게임 개발 보고서

‘리볼버 RPG’

|  |  |
| --- | --- |
| 작품명 | 미정 |
| 과목명 | 융합소프트웨어종합설계 |
| 지도교수 | 전병태 교수님 |
| 제출일 | 2024mmdd |
| 팀원 | 안영남, 민병수, 이준희 |

**1.프로젝트 이름 및 개요**

1.1개요 및 프로젝트 동기

**2.개발 구상**

2.1 게임 내용 구상

2.2 플레이 시스템 특징

2.3 도표및 스케치

2.4 사용 도구, 협업 관리

**3.다른 작품들과 비교**

3.1 영감을 받은 게임

**4.구현 일정 및 과정**

4.1 개발 과정

**5. 구현 과정 보고서**

**6.결론과 예측**

**7.참고문헌**

**1.프로젝트 이름 및 개요**

가제 – 리볼버 RPG

장르 – 턴제 RPG, CRPG, 로그 라이트

컨셉 – 사이버 펑크를 배경으로 하는 턴제 전환 RPG

1.1개요 및 프로젝트 동기

현 시대에 게임 개발은 상당히 간단해지고 쉬운 일이 되었다. 간편한 게임 엔진들이 제한적이지만 무료로 풀리기 시작했고 정보의 시대이니만큼 개발에 대한 정보를 인터넷에서 정말 쉽게 얻을수 있기 때문에 게임 개발에 대한 접근성이 낮아졌다. 이로 인해 다양한 아이디어를 가진 1인 개발자 및 적은 인원으로 개발되는 인디 게임이 늘어났다. 특히 인디 게임의 경우 개발자의 생각이나 게임 디자인을 직접적으로 반영할수 있다는 점에서 매력적이다. 어떻게 보면 비디오 게임은 종합 예술이라고 할수 있지만 거대 개발사에서 만드는 경우 개인의 생각이나 의도를 직접 담기는 어렵고 자본과 타협도 해야하기에 그런 점이 상대적으로 많이 드러나지 않는다.

하지만 그렇다고 할지라도 인디 게임의 수준이 높다고 말할수는 없다. 일단 겉으로 보기에도 대부분의 인디게임은 적은 개발 시간, 인력으로 인해 기업구조의 개발사보다 그래픽이나 사운드, 시스템 구현면에서 뛰어나기 어렵다. 또한 인디 게임의 경우 피드백이 적다는 큰 단점이 있다. 따라서 잘못된 게임 디자인으로 인해 게임 자체의 구조가 망가지기도하고, 개발자의 적은 게임 플레이및 개발 경험으로 인해 플레이어에게 불쾌감을 선사하거나 게임의 본질인 ‘재미’를 아예 배제하고 지나치게 예술적인 움직임을 보이며 플레이어들이 이해하기 어려운 수준의 게임을 만드는 경우가 대부분이다.

따라서 게임을 좋아하고 자주 플레이했던 우리들이 직접 게임을 개발하면서 어떤 점이 게임의 개발 과정에 필수적인지, 무엇이 게임을 재밌게 만드는지, 게임에 가치를 어떤식으로 적절하게 부여할것인지 탐구 하고자 한다. 우리들도 개발 경험이 적은만큼 훌륭한 개발자라고 할 수 없겠지만 이 과정을 진행함에 있어서 앞서 말했던 인디 게임의 문제점을 계속 인지하고 지속적인 자체 피드백을 통해 문제가 보인다면 과감하게 수정하여 그런 문제점들을 최소화하고자 한다.

**2.개발 구상**

사용엔진:유니티

그래픽 → 2D 도트

3인칭 시점 탑뷰

데이터 베이스 → 개발중 구축

2.1 게임 내용 구상

\* 게임 플레이 구상

- 실시간-턴제 혼합 RPG

필드에서는 자유롭게 이동이 가능하지만 상호작용을 통해 턴제로 전환

- 리볼버 시스템

턴제 플레이시 상호작용시 사용 가능한 능력을 리볼버의 탄창으로 표현

턴제에서 플레이어는 리볼버의 탄창 수만큼 상호작용 또는 능력을 사용할수 있음

탄을 다 사용시 상대의 턴 진행

-전투는 아이템에 의존하는 시스템

아이템을 얻고 이를 통해 전투에서 활용하는 시스템

전투는 확률과 운이 꽤 큰 영향이 가도록 만들것

- 전투에 활용하는 아이템은 컨셉에 맞게 대부분 소비형

리볼버를 사용하는 방식으로 설정했기에 각 탄환을 소비하는 형태로 진행

슬레이 더 스파이어에서 카드를 소모하고 발더스 게이트3가 자원을 활용하는 방식으로 전투를 진행하는 것처럼 이 게임은 탄환을 소비하면서 전투를 진행함

- 재장전시 상대 턴 진행

앞서 언급했듯이 재장전을 하면 플레이어의 턴이 종료 됨

NPC들의 턴 진행

- 무기 시스템

무조건 리볼버만 있는게 아니라 자동 권총이나 현대식 소총같은 무기만 아니면 다 등장해도 될것 같음

그에 따라 UI가 바뀌게끔 만들어야 함

ex)쌍엽 산탄총(더블배럴 산탄총)같은 경우 딱 2발만 사용 가능하되 성능은 좋게 만들어서 하이리스크-하이리턴 플레이를 원할경우 차용 가능 → 인터페이스가 알아서 더블배럴 형태로 바뀌어야함

- 필드-던전 구성

필드에서는 NPC의 밀도를 낮게만들고 던전은 밀도를 높이고 긴 시간을 투자해야 클리어할수 있는 방식

던전은 셀-셀 형태지만 실시간 이동으로 구성됨

\* 게임의 시스템과 구조

- 엔진과 언어

2D 도트 그래픽에 어울리고 개발자들의 숙련도를 고려하여 유니티로 결정

유니티 엔진은 C#/C++ 기반

- 진행상황 저장 방식

JSON 파일 형태로 외부에 저장

일단은 특정 지점에 도달시 저장하는 형태를 목표

- 데이터 베이스

로컬 데이터 베이스를 활용 SQLite를 이용하여 게임 오브젝트, NPC, 아이템의 효과나 정보를 저장함

- 컴포넌트 활용

유니티에서 지원하는 컴포넌트를 활용하여 비슷한 오브젝트나 NPC를 재사용하여 개발시간을 앞당김

- 자원 관리

\*비동기 로딩을 이용하여 이후에 로딩이 필요한 경우 백그라운드에서 불러오기

\*지연 로딩을 이용하여 맵의 특정 부분만 로딩되게 만들어 자원 활용 최적화

- 오류 처리

로그를 이용하여 문제가 되는 부분을 판단할수 있고 이후 디버깅시에도 활용

\* 그래픽 및 사운드

-배경 설정

→ 추가적으로 정립이 더 필요하지만 일단 방향성은 사이버 펑크

- 도트 그래픽 활용

도트 그래픽은 에셋을 사용하는것에 비해 더 독특한 느낌을 줄수 있음

대신 시간 투자와 노력 필요

- 배경 창작 이후에 적절하게 도트 그래픽을 만들 필요가 있음

아예 2D 그림을 만들기보다는 3D 모델링을 2D처럼 보이게 만드는 그래픽 기법 사용하는것이 좋아보임

- 조작과 네비게이션 시스템

일단은 WASD와 포인트 앤 클릭 모두 사용하는 방식이지만 구현 이후 다시 고려해볼 필요가 있음

네비게이션 시스템도 조작 방법을 정립한 후에 다시 생각해야함

- 사운드

→ 이후에 추가적인 회의 필요함

도트 그래픽을 사용할 예정이기에 8비트 사운드를 사용하는것도 나쁘지 않아보임

\* 추가 목표

- RPG요소

능력치 시스템을 사용해서 기본적인 성장시스템을 구현

전투는 아이템에 영향을 많이 받지만 능력치를 통해 사용가능한 아이템에 제한을 거는 방식

기본적으로 정립한 능력치

행운, 매력, 힘, 손재주, 민첩, 지구력

행운 – 아이템 드랍율이나 치명타

매력 – 화술->대화시스템

힘 – 아이템 사용에 대한 제약 해제(힘이 낮으면 무거운 아이템을 사용못하는 방식)

손재주 – 전투 보조

회피 – 전투 보조

지구력 – 체력과 들고 다닐수 있는 아이템 수나 무게

캐릭터 자체에 대한 설정이나 커스터마이징은 데모 완성이후 구현 가능 판단시 구현

- 전투 난이도와 레벨 디자인 방향성

명성 증가에 따라 적의 수와 체력같은 능력치를 올리는 방향

명성은 죽으면 떨어지기에 그에 맞춰 적절히 잘 구성되어야 하며 플레이어가 강해졌다는것을 어느정도는 체감할수 있게끔 만들것

- 명성 시스템:다른 게임의 레벨과 비슷한 개념

명성에 따라 아이템을 얻을수 있는 임무를 맡게 해서 더 강력해짐

명성은 단순히 임무에만 연관이 있는것이 아니라 특정 명성마다 보너스를 제공하는 방식

대화지문같은 것에도 영향을 미치며 전투에서 패배하면 명성이 떨어지고 아이템을 떨굼

- 상호작용 지향 게임

대화, 물건 사용같은 사소한 부분에도 턴제로 전환하되 이런 상호작용의 난이도는 낮고 빠르게 진행 → 전투시에는 더 힘들지만 다른 상호작용을 통해 전투를 줄이는 방법을 만들고, 전투 자체도 자주 일어나지는 않게끔 설계하는것이 중요

2.2 플레이 시스템 특징

- 전투 시스템

전투시에 리볼버 탄창만큼만 턴을 진행할수 있게 제약을 걸어둠

전투 시스템 자체는 이전의 다른 턴제 게임들 (영향받은 작품들)과 크게 다르지 않음

예를들어 슬레이 더 스파이어에서 카드를 탄환으로 변환시킨것과 비슷함

다만 모든 탄환이 소모품이고 턴을 마치기 전에 장전할 탄환을 선택하기 때문에 선택의 중요성이 부각되고

빠른 대처가 어렵기에 플레이어에게 선택에 고민을 유도함

또한 덱 빌딩 요소를 차용하여 탄환의 연계를 통해 플레이어에게 다양한 조합을 만들수 있음

예를 들어 기름을 쏘아 뿌려놓고 이후에 불을 붙여 여러 적에게 피해를 준다던가

우호적인 NPC를 돕기 위해 냉동탄을 쏘아 잠깐 적을 움직이지 못하게 한다던가 등등

- 패배 페널티

고전적인 RPG인 로그의 요소를 어느정도 차용함 → 사망시 막대한 페널티

다만 죽는다고 게임이 초기화 되는 로그및 로그 라이크 게임들과 다르게 아이템을 잃어버리는 정도이며 이는 해당 장소로 돌아가 다시 되찾거나 게임 내 시스템을 활용하여 떨어트리는 아이템의 수를 줄이는 방식으로 보완을 할 수 있음

- 맵의 구성

맵은 크게 3가지로 구분

\* 거점 : 일반적으로 적대적인 NPC가 없음, 임무 수락 가능, 물건 매매 가능

\* 필드 : 거점 밖의 필드는 적대적인 NPC 존재 가능 대놓고 적대적인 NPC도 존재하지만 대화나 행동을 통해 적대적으로 바뀌거나 우호적으로 대하는 NPC도 존재

필드의 NPC 밀도는 높지 않고 필드에서 좋은 보상을 얻기는 어려움

필드에서도 몇몇 우호적인 NPC와 물건 매매 가능

\* 던전 : 필드에서 진입 가능

- 던전의 구성

기본적으로 여타 턴제전환게임 처럼 넓은 오픈 필드에서도 적을 마주치지만 밀도가 낮으며 필드에서 들어갈수 있는 던전을 통해 밀도가 높고, 보상이 좋은 적들과 상대할수 있음

특히 던전은 매우 길게 구성되고 준비를 제대로 하지 않을경우 난이도가 어렵게 설정되어야 함

어느정도 운과 확률의 요소가 작용하지만 플레이어가 제대로 된 준비와 게임에 대한 경험이 충분해야 클리어가 가능하게끔 설계

던전 내부에서도 필드처럼 실시간 움직임→NPC 조우시 턴제로 전환하는 형태지만 실시간 이동중에도 셀이 나뉘어 있어 한쪽 셀을 클리어하면서 단계적으로 이동하는 방식에 가까움

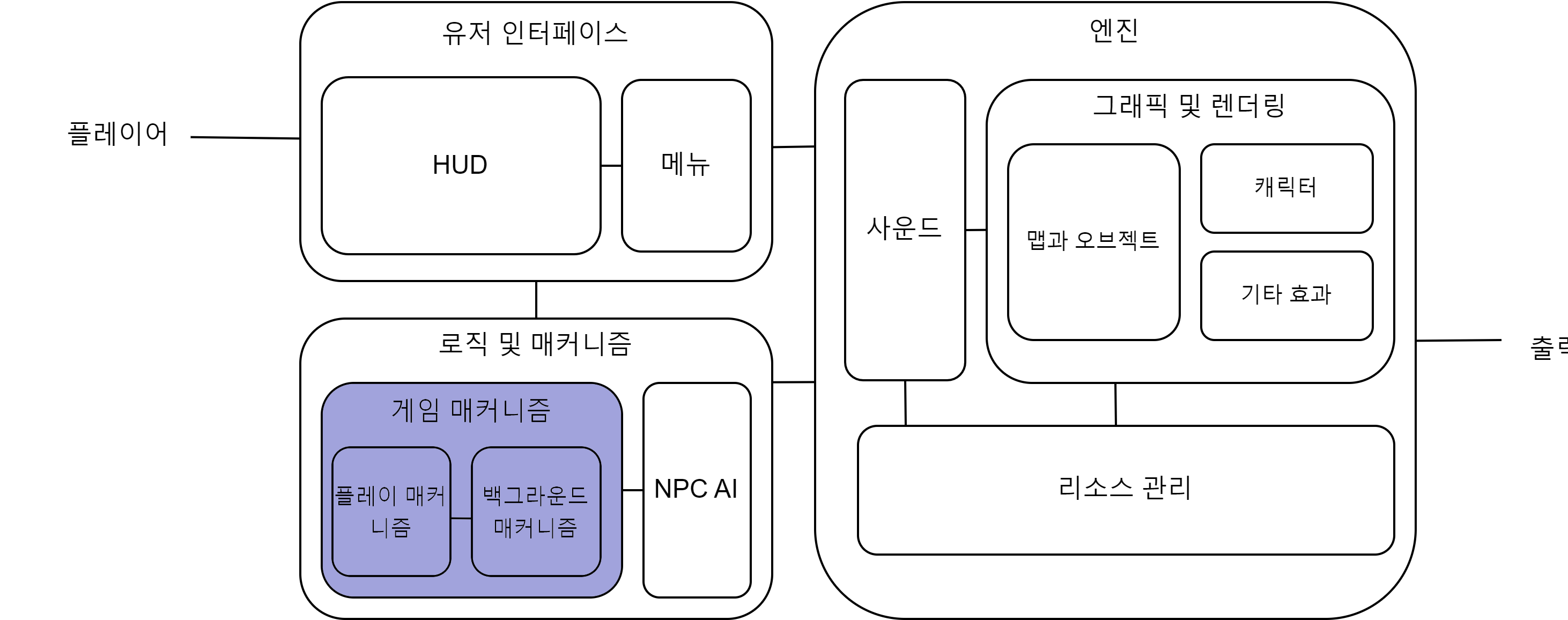
한 던전의 구조와 적들의 구성은 어느정도 일관성이 있게 설계되어야 함

따라서 준비를 철저히 하고 숙련된 플레이어에게 경험적인 이득이 있어야함

또한 던전 진입 전에 어느정도 구조에 대해 힌트를 주는 방식

던전이라고 꼭 적대적인 NPC만 있는것은 아니고 가끔씩 상인 NPC를 등장하게끔 만들어 위험한 상황에서 플레이어에게 도움이 되거나 희귀한 아이템을 비싸게 판매함

2.3 도표



시스템 구조도 (진행상황 - 파란색)



상호작용을 통한 턴제 전환 순서도

2.4 사용 소프트웨어 및 도구, 협업 관리

- 게임 개발

→ 유니티

- 사용 에셋

- 도표 및 그림

→ PhotoScape

→ draw.io

- 협업 관리

github 이용

- 모델링 및 도트 아트

- 사운드

**3.다른 작품들과 비교**

3.1 영향 받은 작품 및 모티브

발더스 게이트 3, 포켓몬스터 시리즈, 폴아웃 클래식

- 오픈 필드이동 → 턴제 전환 시스템 차용

- 폴아웃 시리즈의 S.P.E.C.I.A.L 능력치 모티브

- 지형지물과 오브젝트 활용

엔터 더 건전

- 시점이나 캐릭터 설정, 도트 디자인



출처: Enter The Gungeon, 개발: Dodge Roll, 배포: Devolver Digital, 직접 촬영(2024.04.12.21:28), © Dodge Roll, Devolver Digital, All Rights Reserved.

- 전반적으로 간소화되고 데포르메된 도트 디자인

- 탑뷰에 가까운 시점

- 씬 활용을 통해 유니티에서 2D 도트 그래픽 활용

슬레이 더 스파이어

- 덱 빌딩과 턴제 전투의 혼합

디스코 엘리시움

- 전투이외의 상호작용 모티브

**4.구현 일정 및 과정**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 개발 내용 | | 일정 | | | | | | | | 예상 결과/**실제 결과** |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 게임 개발 기획 | 플레이 메커니즘 |  | |  |  |  |  |  |  | **플레이 메커니즘 확립** |
| 시스템 |  |  | |  |  |  |  |  | **시스템 구현 준비** |
| 게임 시스템 구현 | 플레이 |  |  |  | | |  |  |  | 플레이 메커니즘 구현/ |
| 맵 및 레벨 |  |  |  |  | | | |  | 맵 및 레벨 구현/ |
| 아트 및 사운드 마무리 | |  |  |  |  |  |  | |  | - |
| 최종 데모 테스트 | |  |  |  |  |  |  |  | | 완성도 및 디버깅 필요 측정/ |
| 디버깅 | |  |  |  |  |  |  |  | | - |

4.1게임 시스템 개발 순서

1. 턴제 전환 및 턴제 시스템과 인터페이스

게임의 기본 메커니즘인 실시간-턴제 전환을 만들고 턴제 시스템에 대한 완성과 그에 어울리는 UI개발

2.게임 구조에 맞는 시스템 구현

데이터베이스, 메뉴, 씬 구축

3. 상호작용

NPC및 필드의 오브젝트와 아이템에 대한 상호작용 구현

4. 아이템 설계

아이템을 설계하고 게임에서 실제도 작동되도록 구현

5.맵, 레벨 디자인

전체 맵을 만들고 적절한 난이도 분배와 상승곡선을 통해 레벨디자인 구축

**5.구현 과정 보고서**

20240520~20240602

프로젝트 설정 및 캐릭터 컨트롤러: 기본 움직임 시스템과 카메라 설정

NPC 조우 시스템: NPC와의 충돌 감지 및 턴제 전환 시스템

턴제 전투 시스템: 턴 관리 시스템 구축

20240520~20240522

기초 테스트 공간 제작 – 타일 맵 사용

기본 wasd 움직임 구현

카메라 이동 구현

20240525~20240526

NPC 조우시 레이캐스팅을 활용하여 턴제로 전환됨

턴 관리 시스템 구축 시작

