

2023학년도 1학기 전공 캡스톤디자인 최종보고서

교과목명	게임엔진프로그래밍(캡스톤디자인)			
과제명	우당탕탕 마트 털기			
팀명	멀1조		지도교수	조경은
팀원	성명	학과 / 학번	성명	학과 / 학번
	심서현	멀티미디어소프트웨어/2021112467	장선우	멀티미디어소프트웨어/2021112725
	안가영	멀티미디어소프트웨어/2021112497		
	이수연	멀티미디어소프트웨어/2021112494		
KeyWords	마트	중력	시간	랜덤 애니메이션
과제 개요	유니티와 C#을 이용한 3d 게임 제작			
과제 수행 목표 및 필요성	<p>인터넷과 스마트폰의 발전으로 인해 콘텐츠를 쉽게 소비할 수 있게 되면서 현대인들은 짧고 중독적인 콘텐츠를 찾게 되었다. 사람들이 즐겨든 내용의 미디어에 익숙해진만큼 우리도 이런 흐름에 맞추어 짧은 시간 안에 다양한 체험을 할 수 있는 게임을 만들고자 했다. 게임을 단 60초 안에 끝나도록 제작함으로써 굳이 많은 시간을 투자하지 않고 가벼운 마음으로 즐길 수 있는 게임, 60초라는 시간 안에도 충분한 경험을 할 수 있는 충실한 콘텐츠를 가진 게임을 만들고자 했다.</p>			
과제 추진 내용	<p>1.기획과정 / 아이디어 도출</p> <p>1인칭 시점의 게임을 만들고자 했다. 플레이어는 마트 안 진열대 사이를 가로지르며 제한시간 내 최대한 많은, 가장 비싼 물건을 카트 안에 담아 이동하는 것이 목표이다. 상품들의 가격은 전부 다르다. 시간 내에 들어올 경우, 카트에 담긴 상품 가격의 총합이 클수록 승자이다. 가격의 차이를 둬으로써 플레이어들이 보다 지능적인 플레이를 할 수 있는 동기를 만들었다. 맵 안에는 플레이어가 이동하는 카트 뿐만 아니라 다른 npc들 또한 존재한다. 플레이어는 npc들을 피하며 맵 안을 이동해야하며 npc와 접촉할 경우 일정 시간 움직이지 못하는 패널티를 받게 된다.</p>			

2.구현 방법 및 내용

① 시야조작과 카트 이동

게임은 1인칭 뷰로 진행된다. 플레이어는 방향키와, WSAD키를 이용하여 기본적인 움직임을 진행할 수 있다. 마우스 우클릭시 마우스의 움직임을 이용하여 시야를 회전시킬 수 있다.

가장 기본이 되는 player 오브젝트에는, Main camera와 Cart 오브젝트가 있다. 이를 통해, 플레이어가 player 오브젝트를 움직이면 Main camera와 Cart 오브젝트가 같이 움직이게 된다.

: Update 함수를 통해 playerMove 함수와 rotCtrl 함수를 실행시킨다. playerMove 함수를 통하여 입력 받은 키 값만큼 player 오브젝트의 위치를 변경시켜 움직임을 진행하고, Input.GetMouseButton(1)를 이용해 우클릭을 판별하고, 우클릭 중이라면 rotCtrl 함수를 통하여 player 오브젝트를 회전시켜 Main camera를 회전할 수 있도록 한다.

② 오브젝트 잡기

마트에 진열되어 있는 상품을 마우스로 클릭한 채로 드래그하면 상품을 카트로 이동시키는 것이 가능하다. 마우스의 버튼을 놓으면 상품은 그 위치에서 떨어진다. 이때, 선택된 물체는 식별 가능하도록 하얀색 외곽선이 생긴다.

: OnMouseDown 함수를 통해 물체를 잡고 움직이는 기능을 구현한다. 마우스를 드래그 했을때, 마우스 커서의 위치를 받는다. 이때, 물체를 잡아서 가져왔다는 것을 표현하기 위해, z축의 좌표를 1로 지정한다. 월드좌표를 카메라 좌표로 변환하고, 선택된 오브젝트의 위치를 변환한다.

OnMouseEnter OnMouseExit 함수로 물체를 선택하는 기능을 구현한다.함수와 OnMouseEnter 함수를 통해, 물체 위에 마우스 커서가 올라갔을 때 물체의 Material에 외곽선 셰이더 (Unlit/OutLine)을 추가한다. OnMouseExit 함수를 통해, 마우스의 커서가 물체 밖으로 나갔을 경우, Material에서 외곽선 셰이더를 제거한다.

③ 아이템과 인벤토리

아이템 목록은 CVS 파일로 작성한다. CVS 파일에는 아이템의 속성인 아이템번호(NO), 게임 오브젝트 이름(Name), 가격(Price), 오브젝트의 한글이름(KName)에 대하여 작성되어 있다. 아이템을 집어, 카트에 넣으면 Inventory 리스트에 아이템이 추가된다. 아이템이 카트 밖으로 나올 경우, Inventory 리스트에서 아이템이 제거된다. 아이템의 인벤토리는 receipt에 영수증 형태로 볼 수 있다. 약 3000개의 아이템 중 40개 정도는 specialItem이 된다. 이 아이템은, 보너스 아이템으로 다른 오브젝트와 달리 빛이 나며 무조건 가격이 10000원으로 측정된다.

게임 실행시, objPrice.csv 파일을 불러와 itemList를 만든다. 오브젝트에 대한 정보를 받아 Item이라는 클래스로 저장하고, Item들을 List로 만들어 관리한다. 그 후, Scene에 있는 전체 Item을 materialList로 받아와, 그 중 랜덤으로 40개를 정해 태그를 specialItem으로 수정한다.

Cart 오브젝트에, 직육면체 모양의 콜라이더를 추가하였다. OnTriggerEnter함수를 통해, 물체가 카트 콜라이더 안에 들어오면 이 게임오브젝트를 Inventory에 저장한다. 이후 pritItem 함수를 호출하여, 영수증에 이름과 가격을 띄운다.

OnTriggerExit 함수를 통해 물체가 카트 콜라이더 밖으로 나가면 Inventory에서 해당 물체에 관련된 내용을 삭제한다. 그 후, pritItem 함수를 호출하여, 영수증에 이름과 가격을 새로 띄운다.

④ 타이머 기능

게임이 시작되면 왼쪽 위의 타이머가 작동한다. 제한시간은 60초이며, 시간이 흐르면서 타이머 안의 빨간 원이 점점 줄어든다. 제한시간 안에 골인 지점에 들어가지 못한 경우, GameOver Scene을 로드한다.

: If문을 이용하여 일시정지 버튼이 비활성화 되어 있는 경우 시간이 흐르도록 하였다. 현재 게임이 재생 중이라면 흐른 시간 만큼 currentValue가 늘어난다. 타이머 이미지 속 빨간 원은 currentValue 값이 커지는 만큼 줄어든다. currentValue가 60초를 넘기면 게임 오버 씬으로 넘어간다.

⑤ 일시정지 기능

게임 중 오른쪽 위의 일시정지 키를 누르면 재생 키로 바뀌면서 restart와 menu 키가 나타난다. 일시 정지 동안에는 플레이어도 움직일 수 없고, 마트 직원 등 모든 요소가 멈춘다.

: timeScale문을 이용하여 일시정지 상태에서 버튼을 누른 경우 재생 상태로, 반대의 경우 다시 일시정지로 바꿔준다. 일시정지가 되면 restart 버튼과 menu 버튼이 활성화되어 화면에 보이게 된다. 오른쪽 이미지도 일시정지 이미지와 재생 이미지가 바뀐다.

⑥ 미니맵 기능

미니맵은 3인치 뷰로 카트를 보여준다. 미니맵에는 플레이어(카트), 맵의 위치 뿐만 아니라 플레이어가 이동하는 것도 실시간으로 보여야하기 때문에 실제 맵을 카메라를 통해 찍어서 띄워줘야한다 .

: 미니맵을 배치하기 위해 UI-Raw Image를 생성한 후 왼쪽 위에 추가한다. 미니맵 제작을 위해 main camera 이외에 새로운 카메라를 추가한다. 카메라를 만들 때, 원근감을 제거하기 위해 projection을 orthographic으로 변경한다. 배경과 플레이어만 나오도록 culling mask를 따로 설정한다. 미니맵 전용 카메라로 찍은 화면이 아까 만든 UI에 띄우기 위해 프로젝트창에 rendure texture를 만든다. 미니맵 전용 카메라의 target texture에 redure texture를 할당하면 렌더텍스처에 미니맵 전용 카메라로 찍은 화면이 저장된다. 렌더텍스처를 UI에 넣으면 미니맵 카메라로 찍은 화면이 실시간으로 UI에 출력된다. 미니맵 모양을 원으로 수정하기 위해 미니맵이 들어 있는 Raw image 오브젝트의 부모로 일반 이미지 UI를 생성해 원 모양의 스프라이트를 넣는다. 부모 (원) 이미지에 mask 컴포넌트를 추가하면 부모 모양만큼만 보이고 나머지 부분은 보이지 않는다.

⑦ NPC와의 상호작용

1. 위치 이동

이동은 Update()에서 구현한다. 상하 또는 좌우로 움직이는 적의 경우, 이동 상태는 1:아래(또는 오른쪽)로 이동, 2:위(또는 왼쪽)로 이동, 0:멈춰있음, 총 세가지로 구분된다. int형의 mode 변수를 선언하여 현재 모드의 값을 0,1,2 중 하나로 저장한다. 프레임이 update 될 때마다, if문을 통해 어느 mode인지 검사하고, 해당 mode에 맞는 코드가 실행될 수 있도록 한다. 상하로 이동하고, 현재 모드가 1인 적의 경우에는 다음과 같이 구현된다. Awake() 호출 시, 변수 position에는 시작할 때 적의 위치가 저장된다. update가 한번 호출 될 때마다, position의 z 값을 0.4씩 감소시키고, transform.position 값은 이렇게 매번 연산되는 position 값을 가져와서 저장함으로써 적의 위치가 계속 아래로 내려갈 수 있도록 한다. 또한, 적이 바라보는 방향이 자신의 이동 방향과 같아야하므로, transform.forward를 사용해 -Z 방향을 바라보도록 한다. 좌우로 이동하는 적의 경우에는, z가 아닌 x값을 증감시킨다.

2. 적이 벽에 충돌할 시

적은 벽을 만나면 mode를 바꾸어야 한다. 벽에 가까워졌다는 것을 인식하기 위해, 양 쪽의 벽 앞에 box collider를 각각 둔다. 적이 어떠한 물체와 충돌을 감지할 시, 충돌한 물체의 태그를 확인해보고 해당 태그가 Wall1 또는 Wall2에 해당하면, mode를 1 또는 2로 바꾸도록 한다. 이를 위해 두 개의 box collider에 wall1, wall2 태그를 설정한다.

3. 적이 플레이어와 충돌할 시

적이 벽에 부딪히면 방향이 바뀌어야 한다. 벽과의 충돌을 인식할 수 있도록 양쪽의 벽에 각각 Wall1, Wall2 태그를 설정하고, 적의 스크립트는 이를 OnTriggerEnter 함수와 CompareTag로 인식한다. 부딪힌 벽이 Wall1인지 Wall2인지 CompareTag를 통해 태그 비교를 한 후, mode 1 또는 2로 변경한다.

4. 플레이어와 충돌 구현

우선 cart에 박스 콜라이더와 Player 태그를 설정해 적과의 충돌을 감지할 수 있도록 한다. 적이 충돌을 감지하면 stopwatch를 시작하고, bump와 freeze 변수에 true값 전달을 전달한다. update 문에서 bump 가 true임을 감지하여, bump() 코루틴을 시작한다. freeze가 true로 바뀌면, 플레이어 스크립트의 update문 실행이 중단되어, 플레이어는 움직일 수 없다.

bump 코루틴은 첫째, 적의 애니메이션을 걷기(또는 청소, 전화하기) 애니메이션에서, 놀라는 애니메이션으로 전환시킨다. 둘째, mode를 0으로 바꾸어 적이 멈춰있게 한다.

앞서 시작한, 스탱워치가 1초를 넘기면 StopMethod()를 호출해 코루틴을 중단한다. bump 값은 다시 false로, mode와 animation은 다시 이동하는 움직임에 맞춰지도록 설정한다. 또, 시간이 1.5초를 넘으면 freeze값은 false로 바뀌어 다시 움직일 수 있다.

5. 충돌 시 text 효과 구현

충돌 시 현재 적의 위치 근처에 충돌을 알리는 애니메이션을 띄운다. FXposition 변수에 현재 적의 위치 정보를 담고, FXposition을 증감시킨 후, Instantiate함수를 호출해 원하는 위치에 효과가 나오도록 한다.

6. 적의 애니메이션 전환

적이 두 가지 이상의 애니메이션을 수행할 수 있도록 animator controller의 transition에서 bool 변수를 인식한다.

⑧ GAME OVER

주어진 60초 동안 Goal에 들어가지 못하는 경우, 플레이어는 경찰에 붙잡히며, GameOver Scene을

로드한다.

: 60초가 지난경우 SceneManager.LoadScene을 사용해 GameOver 씬을 불러온다. 그 후, restart 버튼을 띄워 새로 시작할 수 있도록한다.

⑨ GAME END -

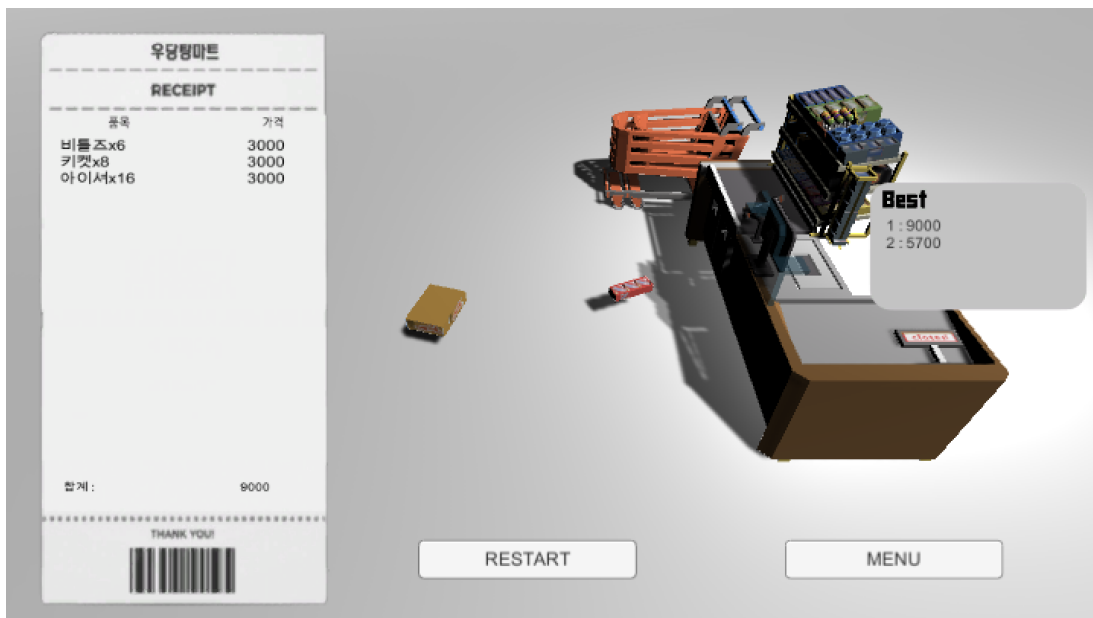
60초 내에 Goal에 들어가면, 플레이어는 도둑질에 성공한다. 카트에 담은 물건이 위에서 아래로 차례로 떨어지며 쌓이는 애니메이션을 보여주면서, 영수증에는 가격과 이름을 띄운다. 모든 물건이 다 정산되면 총 합계가 영수증에 출력된다.

: OnTriggerEnter 함수를 통해 cart 오브젝트가 goal 콜라이더와 닿게되면 SceneManager.LoadScene를 사용해 GameEnd 씬을 불러온다. 이때, 카트 안에 있던 아이템들을 Player 오브젝트 안에 있는 Inventory에 옮긴 후, DontDestroyOnLoad를 사용해, player 오브젝트를 GameEnd씬에서도 사용할 수 있도록 한다.

GameEnd씬으로 넘어가면, Inventory에 있던 아이템들을 objList에 추가한다. Updatd 함수를 통해 2f에 한번씩 카트에 있던 물건이 위에서 아래로 떨어진다. ScriptTxtName에는 물건의 이름이 ScriptTxtPrice에는 물건의 가격이 뜬다. 만약 objList에 있던 아이템들을 전부 계산 했으면, 영수증에 합계를 띄운다.



⑩ 최고기록

	<p>게임 결과를 저장하여 최고기록을 도출한다. 최대 3위까지 최고기록을 볼 수 있다.</p> <p>: 게임 오브젝트 Scores를 만든다. DontDestroyOnLoad로, restart를 해도 Scores는 삭제되지 않도록 설정한다. 게임 엔딩 이후, total 값을 계산하면 순위를 갱신하여 Best text asset 에 띄운다.</p> <p>⑪ 5초 화면 깜빡임 효과</p> <p>제한시간이 10초 남았을 때, 플레이어에게 알림을 주기 위해 화면이 붉은색으로 깜빡인다.</p> <p>: 캔버스에 이미지를 추가하고 (1,0,0,0)으로 색상과 알파값을 설정한다. 스크린에 이미지를 출력해주는 함수를 코루틴으로 구현하기 위해 IEnumerator형 함수를 생성한다. Red 이미지의 투명값을 재선언하는 코드 사이에 yield return waitforseconds 구문을 넣어 깜빡임 효과를 생성한다. 시간이 5초가 지나갈 때부터 깜빡임 효과를 주기 위해 타이머 함수 안에 showRedscreen()함수를 선언한다.</p> <p>⑫ 힌트 버튼</p> <p>게임 시작 메뉴에서 오른쪽 아래 전구 모양 버튼을 클릭하면 게임에 대한 메뉴얼을 확인할 수 있다. 일정 시간이 지나면 힌트 이미지는 자동으로 사라진다.</p> <p>: SetActive 함수를 통해 게임 시작 직후에 힌트 이미지를 비활성화 시킨다. onClick 함수와 AddListener 함수를 통해 힌트 버튼이 클릭되면 ShowHint 함수가 실행되도록 한다. ShowHint 함수는 힌트 이미지를 활성화 시키고 5초 뒤 HideHint 함수를 통해 이미지를 비활성화 시킨다.</p>
<p>과제 결과물</p>	<p>시작화면 ui 구성은 게임로고, 좌측하단 start버튼, 우측하단 hint버튼으로 이루어져 있다. hint버튼을 누를 경우, 간단한 게임컨셉에 대한 소개와 게임팁을 확인할 수 있다. start버튼을 누를 경우, 타이머도 동시에 작동하기 시작한다. 제한 시간 이내에 플레이어는 골인 지점인 계산대까지 도착해야 하며, 도착에 실패할 경우 게임오버가 된다. 진열대에 있는 수많은 상품들을 마우스와 키보드 조작을 통해 카트에 옮겨 담아야하고 상품들의 가격은 전부 다르게 설정되어 있다. 게임을 플레이할 때 마다 speical item이 랜덤으로 설정되어 해당 아이템의 가격에는 10000원이 할당된다.카트 안에 진열대 상품을 담을 때마다 좌측에 있는 영수증 모양의 ui에 상품의 이름과 가격이 실시간으로 작성된다. 제한시간이 10초 남았을때부터 플레이어 게임 화면은 빨간색으로 깜빡이기 시작하며 시간이 얼마 남지 않았음을 보여준다. 인게임에는 일시정지 기능을 추가하여 게임을 잠시 멈추는 것도 가능하다.</p>





<p>활용방안 및 기대효과</p>	<ol style="list-style-type: none"> 1. 게임 배경을 다양하게 바꿈으로써 마트 홍보 효과 2. 중력이 작용하는 물체 게임을 통해 물리학의 기초를 이해하고 학습할 수 있다. 3. 공간 인식 능력 강화하는데 유용. 게임을 통해 물체의 위치, 움직임, 속도를 파악할 수 있다.
<p>참고 문헌</p>	
<p>팀 활동과 팀원 별 역할</p>	<p>심서현: 시야조작과 카트 이동 / 아이템과 인벤토리 / GAME END / 최고기록 안가영: 타이머 기능 / 일시정지 기능 / GAME OVER / 힌트 버튼 이수연: 미니맵 기능 / 5초 화면 깜빡임 효과 장선우: NPC와의 상호작용 (애니메이션)</p>
<p>성 과</p>	<p>논문 발표</p>

	특허 출원	
	SW 등록	
	시제품, 영상물 제작	
	외부대회 출품	
	기타	
결과물 사진		
	시작 화면	플레이 화면