

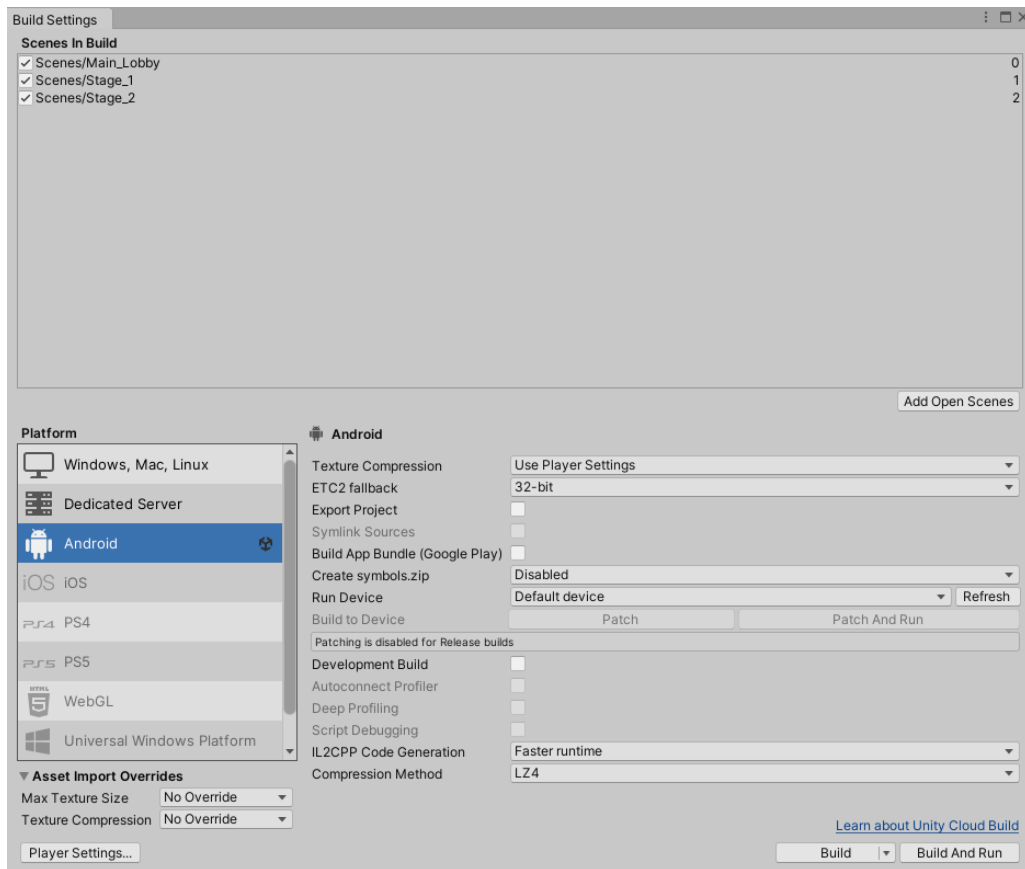
개발자 기술문서

어르신 나이스샷

목차

1. Build 방법
2. Main Function Script
3. UI & Sound Function Script
4. 조작키 할당방법

1. Build 방법



<Build 세팅>

Unity 에디터로 프로젝트를 실행한 뒤, 좌측 상단의 File – Build Setting을 선택, 위 사진과 같이 세팅하고 Build를 누르면 완료된다.

2. Main Function Script

1. GameManager.cs

중요 변수

Instance – 다른 스크립트에서 GameManager로 접근할 수 있게 해주는 변수이다.

cur_ball – 현재 차례인 공의 번호를 나타내는 변수이다.

phase – 단계를 나타내는 변수이며, start, ready, strike, done, lobby 다섯 단계를 사용한다.

turnPhase – 현재 단계를 나타내는 변수이다.

balls, hammer, timer – 스테이지 내에서 GameManager와 오브젝트를 연결해주는 역할로, 스테이지 내에 배치되어 있는 오브젝트들을 에디터에서 직접 넣어서 사용한다.

number_of_players – 플레이하는 사용자 수로, 사용하는 공의 개수와 같다.

score, red_score, blue_score – 점수를 관리하는 변수로 score에는 각 공의 점수가 저장된다.

함수

StartGame – 게임을 시작하는 함수이다. 각종 요소들을 초기값으로 초기화한다.

UpdateScore – 각 팀의 점수와 점수판을 최신값으로 업데이트하는 함수이다.

OnToggleStick – Start 단계에서 Ready 단계로, Ready 단계에서 Start 단계로 바꿔주는 함수이다.

OnNextTurn – Done 단계에서 Start 단계로 넘겨주며, 공의 번호도 다음으로 넘어간다.

OnTestStart – Lobby 단계일때, 조작을 통해 StartGame 함수를 호출한다. 조작키는 B버튼으로 설정되어 있다.

EndGame() – 게임을 종료하는 함수이다. 점수를 총합하여 어느 팀이 승리했는지 알려준다

2. Ball.cs – 공에 적용시키는 Script

변수

ball_number – 공의 번호를 나타내는 변수 1~8까지 할당 가능하며, 에디터상에서 설정 해야한다.

target_gate – 목표 게이트를 나타내는 변수로 초기값은 1, 게이트를 통과할 때마다 1씩 늘어난다.

out_ball – 공이 아웃되었는지 아닌지 판단하는 변수이다.

complete – 모든 게이트를 통과했음을 체크하는 변수이다.

last_loc – 공의 마지막 위치를 기록하는 Vector3 변수이다.

함수

OnCollisionEnter(Collision collision) – 공이 충돌했을 때 호출되는 함수이다. 충돌한 물체의 태그가 Hammer일때와, Ball일때로 구분해서 각각의 기능을 따로 발동한다.

CheckThisBallIsMoving() – 공이 움직이고 있는지 체크하기 위한 함수로, MovementDetector 스크립트를 활성화 시킨다.

SaveLocation() – 공의 현재 위치를 last_loc 변수에 저장한다.

3. Border.cs – Stage1 경계선(Border)에 적용시키는 Script

변수

isBorderlineX – 경계선의 방향이 X축이면 true, Z축이면 false인 변수이다

outline – 안쪽 경계면의 좌표값을 저장하는 함수이다. 게이트의 방향이 X축이면 x좌표 값을, Z축이면 z좌표값을 저장한다.

함수

OnTriggerEnter(Collider other) – 공이 경계에 닿았을 때 발동하는 함수로, 공이 나간 좌표값을 저장하고 공을 비활성화 시킨다.

4. Border2.cs – Stage2 경계선(Border)에 적용시키는 Script

함수

OnTriggerEnter(Collider other) – 공이 경계에 닿았을 때, 공이 들어온 방향으로 반사해 튕겨낸다.

5. gate.cs – 게이트에 적용시키는 Script

변수

gateAxisIsX – 게이트의 방향이 X축이면 true, Z축이면 false인 변수이다

isFrontPositive – 게이트의 앞면이 +방향이면 true, -방향이면 false인 변수이다.

point_in, point_out – 공이 들어오고 나간 좌표를 기록하는 변수이다.

gate_number – 게이트의 번호를 나타내는 변수이다. 에디터내에서 설정해줘야 한다.

함수

OnTriggerEnter(Collider other) – 공이 들어온 좌표를 point_in에 기록한다.

OnTriggerExit(Collider other) – 공이 나간 좌표를 point_out에 기록하고 PassingBall 함수를 호출한다.

PassingBall() – 공이 게이트를 지나갔을 때 호출되며, 공이 들어온 좌표와 나간 좌표를 비교해 공이 올바른 방향으로 지나갔는지 판별해주는 함수이다.

CheckPass() – 공이 올바른 방향으로 지나갔을 때 호출되며, 해당 공 번호의 점수를 올리고 공의 목표 게이트를 다음 게이트로 설정해준다.

ResetValue() – 공이 지나가고 난 뒤, point_in과 point_out값을 초기화 시켜주는 함수이다.

6. GolePole.cs – 골 폴에 적용시키는 Script

변수

gate_number – 게이트의 번호를 나타내는 변수이다. 에디터내에서 설정해줘야 한다.

함수

OnTriggerEnter(Collider other) – 공이 골 폴에 충돌했을 때, 공이 앞 게이트를 모두 통과했다면 공의 complete 변수를 true로 바꾸고 공을 비활성화 시킨다.

7. MovementDetector.cs – 공에 적용시키는 Script

함수

disableThisScript() – 공의 움직임이 정지했을 때, 이 스크립트를 비활성화 시키는 함수이다.
Strike 단계에서 Done 단계로 넘겨주며 다음 플레이어가 준비될 때까지 Timer를 정지한다.

8. Timer.cs – 타이머에 적용시키는 Script

변수

pause – true면 시간이 멈추고, false면 흘러간다.

remain_time – 남은 시간을 나타낸다. 0 이하로 떨어지게 되면 이 스크립트를 비활성화한다.

함수

TimerStart(_time) – 입력한 시간값을 초기값으로 타이머를 시작한다.

PauseTimer() – 타이머를 일시정지한다.

ResumeTimer() – 타이머의 일시정지를 해제한다.

3. UI & Sound Script

1. UISelectBtn.cs

개요 : 이 스크립트는 게임에서 컨트롤러 입력을 감지하여 HammerUIPanel에 있는 버튼 선택 및 동작을 수행한다. 조이스틱 이동에 따라 버튼을 선택하고, 버튼을 누르면 해당 버튼의 동작이 실행되는 기능을 구현한다. 또한, 초기에 설정된 버튼의 색상을 변경하여 선택된 버튼을 강조한다.

변수

rightControllerCharacteristics, leftControllerCharacteristics : 오른손 및 왼손 컨트롤러 특성을 지정하는 변수이다. (InputDeviceCharacteristics 타입)

manualButton, backButton, closeButton, exitButton : 게임 오브젝트에서 연결된 UI 버튼을 나타내는 변수이다.

ManualPanel : 게임 오브젝트를 나타내는 변수로, 이 변수의 활성화 여부에 따라 패널이 표시되거나 숨겨진다.

UIButtons : UI 버튼들을 저장하는 배열이다.

selectNum : 현재 선택된 버튼의 인덱스를 저장하는 변수이다.

oldCol : 각 버튼의 이전 색상을 저장하는 배열이다. (ColorBlock[] 타입)

selectCol : 선택된 버튼의 변경된 색상을 저장하는 변수이다. (ColorBlock 타입)

함수

Start() : oldCol 배열에 각각의 버튼들의 colors를 할당한다.

(Color 속성은 재선언으로 바꿀 수 없기에, selectCol에 UIButtons 배열 0번째의 manualButton colorBlock을 할당한 후, selectCol.normalColor을 빨간색으로 설정한다. normalColor가 빨간색인 colorBlock을 0번째 manualButton의 colors로 할당한다.)

SetButtonArray() : UI 버튼들을 UIButtons 배열에 할당하는 함수이다.

MoveJoyStick(InputAction.CallbackContext context) : 조이스틱 이동 입력을 처리하는 함수이다. 조이스틱의 y 값에 따라 선택된 버튼을 위로 이동하거나 아래로 이동한다.

PressButton(InputAction.CallbackContext context) : 버튼 입력을 처리하는 함수이다. 버튼을 누르면 현재 선택된 버튼의 onClick 이벤트를 호출한다.

ReturnOldBtnColor() : 이전에 선택된 버튼의 색상을 원래대로 되돌리는 함수이다.

MoveUp() : 현재 있는 버튼에서 위로 이동하는 함수이다. selectNum 인덱스 1만큼 감소, 현재 selectNum의 normalColor 빨간색 할당한다.

MoveDown() : 현재 있는 버튼에서 아래로 이동하는 함수이다. selectNum 인덱스 1만큼 증가, 현재 selectNum의 normalColor 빨간색 할당한다.

2. HammerUIPanel.cs

개요 : 이 스크립트는 HammerUIPanel의 각 버튼에 있는 이벤트를 설정하고 수행한다. 또한, Hammer UI 패널의 초기 상태를 설정하고, 입력을 통해 UI 패널을 열거나 닫을 수 있도록 구현되었다.

변수

manualButton, backButton, exitButton, closeButton : 게임 오브젝트에서 연결된 UI 버튼을 나타내는 변수이다.

hammerUIPanel : Hammer UI 패널을 나타내는 게임 오브젝트이다.

manualPanel : 메뉴얼 패널을 나타내는 게임 오브젝트이다.

함수

SetAddListener() : 버튼들의 클릭 이벤트를 설정하는 함수이다.

buttonManual_clicked() : 메뉴얼 버튼이 클릭되었을 때 호출되는 함수이다. Hammer UI 패널을 비활성화하고 메뉴얼 패널을 활성화한다.

buttonBack_clicked() : 뒤로 가기 버튼이 클릭되었을 때 호출되는 함수이다. "Main_Lobby" 씬으로 돌아간다.

buttonExit_clicked() : 종료 버튼이 클릭되었을 때 호출되는 함수이다. Unity 에디터에서는 플레이 모드를 종료하고, 빌드된 게임에서는 애플리케이션을 종료한다.

buttonClose_clicked() : 닫기 버튼이 클릭되었을 때 호출되는 함수이다. Hammer UI 패널을 비활성화한다.

OpenUIPanel(InputAction.CallbackContext context) : UI 패널을 열기 위한 입력을 처리하는 함수이다. 키가 입력되면 Hammer UI 패널을 활성화한다.

3. SelectPanel.cs

개요 : 이 스크립트는 스테이지 선택 패널의 동작을 제어한다. 좌우 버튼을 통해 스테이지를 선택하고, 선택한 스테이지의 정보를 업데이트하여 표시한다. 또한, 게임 시작 버튼을 클릭하면 해당 스테이지 씬으로 이동하며, 뒤로 가기 버튼을 클릭하면 레벨 선택 패널로 되돌아간다.

변수

buttonLeft, buttonRight, buttonBack, buttonStart : 게임 오브젝트에서 연결된 UI 버튼을 나타내는 변수이다.

stageText : 스테이지 이름을 표시하는 텍스트 오브젝트이다.

selectPanel, chooseLVPanel : 스테이지 선택 패널과 레벨 선택 패널을 나타내는 게임 오브젝트이다.

stage1, stage2 : 스테이지 미리보기 이미지에 사용되는 스프라이트이다.

stageImage : 스테이지 미리보기 이미지를 나타내는 게임 오브젝트이다.

stageImageImageComponent : 스테이지 미리보기 이미지의 Image 컴포넌트이다.

currentStage : 현재 선택된 스테이지의 인덱스이다.

stageCount : 전체 스테이지 수이다.

previewSprites : 스테이지 미리보기 이미지에 대한 sprite 배열이다.

함수

SetButtonEvent() : 버튼들의 클릭 이벤트를 설정하는 함수이다.

MoveToGame() : 게임 시작 버튼이 클릭되었을 때 호출되는 함수이다. 선택된 스테이지에 따라 "Stage_1" 또는 "Stage_2" 씬으로 이동한다.

LoadingStage1(), LoadingStage2() : 선택된 스테이지에 대한 씬 로딩을 처리하는 코루틴 함수이다.

ButtonBackClicked() : 뒤로 가기 버튼이 클릭되었을 때 호출되는 함수이다. 레벨 선택 패널을 활성화하고, 스테이지 선택 패널을 비활성화한다.

ButtonLeftClicked() , ButtonRightClicked() : 좌우 버튼이 클릭되었을 때 호출되는 함수이다. 이전 또는 다음 스테이지를 선택한다.

change_stage(int stage) : 스테이지 변경을 처리하는 함수이다. 스테이지 미리보기 이미지와 스테이지 이름을 업데이트한다.

set_stage_text(int stage) : 스테이지 이름을 설정하는 함수이다.

set_stage_image(int stage) : 스테이지 미리보기 이미지를 설정하는 함수이다.

buttonExit_clicked() : 종료 버튼이 클릭되었을 때 호출되는 함수이다. Unity 에디터에서는 플레이 모드를 종료하고, 빌드된 게임에서는 애플리케이션을 종료한다.

4. FollowObject.cs

개요 : 이 스크립트는 UI 패널과 메뉴얼 패널을 카메라에 따라 이동 및 회전시키는 역할을 한다. 이를 통해 사용자는 카메라의 시점과 관련된 UI 요소들을 항상 시야에 맞추어 볼 수 있다.

변수

mainCam : 메인 카메라를 나타내는 게임 오브젝트이다.

UIpanel : UI 패널을 나타내는 게임 오브젝트이다.

manualPanel : 메뉴얼 패널을 나타내는 게임 오브젝트이다.

함수

Update() : UI 패널과 메뉴얼 패널을 메인 카메라의 위치와 회전에 맞추어 이동시키고 회전시킨다. `UIpanel.transform.position` 및 `manualPanel.transform.position` 은 메인 카메라 위치에 상대적인 위치를 설정하여 패널을 이동시킨다. `UIpanel.transform.rotation` 및 `manualPanel.transform.rotation` 은 메인 카메라의 회전을 설정하여 패널을 회전시킨다.

5. ChoosLVPanel.cs

개요 : 이 스크립트는 레벨 선택 패널의 동작을 제어한다. 메뉴얼을 표시하고 닫을 수 있으며, 게임 시작을 준비한다. 레벨 1 또는 레벨 2를 선택하면 해당하는 스테이지 선택 패널로 이동하게 된다. 종료 버튼을 클릭하면 Unity 에디터에서는 플레이 모드를 종료하고, 빌드된 게임에서는 애플리케이션을 종료한다.

변수

chooseLVPanel, selectPanel, FirstPanel, ManualCube : 게임 오브젝트에서 연결된 패널들을 나타내는 변수이다.

stageImage : 스테이지 이미지를 나타내는 게임 오브젝트이다.

currentLevel : 현재 선택된 레벨을 나타내는 변수이다.

level1Button, level2Button, exitButton : 게임 오브젝트에서 연결된 UI 버튼들을 나타내는 변수이다.

startButton, lobbyManualButton, closeManualButton : 게임 오브젝트에서 연결된 UI 버튼들을 나타내는 변수이다.

함수

ShowManual() : 메뉴얼을 표시하는 함수이다. 메뉴얼을 보여주는 큐브를 활성화하고, 관련 버튼을 숨긴다..

CloseManual() : 메뉴얼을 닫는 함수이다. 메뉴얼을 보여주는 큐브를 비활성화하고, 관련 버튼을 다시 표시한다.

ReadyToPlay() : 게임 시작을 준비하는 함수이다. 첫 번째 패널을 비활성화하고, 레벨 선택 패널을 활성화한다.

choose_level1(), choose_level2() : 레벨 1 또는 레벨 2를 선택하는 함수이다. 스테이지 선택 패널을 활성화하고, 레벨 선택 패널을 비활성화한다.

buttonExit_clicked() : 종료 버튼이 클릭되었을 때 호출되는 함수이다. Unity 에디터에서는 플레이 모드를 종료하고, 빌드된 게임에서는 애플리케이션을 종료한다.

6. SoundManager.cs

변수

eSFX : sound effect 에 대한 enum 타입변수이다. PlayEffectSound() 함수의 파라미터로 쓰인다.

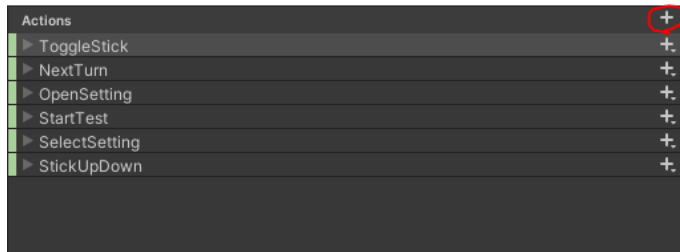
eBGM : 게임의 브금 sound에 대한 enum 타입변수이다. PlayBGM의 파라미터로 사용된다.

7. DataHandler.cs : Player 에 대한 데이터가 있으면 save 함수 호출 시 저장하고, 없으면 Player 에 대한 데이터를 생성한다.

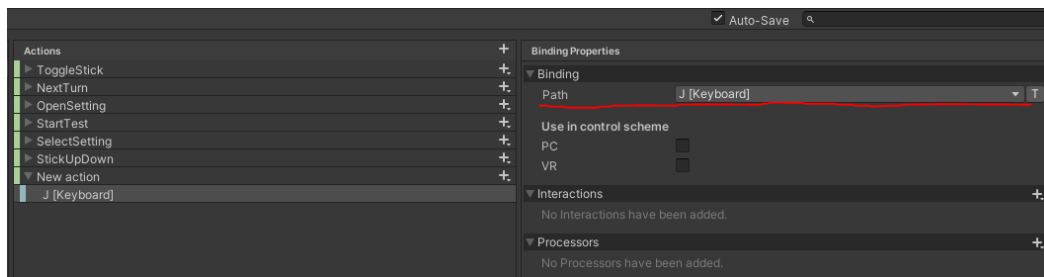
8. SceneShiftManager.cs : Scene 이동에 관한 정보를 담고있다.

4. 조작키 할당방법

1. Script 폴더 내의 PlayerActions 열기
2. Actions 오른쪽의 +버튼을 눌러 새로운 Action 생성



3. Binding에서 새로운 키 할당

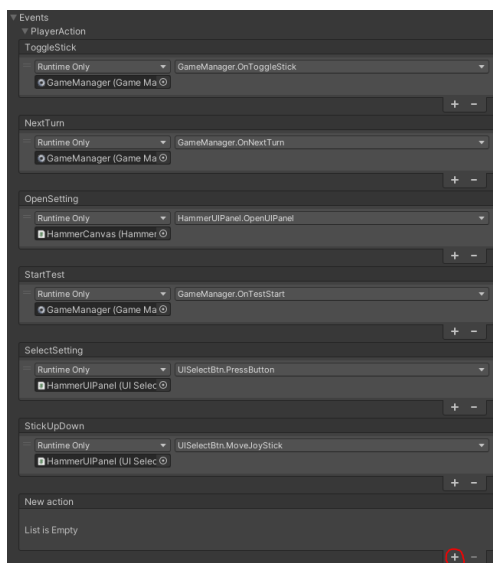


4. 함수 작성시 인자로 InputAction.CallbackContext context를 넣어줌

ex) public void OpenUIPanel(InputAction.CallbackContext context)

5. 에디터 내에서 Player 오브젝트 선택 후 (Stage1 내에선 XR Rig)

Player Input 컴포넌트 – Events – PlayerAction에 추가한 Action +버튼으로 이벤트 추가



6. 호출할 함수를 가진 오브젝트를 넣고 함수가 호출될 수 있도록 설정

