|  |
| --- |
| (한국정보기술연구원) 게임콘텐츠 팀 프로젝트 보고서  **Dark Forest**  게임16기 7조  (Dark Forest) |

조장:

김용식(데이터처리)

팀원:

이유빈(맵 디자인, 시야)

박경수(Actor)

최진영(UI, Sound)

1. 목차

[1] 게임 소개

[2] 게임의 목적

[3] 게임의 목표

[4] 개요

[5] 핵심 플레이 싸이클

[5-1] 게임 대전 구조

[[5-2] 전체 순환 구조](#_Toc487817966)

[6] 게임 조작 방법

[6-1] 조작 방법

(1) 게임 조작 설명

[6-2] 전투 조작

[7] 기능

[7-1] 조합창

[7-2] 맵

(1) 전체맵

[7-3] 아이템

(1) 일반 아이템

(2) 조합 아이템

[9] 게임 로직 구현 설명

[10] 프로젝트 후기

# [1] 게임 소개

TopView Horror RPG게임



# [2] 게임의 목적

스팀플랫폼 DarkWood의 게임 방식의 특성인 TopView시점의 시야제한과 분위기를 부각시키려 하였다.

# [3] 게임의 목표

섬에 고립된 플레이어가 Quest를 부여받아 아이템을 수집, Creature를 물리치고 최종 Boss를 제거하여 섬을 탈출한다.

# [4] 개요

|  |  |
| --- | --- |
| 항목 | 내용 |
| 제목 | Dark Forest |
| 장르 | TopView Horrer RPG |
| 플랫폼 | PC, Mobile |
| 이용등급 | 12세이상 이용가 |
| Test 스마트폰 버전 | Android 6.0 |

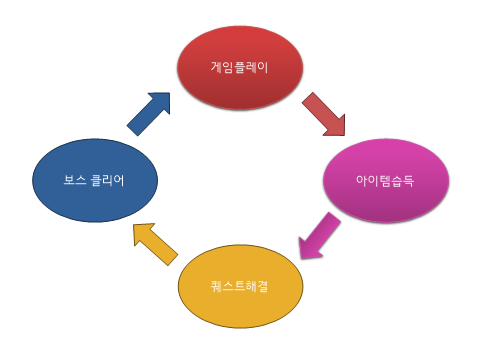
# 

# [5] 핵심 플레이 싸이클

## [5-1] 게임 진행 구조

|  |  |
| --- | --- |
| 순서 | 설명 |
| 아이템 획득 | 맵상에 스폰된 주인공 주위에서 활과 검을 획득한다. |
| 재료 조합 | 나뭇가지와 돌을 조합하여 화살을 제작한다 |
| 보스 처치 | 지금까지 해결한 퀘스트를 바탕으로 보스를 처치한다 |
| 탈출 | 보스 처치 후 섬을 탈출한다. |

## [5-2] 전체 순환 구조



|  |  |
| --- | --- |
| 순서 | 설명 |
| 무기 획득 | 무기를 획득하여 장착한다. |
| 재료 조합 | 활을 쏠 수 있는 화살을 제작한다 |
| 보스 처치 | 퀘스트를 통하여 얻은 아이템을 사용하여 보스를 처치한다. |
| 탈출 | 보스 처치 후 섬을 탈출한다. |

# [6] 게임 조작 방법

## [6-1] 조작 방법

PC : 방향키 (W, S, A, D), 마우스 클릭

Mobile : 조이패드 MultiTouch사용

### (1) 게임 조작 설명



|  |  |
| --- | --- |
| 구분 | 설명 |
|  | 플레이어의 움직임을 조작하는 이동버튼.(조이스틱) |
|  | Enemy를 공격하는 버튼 |
|  | 메뉴 및 퀵슬롯 버튼 |

## [6-2] 전투 조작

|  |  |  |
| --- | --- | --- |
| 이미지 | 구분 | 내용 |
|  | 시점 | 탑뷰 |
| 타겟팅 | Enemy |
| 전투 컨셉 | 시야가 제한된  사정거리 내의  근접 또는 원거리 무기를 사용하여 적을 처치한다. |

# [7] 기능

## [7-1] 조합

### (1) 조합창

User는 획득한 아이템 중 인벤토리 조합창 내의 조합 가능한 Item만 볼 수 있다.

|  |
| --- |
| [ 조합창 ] |
|  |

특정재료들만 조합할 수 있다.(나뭇가지, 돌)

## [7-2] 맵

### (1) 전체맵



User가 고립된 상황, 제한된 시야에 몰입하도록 맵 디자인을 제작하였다.

## 

## [7-3] 아이템

### (1) 일반 아이템



게임 시작 시, Character가 발견하도록, 주위에 기본 아이템 외의 생존에 필요한 Postion, 소비형 원거리무기 조합재료를 획득 및 이용.

|  |  |  |
| --- | --- | --- |
| 분류 | 종류 | 설명 |
| 무기 | 활 | 화살이 소지 시, 원거리 공격이 가능하다. |
| 검 | 근거리 공격이 가능하다. |
| 아이템 | HP포션 | Character의 HP를 회복 |
| STAMINA포션 | STAMINA를 회복하여 공격 및 달리기 가능 |

### (2) 조합 아이템



재료 아이템을 조합하여 생존력을 높일 수 있다.

|  |  |  |  |
| --- | --- | --- | --- |
| 분류 | 종류 | 설명 | |
| 재료 | 나뭇가지 | | 조합에 필요한 아이템\_1 |
| 돌 | | 조합에 필요한 아이템\_2 |
| 조합아이템 | 화살 | 두 가지의 조합아이템 화살을 사용하여 모두 소비직전 까지 원거리 공격을 할 수 있다. | |

# [8] 게임 로직 구현 설명

|  |  |  |
| --- | --- | --- |
| 분류 | 기능 | 구현 로직 설명 |
| UI | 퀵슬롯 | NGUI를 사용하였고 Onclick 함수를 사용하여 클릭시 ItemManager한테 ThrowEvent로 아이템을 사용하였다고 전달해주고 label 개수를 최신화 한다. |
| 인벤토리 | 아이템을 먹었을시 ItemManager에서 Inven 정보를 저장하고있는 Dictionary를 검사 후 아이템이 있을시 아이템 프리펩을 Resources에서 불러오고 클릭시 장비는 착용 여부 나머지 아이템들은 퀵슬롯에 등록할수있게 하였다. |
| 조합창 | ItemManager에서 Inven정보를 저장하고있는 DIctionary에서 재료타입만 리스트에 넣은후 재료아이템만 보여주고 아이템 클릭시 재료창에 올릴수있으며 조합하기를 눌렀을시 ItemManager한테 ThrowEvent로 조합을 하였다고 전달해준다. |
| 조이패드 | NGUI에 방향키패드는 Onpress함수를 사용하여 클릭시 눌렸다는 bool판정과 Drag함수를 사용하여 최소의 움직일시에는 Vector3.zero로 막고 그이외의 움직임은 큰원안에서만 움직일수 있도록 하였으며, 플레이어한테 bool값과 움직인 포지션을 전달해준다.  공격패드는 OnPress함수로 눌렸는지 안눌렸는지 검사후 bool값을 전달해준다. |
| 로딩창 | 오브젝트 풀을 사용하여 씬이 넘어갈 때 활성화 비활성화로 로딩창을 보여주며 게임씬으로갈때에는 게임씬로드가 다되면은 로딩이 끝난다. |
| 데이터처리 | 아이템 | 모든 아이템에 정보를 관리하며 아이템을 먹었을시 인벤토리 아이템을 관리하고 장착하였을때는 장착아이템슬롯 관리를 따로하며 아이템 사용시에는 UI에서 아이템을 사용하였다고 받은 뒤 캐릭터한테 정보를 보내준다. |
| 퀘스트 | JSON을 사용하여 데이터를 받아오고 옵저버 패턴을 사용하여 퀘스트를 순서에 따라서 처리하였다. |
| Scene | Title -> Game  Game -> Title | 타이틀 화면에서 게임시작을 누르면 씬 전환을 한다. |
| 사운드 | BGM | 씬이 바뀔때마다 SoundManager에서 씬이 변경될 때 BGM을 변경해주는 함수를 호출하여 BGM이 변경된다. |
| Player | 생성, 삭제 | Actor의 하위 객체인 Player와 Enemy를 생성할 때 ActorManager를 통해 생성하여 리스트를 통해 관리하고 TeamType를 통해 Player와 Enemy를 구분한다 |
| status | JSON을 통해 Actor들의 status 를 받아오고 HP가 감소하거나 회복할 때, SP를 소모하거나 회복할 때 ThrowEvent를 통해서 각각의 cost를 소모한다 |
| Animator | Actor의 상태가 변경될 때 상태에 맞는 Animator를 실행하고 타겟형 공격의 경우 Animator의 진행 상황속도에 맞춰서 행동을 실행한다 |
| 이동 | GetAxisRow를 이용해서 wasd로 이동할 방향을 입력받고 Rigidbody.moveposition을 이용해 캐릭터의 위치를 이동시킨다. |
| 회전 | 마우스 포지션에 카메라에서 ray를 이용하여 벡터를 구하고 그곳을 바라보게 회전하며, y값에는 플레이어 위치의 y값을 대입해서 y축이 아닌 다른 방향으로 회전하는 것을 방지한다. |
| Enemy | 공격 | 공격 버튼을 누를 때 활을 착용하고 있다면 활을 이용한 공격을 실행하고, 검을 착용하고 있다면 검을 이용한 공격을 실행한다. |
| 생성 | Player가 EnemyGenerator 오브젝트의 충돌 범위 안으로 들어갈 때 지정된 위치에서 Enemy를 생성하고 한번 Enemy를 생성한 Generator는 더 이상 생성하지 않는다 |
| 일반 몬스터 | Player가 자신의 추적범위 안에 접근하면 NavMeshAgent를 이용하여 찾아간다. 자신의 Collider와 타겟의 Collider의 반지름을 계산하여 충분히 접근했다면 NavMeshAgent를 종료하고 타겟을 공격한다. 공격은 Player에가 타게팅형 공격을 한다. Player가 공격이 유요한 시점까지 진행되기 이전에 공격범위를 벗어난다면 공격을 멈추고 다시 따라간다. 타겟이 추적범위 밖으로 벗어난다면 추적을 종료하고 자신이 생성된 위치로 돌아간다.  추적 범위 밖에서 Player가 공격한다면 공격한 위치를 찾아가고 Player를 발견하지 못한다면 다시 자신이 생성된 위치로 돌아간다 |
| 보스 몬스터 | 보스 몬스터가 생성되면 Player와 거리에 관계없이 추적하고, 일정 범위 안에 들어왔을 때 Player의 거리에 따라 근접공격, 점프공격, 원거리 공격을 실행한다  근접공격의 경우 일반 몬스터와 동일하고 점프공격은 점프를 하는 선행 애니메이션 이 후 플레이어의 방향을 보게하고 몬스터의 정면으로 Rigidbody.velocity를 이용해 점프하고 점프 도중에 충돌하는 대상에게 피해를 준다. 원거리공격은 Player의 방향을 기준으로 여러 각도로 여러 개의 투사체를 발사한다. |
| Weapon | Bow | Player가 활을 장착 하고 있을 때 활을 이용한 공격을 실행하고 처음 공격버튼을 누르면 활을 장전하고 누르고 있는 시간에 비례하여 활 시위를 당기는 힘을 증가시킨다. 이 때 일정수치 이하일 때는 화살을 발사하지 않으면 최대 힘도 제한해 둔다. 공격버튼을 뗄 때 화살을 생성하여 발사한다 |
| Arrow | 화살이 생성되면 rigidbody.velocity를 이용하여 전방으로 날아가며 velocity의 방향으로 자신을 회전시킨다. Enemy와 충돌 할 때 피해를 주고 지형과 충돌할 때는 지형에 박혀 흔적이 남는다 |
| Sword | Player가 검을 착용하고 있을 때 근접 공격을 실행한다. 공격시 시작되면 애니메이션의 속도에 맞춰서 무기와 동일한 위치에 Collider를 생성하고 Collider에 충돌한 Enemy 피해를 준다. 공격이 종료되면 Collider를 삭제하여 공격중이 아닐 때 피해를 주는 것을 방지한다 |
| 시야효과 | FogOfWar | User가 바라보는 곳 외의 Enemy, Item을 화면상에서 보이지 않게 해준다.  반지름의 길이와 시야각에 따라 User의 시야가 정해지고 정해진다.  meshResolution의 값이 높아질수록 vertex의 개수가 늘어 Obstacle에 의해 가려지는 시야판정을 더 섬세하게 조절가능  Obstacle, Enemy, Item을 Layer지정에 따라 판단하여 List에 추가 후 시야판정에 따라 보일지 안보일지 Renderer 처리에 따라 기능제공  FogProtector를 이용하여 vison시야를 분리하여 시야만큼 덮어씌운다. |
| TreeObject | TopView 형식으로 기획했기 때문에 나무에 의해Character가 가려지거나 시야를 방해하면 Play에 지장을 줄 수 있기 때문에 Collider 범위판정을 하여 User Character가 가까이 가면 Tree Object가투명화가 된다. |

# [10] 프로젝트 후기

김용식 : 능력에 맞는 일정 배분과 각자의 코드를 취합해서 하나의 프로그램으로 만들어 내는 과정이 쉽지 않음을 알게 되었습니다. 이를 통해서 팀 단위 프로젝트에서 소통의 중요성을 다시 한번 새기게 되었습니다.

이유빈 : Dark Wood 게임을 Reference로 참고하여 제한된 시야의 답답함과 어두운 분위기, Crafting, Monster와의 전투를 살리고자 하였다.

회의를 통하여 각자 역할을 정하였고, 생각보다 참고할만한 자료가 부족하기도 했다.

한정된 시간안에 맡은 기능을 구현해야 했고 합치는 과정에서도 애를 먹고 어디서 문제를 해결해야하며, 팀원들이 구현해놓은 기능들을 알아보기에 부족했던 것 같다.

배경, 분위기를 표현하는데 아쉬움이 많았고 결국 초기에 기획했던 아이디어을 집어넣지 못해 아쉬웠다.

이번 프로젝트에서 상대방의 코드를 이해하고, 각자의 작업물을 하나로 합치는 과정에서 예상치 못한 bug를 해결하는 과정에서 내가 더 채워야할 것이 무엇이고 서로간의 의사소통이 중요하다는 것을 느꼈다.

최진영 : 혼자서 코드를 짤때에는 내 코드에 맞게 짜면되지만 팀으로 하다보니 만약을 생각해서 코드를 구현해야하는 것과 팀원과의 커뮤니케이션이 중요하다는 것을 알게 되었고 코드를 짤 때 좀더 생각하게 되어서 공부가 되었습니다.

박경수 : 처음 협업을 해서 게임을 만들다 보니 다른 사람과 공유하기 쉽도록 코드를 만드는 방법도 서툴렀고. GitHub를 이용하여 로컬과 서버의 코드를 병합하는 과정도 쉽지 않았다. 여러 사람이 의견을 모아 기획을 하다 보니 의견이 충돌하는 부분도 있었고, 처음 기획의도와 같게 진행하는 것도 쉽지 않았다. 하지만 혼자 구현하기 힘들었던 기능이나 쉽게 발견하지 못할 버그들도 여러 사람이 같이 테스트하고, 고민하면서 해결 할 수 있었다. 팀 프로젝트를 진행하면서 좀더 구조화된 코드를 만드는 방법, GitHub를 이용하여 코드를 병합하고 관리하는 방법 등을 배울 수 있는 좋은 경험이 되었다.