list 소유하고 있는 인형 목록입니다. Instance ID: 9A39FD4C407B8D64	October 28, 2018 7:24 AM		1	Revo ke
Active	inventory_list 인벤토리에 들어갈 리스트입니다. Instance ID: 8BB97367D1015C07	October 28, 2018 7:24 AM			Revo ke

상점



뽑기권과 여러가지 상품을 구매하는 창입니다.
inventory_list라는 인벤토리의 구조입니다.
각 상품의 key 마다 상품의 정보(json 구조)를 저장하고 그
json은 각키마다 1kb가 넘어가지 않습니다.

key는 아이템의 고유 id이기 때문에 같은 종류의 아이템은
그 key의 갯수가 올라갑니다.

<input type="checkbox"/> Key	Value
<input type="checkbox"/> shop_item_slot_1	<code>{"id":"1","pay_type":"well_done","price":"1","count":"1"}</code>
<input type="checkbox"/> shop_item_slot_0	<code>{"id":"0","pay_type":"well_done","price":"1","count":"2"}</code>
<input type="checkbox"/>	

SAVE CUSTOM DATA



상점에서 구매한 상품이나 게임에서 얻은 상품등을 사용하기 위해 만든 로비입니다.

아직 다른 기능은 제작중이고 랜덤 뽑기기능을 만들었습니다.

제조권을 사용하여 장비나 캐릭터를 뽑습니다.

랜덤뽑기를 제작하기 위해 유니티로 외부프로그램을 제작하여 그 어플에서 확률을 넣으면 시드로 서버에 업로드를 합니다.

그 시드를 기준으로 rand함수를 호출해 랜덤 뽑기를 제작하였습니다.

테디 디펜스 인형 목록



인벤토리에서 제조권을 사용하여 랜덤 뽑기를 하여
생성한 인형이나 보상으로 주는 인형 모두를
유저에게 보여줄수 있는 창입니다.

게임에서 사용할 인형을 고르거나 인형의 장비를
변경하거나 인형이나 장비의 정보를 볼수있습니다.

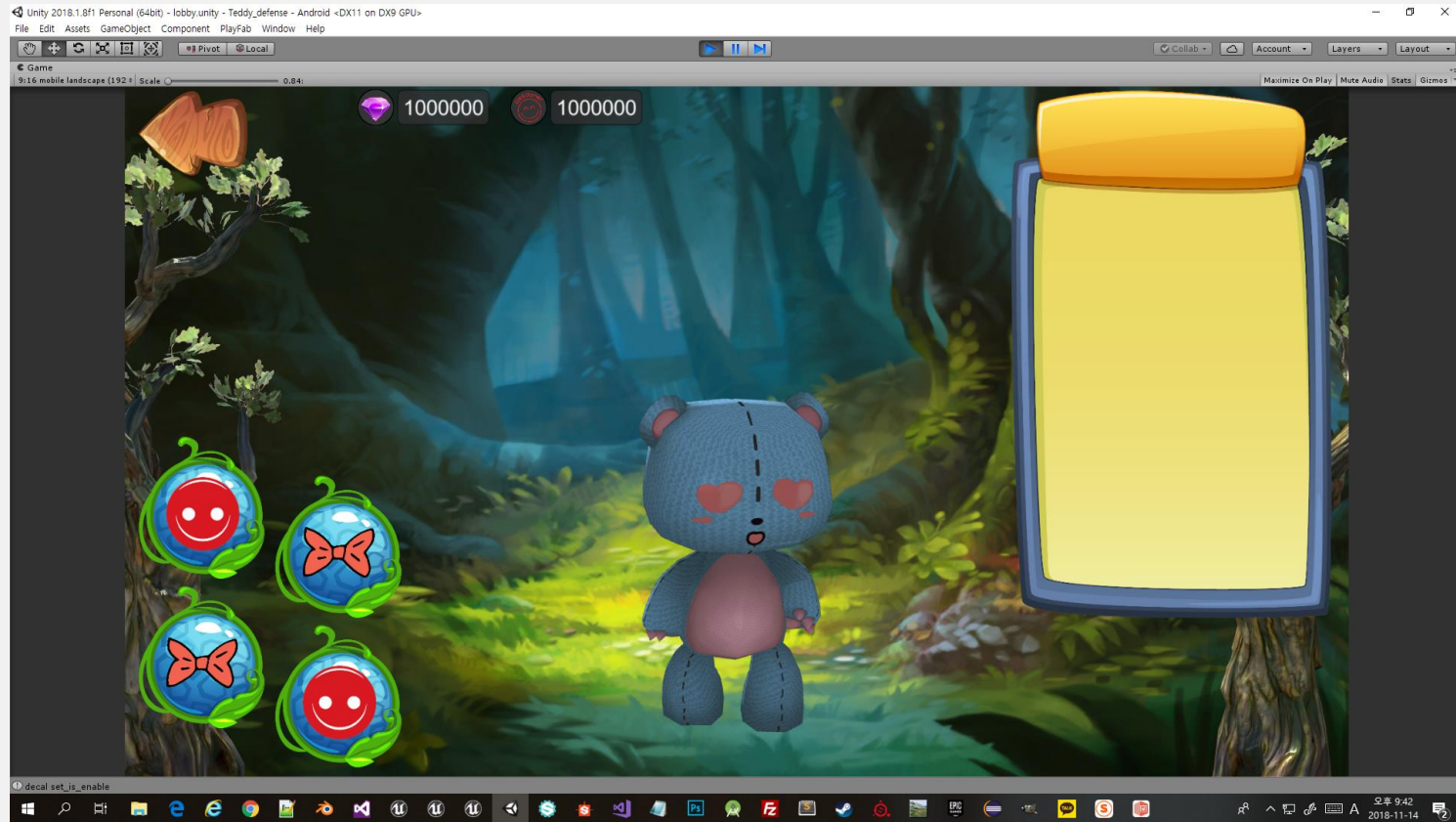
테디 디펜스 인형 목록 이미지



Unity 2018.1.8f1 Personal (64bit) - lobby.unity - Teddy_defense - Android ->DX11 on DX9 GPU<
File Edit Assets GameObject Component PlayFab Window Help



테디 디펜스 인형 목록 이미지



테디 디펜스 친구 로비



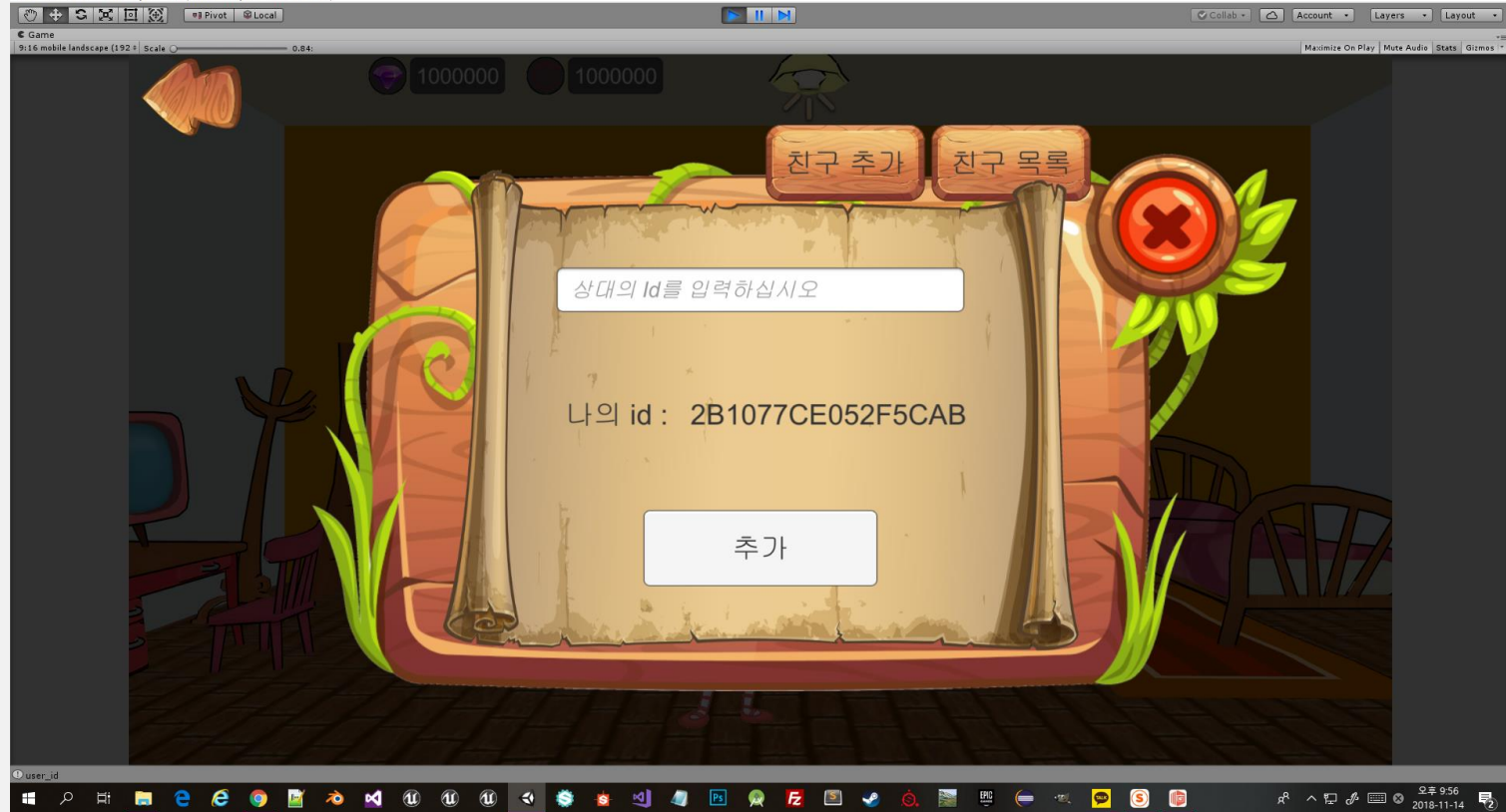
친구 로비에서는 친구의 id를 입력하여 친구
추가를 할수있고 친구와 채팅과 게임
초대문자를 주고 받을 수있습니다.

photon sdk의 chatclient 기능을 사용하여
photon_network_model의 게임 오브젝트에서
채팅 서버와 연결끊김 및 메시지 수신등을
인식하여 social_controller 클래스로 기능
처리를 요청하여 해당 기능을 수행합니다.

테디 디펜스 친구 추가



Unity 2018.1.8f1 Personal (64bit) - lobby.unity - Teddy_defense - Android <DX11 on DX9 GPU>
File Edit Assets GameObject Component PlayFab Window Help



테디 디펜스 채팅 기능



Unity 2018.1.8f1 Personal (64bit) - lobby.unity - Teddy_defense - Android <DX11 on DX9 GPU>
File Edit Assets GameObject Component PlayFab Window Help



테디 디펜스 게임 플레이 로비



스테이지 선택 및 난이도와 파티 편성과 친구 초대 기능을 하는 로비 입니다.

스테이지 선택과 친구 초대,파티 편성을 제작중입니다.

스테이지 선택의 경우 로컬에 저장된 있는 json데이터를 기반으로 스테이지목록을 구성하고

친구 초대의 경우 슬롯을 클릭하여 친구에게 초대 문자를 전송하여 친구가 클릭시 photon 서버에서 방 입장을 인식하여 마스터 클라이언트에서 모든 설정 완료후 인 게임으로 들어가는 기능을 제작중입니다.

테디 디펜스 로비 동영상



<https://youtu.be/wQ9O9eRVRRU>

인 게임 이미지



테디 디펜스 인게임 설계



1. 아군 배치 기능 설계
2. 적군 설계

인 게임 영상



<https://youtu.be/CD5TKHAgXHI>

플레이 영상



카드 UI에서 드래그앤 드롭으로
배치 가능한곳에 배치하고 배
치할때 파란 원안에 들어가 있으면
그룹의 구성원으로 설정 됩니다.

아군 배치 기능



1. 아군 배치 원리
2. 아군 그룹 멤버 설정 원리



터치 이벤트가 들어오면 현재 터치의 위치 값을 받아서 카메라로 부터 레이 캐스팅을 하여 현재 터치중인곳이 U인지 게임안인지를 판단합니다.

만약 터치를 끝냈을때 터치 했었던 곳이 게임이면 현재 아군이 배치가능한지를 미리 설정한 태그로 확인을 합니다.

아군 그룹 멤버 설정 원리



아군이 배치되면 아군 디폴트 클래스에서 OnTriggerEnter이벤트에서 현재 그룹의 범위에 속했는지를 확인하고 속해있다면 그 그룹의 구성원으로 추가하고 아니면 새로운 그룹을 생성합니다.



그룹 클래스는 먼저 생성 되자말자 멤버들이 가야할 자리를 계산하여 저장합니다.(총 6곳)

일단 그룹을 대각선으로 4군대의 점을 찍습니다. (45도와 225도 양점을 찍고 그값들을 재활용하여 나머지 두점을 찍습니다.)

그 이은점들을 비율을 기준으로 점을 찍습니다.



그 이은 점들 가지고 두선의 교차점을 구하는 공식을 사용하여 총 6곳의 위치를 구합니다.

공식 출처:

https://zetawiki.com/wiki/%EB%91%90_%EC%A7%81%EC%84%A0%EC%9D%98_%EA%B5%90%EC%B0%A8%EC%A0%90

그렇게 구한 6곳의 위치를 멤버가 추가되거나 그룹전체가 이동하면 그 위치까지 이동시킵니다.

아군 그룹 멤버의 이동



생각한 이동방식과 비슷한 예

<https://youtu.be/cMBzzO2DEC0?t=23s>

위의 영상의 23초에서 52초까지의 동작들과 유사하게 이동할수 있도록 설계



그룹의 위치가 이동하면서 각 프레임마다 모든 멤버들을 해당 위치로 이동 시켜야 합니다.

필요한 알고리즘 길찾기 알고리즘
(A* 알고리즘, 길 정보를 이용할수 있는 길 생성 알고리즘, 프레임 마다의 위치값을 받아서 이동 하는 역학 공식)

길 찾기 알고리즘을 구현하려면 많은 시간이 필요하기 때문에 인디 개발의 단계에선 제작하기 까다롭습니다.

그래서 구매 하여 훗날 새로 작성할 예정입니다.
(구매한 소스는 저작권 때문에 열어 볼수가 없게 되었었기 때문에 수정이 불가능합니다.)

구매한 알고리즘 평가.



맵의 위치 정보는 실생활의 좌표와 같으므로 이를 바로 알고리즘에 적용하기는 힘듭니다. 적용하기 위해선 간략하게 표현해야 합니다.

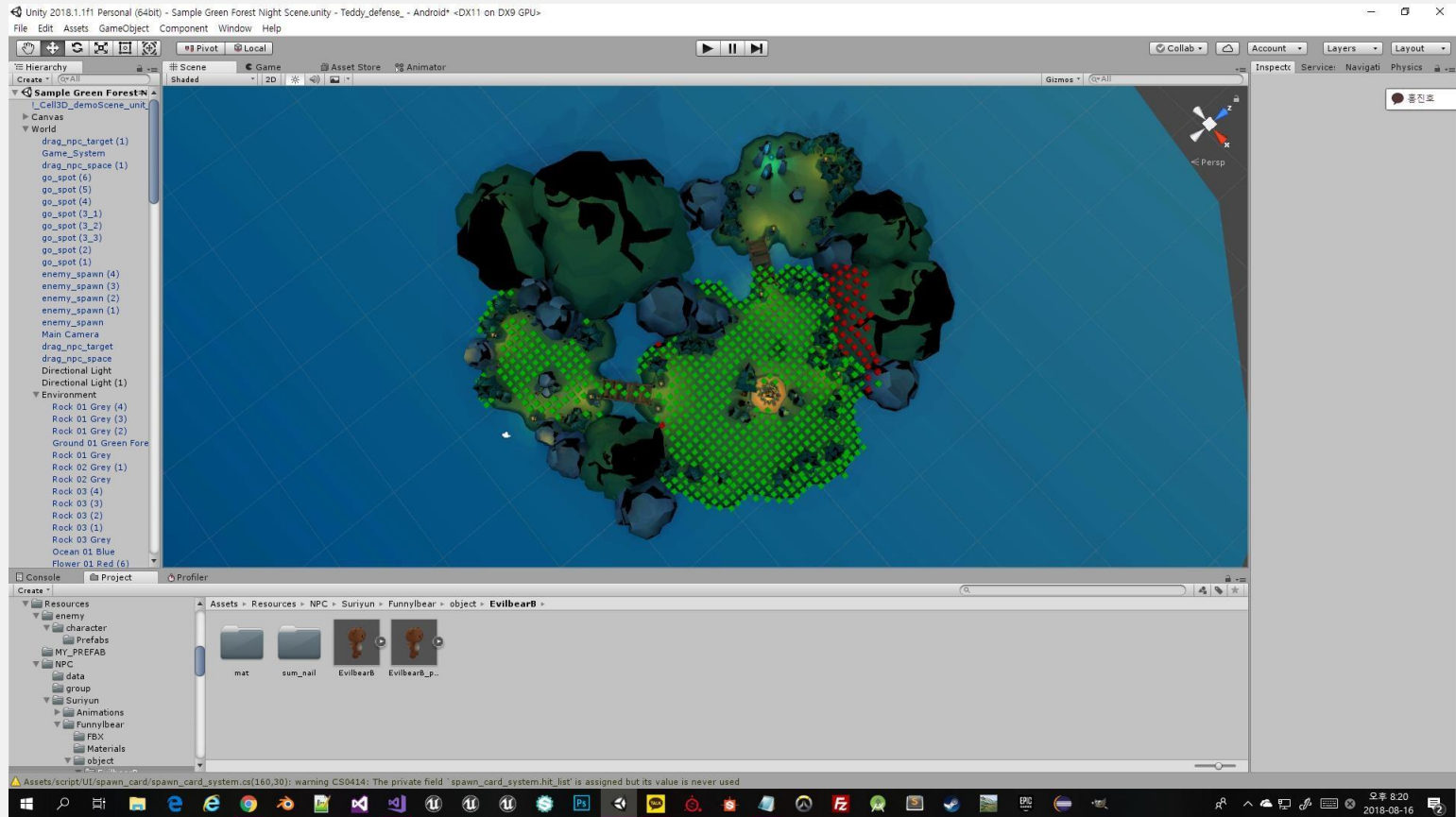
구매한것에는 일정 구간만큼씩 짤라서 좌표로 설정이 되었습니다.

하지만 이 좌표 구간을 베이킹을 하지 않은 것처럼 보이기 때문에 훗날 새로 작성 할때 같이 제작할 예정입니다.

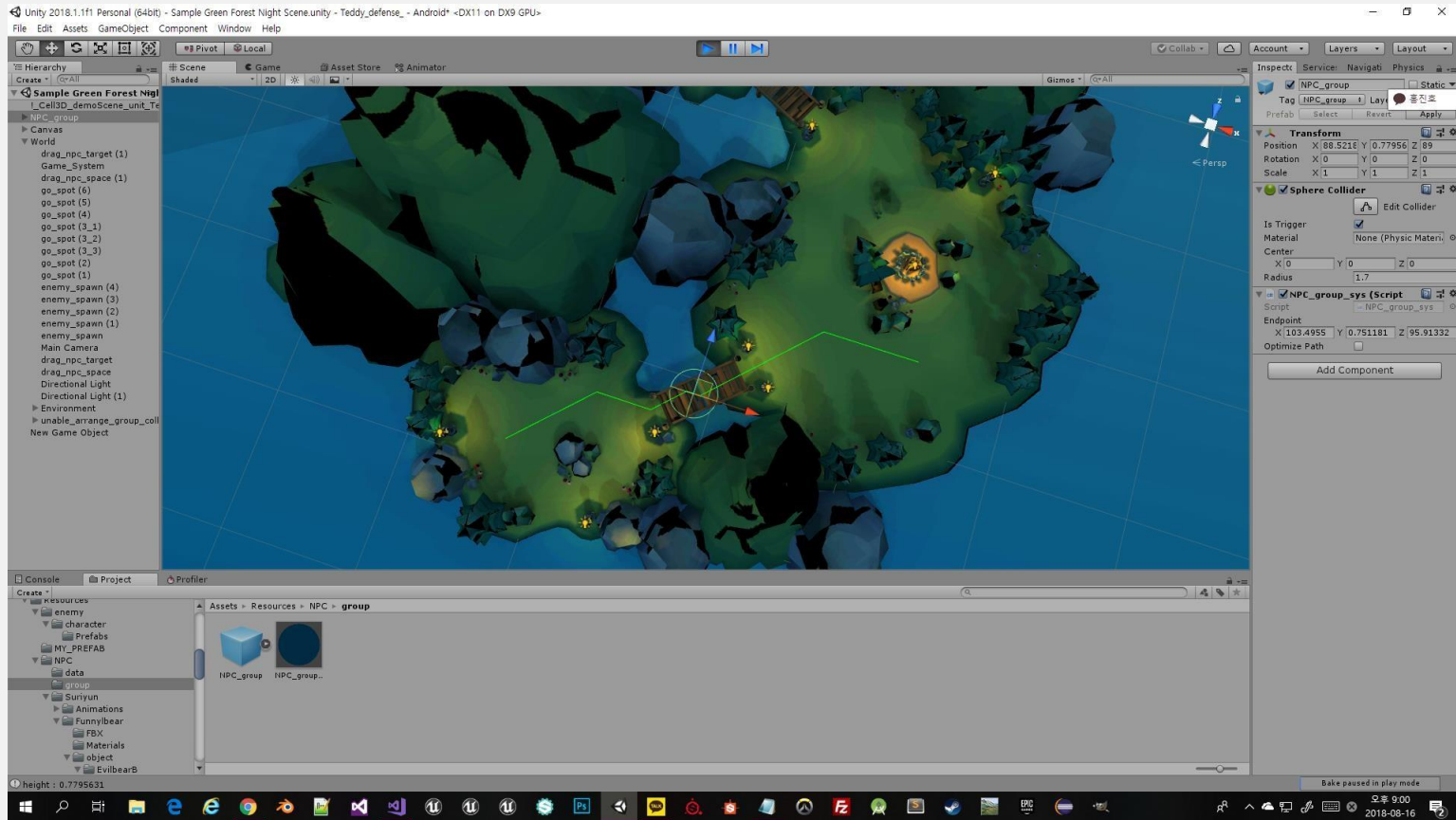
(베이킹을 하지않으면 본게임 실행할때 좌표 작성하는 시간이 걸립니다.다)

길찾기는 A* 알고리즘으로 작성할 예정입니다.

이동 경로를 구하기 위한 구간들을 표시한 이미지



이동 경로를 표시한 이미지 시작점 왼쪽, 목표 오른쪽



길찾기 알고리즘의 출처



<https://assetstore.unity.com/packages/tools/ai/pathfinding-21989>

유니티의 NavMeshagent를 그룹엔 적용안한 이유는 그룹은 is trigger을 설정 했기 때문에 물체와 부딪쳐도 통과하기 때문에 이를 계산하기 위해 길찾기 알고리즘이 필요 했습니다.



현재 구현된 기능은 길찾아 오는 것입니다.

유니티 에디터에서 총 이동할 곳들을 먼저 지정해놓고 그 클래스들에게 level값을 매겨서 level이 낮은 곳부터 지나서 목표 위치 까지 옵니다.

level에 따라 정렬을 하였고 정렬은 유니티의 정렬 api(퀵정렬로 보입니다)로 하였습니다.

같은 레벨은 현재의 교통량을 기준으로 교통량이 적은 곳이 앞으로 정렬 됩니다.

조이스틱으로 화면 이동을 구현하였습니다.

터치로 드래그하는 것은 다른 터치 이벤트 때문에 어색하게 작동하여 변경하였습니다.

(터치한 위치 - 조이스틱의 현재 위치)의 길이를 구하여
조이스틱의 범위에서 나가면

(터치한 위치 - 조이스틱의 현재위치)와 (0,1)의 벡터의 내적을 구하여

아까 구한 길이와 내적을 활용하여

각도 = 아크코사인(v_1 과 v_2 의 내적 / v_1 의 길이 * v_2 의 길이)의
공식을 사용하여[선 위의 점 공식]을 구합니다.

$x = \text{조이스틱의 범위의 반지름} * \sin(\text{위의 공식에서 구한각도})$

$y = \text{조이스틱의 범위의 반지름} * \cos(\text{위의 공식에서 구한각도})$

하여 터치한곳이 조이스틱 범위 밖이면 최대 위치를 구해줍니다

조이스틱 위치 / 조이스틱 최대범위 하여 1로 정규화 해줍니다.



감사합니다
Thank You