**Unity Game Engine을 활용한**

**육성 시뮬레이션 게임**

19101262 이은혁

목차

1. 개요 및 목적
   1. 프로젝트 개요
   2. 개발 동기
2. 배경
   1. 시장 조사
   2. 관련 예시
3. 요구명세
   1. User Requirements
   2. System Requirements
4. 설계 및 구현
   1. 기능 요구 명세
   2. 설계
   3. 구현
5. 실행 결과
6. 개선 사항 및 향후 계획
   1. 개선 사항
   2. 향후 계획
7. 참고 문헌
8. 개요 및 목적
   1. 프로젝트 개요

본 프로젝트는 Unity Game Engine으로 만들어진 텍스트 위주로 진행되는 로그라이크 속성의 육성 시뮬레이션 게임이다. 유저에게 게임 내내 수많은 선택지가 던져지며, 선택에 따라 다양한 방식으로 캐릭터는 성장하게 되고 이를 통해 최종 목적인 보스 처치를 향해 나아가게 된다. 유저는 보스전에서 패배하는 등의 정해진 조건에 따라 게임 도중 패배할 수도 있는데, 이 과정을 겪을 때마다 새로운 아이템 해금 등의 성장 요소를 추가하여 각 플레이 경험 자체의 난이도를 조금씩 낮추어 클리어에 용이하게끔 유도한다. 유저가 새로운 것을 발견하는 과정, 주어진 상황을 활용하여 다양한 방식으로 성장하는 과정에서 즐거움을 느낄 수 있도록 하는 것이 본 프로젝트의 의도이다.

* 1. 개발 동기

게임 시장은 과거 어느때보다도 빠르게 성장하고 있다. 이 과정에서 전례 없는 수의 게임이 끊임없이 출시되고 있는데, 개중에는 그래픽, 물리 엔진, 사운드 등의 품질적인 외부 요소가 정말 훌륭하지만 게임 플레이적 재미가 부족한 사례도 존재한다. 이런 사례들은 ‘무엇이 게임을 재미있게 만드는가’라는 질문으로 이어졌으며, 이것의 연장으로 ‘재미있는 게임’을 만들고자 하는 것이 동기가 되었다.

1. 배경
   1. 시장 조사

분석 과정에선 우선 FPS, RTS, 오픈 월드, 타워 디펜스 등의 게임 플레이를 결정하는 장르는 제외하게 되었다. 이들은 각각 게임 플레이 방식이 되어 플레이어의 기호에 따라 게임을 선택하는 요소가 될 뿐, 장르적 인기와 같은 시장의 흐름과 직접적인 연관은 적다고 생각했기 때문이다. 하나의 게임이 성공했을 때 그것과 비슷한 게임이 우후죽순 출시되는 것은 맞지만, 해당 게임들은 대부분 빠르게 잊히거나 관심을 받지 못하는 반면 오히려 별개로 출시된 새로운 방식의 게임들이 다시 한번 성공의 예시가 되는 경우가 잦기에 이 사례가 게임플레이 장르의 인기를 의미한다고는 하기 어렵다.

때문에 게임의 전개를 정의하는 장르에 집중하게 되었고, 이 과정에서 다양한 게임을 찾아본 결과 두가지의 키워드를 도출해낼 수 있었다. ‘로그라이크’란 1980년도 인기 게임 ‘Rogue’에서 파생된 장르로, 해당 게임의 특징인 각 회차의 랜덤성과 선택의 중요성, 영구적 죽음의 요소를 담고 있는 게임을 칭하는 장르이다. 매번 플레이 할 때마다 새로운 경험을 전달하고, 유저의 선택 하나하나가 게임 전개에 큰 영향을 끼치며, 임의의 세이브가 불가능하기에 게임 오버 시 모든 소지품을 잃게 되는 게임들이 해당 장르에 속한다. 그에 반해 ‘소울라이크’란 비교적 최근인 2009년작 ‘Demon’s Souls’에서 파생된 장르로, 해당 게임의 진행 방식과 조작, 넓게는 세계관적 요소까지 담고 있는 게임을 칭하는 장르이다. 주된 특징으로는 플레이어의 높지 않은 체력으로 인해 보스가 아닌 기본적인 적에게도 패배하기 쉽다는 점과, 특유의 묵직한 조작감과 더불어 실패시엔 대가가 따르는 방식으로 함부로 공격을 난사할 수 없다는 점 등 높은 난이도의 전투 시스템을 갖추었다는 점이 있다.

때문에 본 프로젝트에선 ‘로그라이크’ 시스템의 기본 틀을 따라가되 회차가 진행될 때마다 업그레이드가 일부 진행되어 게임의 난이도를 낮춰주는 ‘로그라이트’ 시스템을 차용하기로 했다. ‘로그라이크’ 그대로의 시스템은 유저에게 게임을 진행하며 얻는 성장감각을 주기 어렵고, 이는 시간이 지날수록 큰 피로감을 주기에 일부 코어 게이머들을 제외한 캐주얼 함을 지향하는 유저에게 어필하기 어렵다는 것이 이유이다. 또한 ‘소울라이크’를 차용하지 않은 이유는 게임 플레이에서 나오는 즐거움을 최소화하고 싶었기 때문이다. 개발 동기에서 지적한 것과 같이 퀄리티만 높고 재미있지 않은 게임이 되는 것을 원하지 않았는데, 게임 플레이 경험에서 나오는 즐거움에만 과하게 의존하는 것은 이런 예시와 공유하는 일면이 있다고 생각한다. 이런 이유로 ‘로그라이트’ 시스템이 본 프로젝트의 핵심 시스템 중 하나가 되었다.

* 1. 관련 예시

‘Slay the Spire’의 유물 시스템은 전투에 직접적인 영향을 주며, 공격과 방어에 장비로써 매번 작동하여 전황을 바꾸어 준다. 이런 유물들은 다양한 종류가 존재하여 게임을 진행하며 랜덤하게 수집할 수 있다. 해당 시스템은 본 프로젝트의 장비 혹은 유물 시스템에 차용하여 적용할 예정이다.

‘A Dark Room’의 UI 및 진행 방식은 대부분 텍스트 혹은 ASCII 코드를 이용한 그래픽 요소로 유저에게 게임 플레이 경험을 전달한다. 이런 방식은 시각적인 요소를 최소화하는 방법으로 플레이어의 상상력을 자극하는 소설책과 비슷한 효과를 갖는다. 이는 때론 영화나 동영상보다 뛰어난 흡입력을 갖는 전개 방식으로, 해당 게임에선 텍스트의 효과를 십분 발휘하여 높은 몰입도를 선사하는 경험을 주었다. 이는 본 프로젝트의 ‘최소화된 퀄리티 자극 요소’라는 특징과 비슷하여 게임의 전개 방식에 차용하게 되었다.

‘Princess Maker 2’의 육성 시스템은 턴제 시스템을 일부 차용하여, 유저에게 매달의 시작 마다 캐릭터의 육성 스케줄을 선택할 수 있게 하여 진행 차도에 따라 특정 이벤트의 결과값을 다르게 하는 형식으로 성장이 쉽게 체감되도록 하였다. 본 프로젝트에선 유저에게 매 턴 마다 전투, 육성 등의 선택지를 주어 원하는 방향으로 캐릭터의 성장이 가능하게 하고, 마주하는 이벤트에서 특정 방향으로의 성장을 돕거나 막는 형식의 진행을 통해 해당 시스템을 일부 차용하게 되었다.

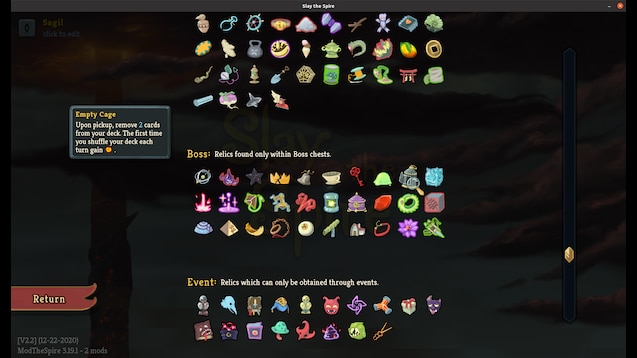


그림 1. ‘Slay the Spire’의 유물 시스템

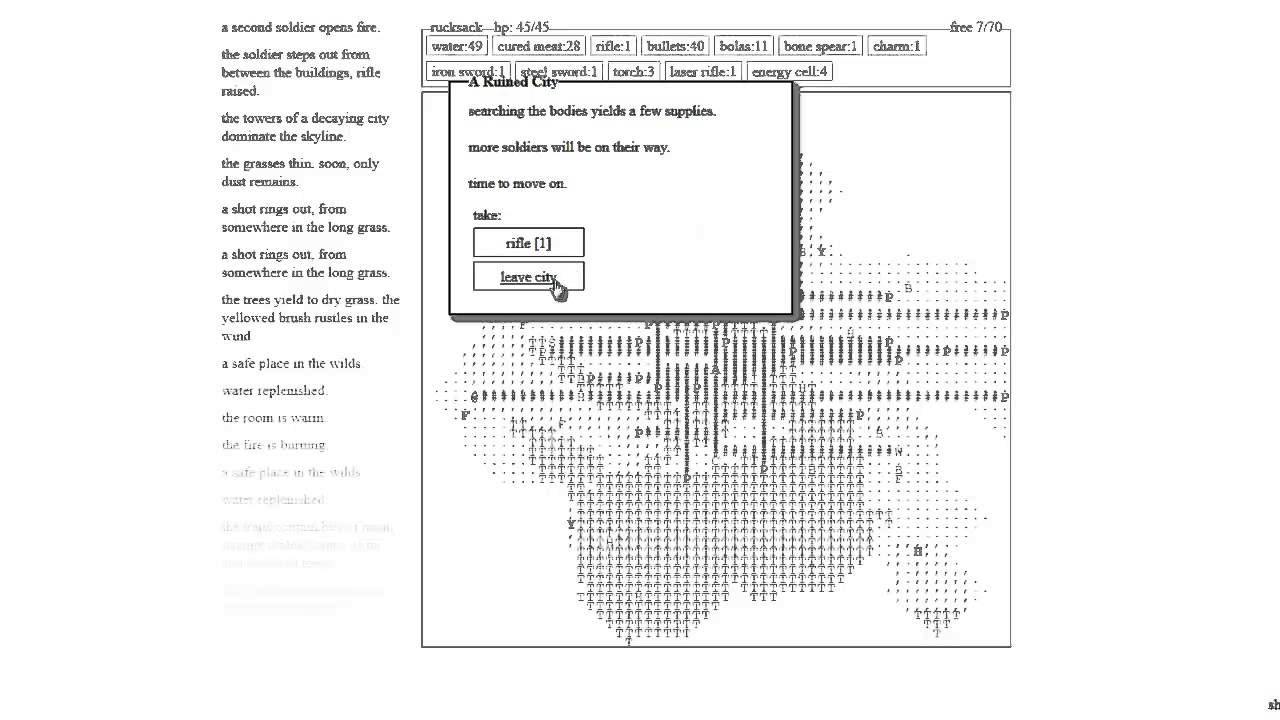


그림 2. ‘A Dark Room’의 UI 및 진행 방식



그림 3. ‘Princess Maker 2’의 진행 구조

1. 요구명세
   1. User Requirements
2. 유저 캐릭터
   1. 유저는 자신의 캐릭터의 특성을 뜻에 맞게 육성할 수 있어야 한다.
   2. 각 특성은 개별적으로 담당하는 효과가 존재해야 한다.
3. 동료 캐릭터
   1. 동료 캐릭터는 아이템과 비슷하게 게임 내의 이벤트를 통해 얻을 수 있어야 한다.
   2. 동료 캐릭터의 육성은 유저 캐릭터와 비슷하게 진행되어 유저에게 무엇을 육성할지 선택할 수 있도록 설계되어야 한다.
   3. 동료는 전투 및 비전투의 과정에서 다양한 방면으로 지원하여 유저가 게임의 클리어를 여러 방면으로 공략할 수 있도록 도움이 되어야 한다.
4. 이벤트
   1. 이벤트의 종류는 시작 시 이벤트, 전투 이벤트, 육성 이벤트의 세가지로, 목표 시점에 맞게 나와야 한다.
   2. 이벤트의 효과는 단순 스탯 증가, 장비 추가, 버프/디버프 추가 등 다양하게 존재해야 한다.
5. 전투
   1. 전투 과정은 텍스트로 이루어지며, 확률에 기반한 운 시스템으로 진행되어야 한다.
   2. 플레이어의 육성 정도 등의 수치에 따라 해당 확률이 변동하여 전투를 승리할 수 있도록 설계되어야 한다.
   3. 전투의 결과에 맞는 보상 혹은 불이익을 가져야 한다.
6. 장비
   1. 장비는 캐릭터의 특성, 전투 시의 효과, 육성 시의 효과 등 다양한 방면으로 유저의 플레이에 영향을 주어야 한다
   2. 장비를 수집하는 방법 또한 여러 이벤트, 전투와 연동하여 다양해야 한다.
7. 엔딩
   1. 결과값에 따라 새로운 장비가 해금되는 방식이 반영되어야 한다.
   2. 게임 종료 시 유저는 자신의 플레이가 갖는 점수 가치를 알 수 있어야 한다.
   3. System Requirements
8. 유저가 만나게 될 이벤트의 랜덤성을 담당하는 시드는 게임 시작과 동시에 저장되어야 한다.
9. 유저의 데이터는 자주 저장되어, 임의적 세이브 로드를 막을 수 있어야 한다.
10. 언어, 볼륨 조절 등 필요 설정을 바꿀 수 있어야 한다.
11. 게임의 모든 과정은 마우스로 진행할 수 있어야 한다.
12. 설계 및 구현
    1. 기능 요구 명세

|  |  |  |
| --- | --- | --- |
| 화면 | 기능 목록 | 세부사항 |
| 시작 화면 | 게임 시작을 비롯한 다양한 외부 행동 선택사항 | New Game, Load Game  Settings  Library(수집한 아이템의 세부사항 확인 가능)  Quit |
| 시작 이벤트 (예언) | 유저의 육성 방향 선택 | 메인, 서브 특성  첫 동료 선택  첫 장비 선택  추가 능력 및 시련 선택 |
| 메인 UI | 육성, 전투, 정보 확인 등 진행 방식 선택 | 육성  전투  동료(동료의 육성)  정보 확인(자신의 특성 및 소지한 아이템 확인) |
| 육성 UI | 다양한 특성을 육성 가능 | 6가지의 특성 중 무엇을 육성할지 선택(동료의 육성도 틀은 같음) |
| 전투 UI | 전투 과정 및 결과 출력, 필요시 선택 요소 삽입 | 각 공격의 표기  공격에 의한 확률 변동 표기  장비에 따른 선택지 등 필요시 삽입  승패 결과 출력 |
| 이벤트 | 이벤트 내용에 관한 설명 텍스트 및 선택지 | 이벤트의 설명  승낙, 거절 등의 이벤트별 적합한 선택지 |
| 게임 종료 화면 | 게임의 결과 출력 | 유저의 기록에 따른 최종 점수  새로운 아이템의 해금 |
| 설정 | 게임의 외부 요소를 유저의 기호에 맞게 설정 | 언어, 볼륨 조절 등의 외부적 요소 설정 |

* 1. 설계

도표, 텍스트, 그림, 스케치이(가) 표시된 사진

자동 생성된 설명

그림 4. 프로젝트 게임 플로우

게임이 시작할 때 유저는 ‘예언’ 단계를 거쳐 플레이 방식의 초기 방향을 정하게 되고, 메인 UI에서 전투 혹은 육성을 매 턴 마다 결정하며 원하는 방식으로 캐릭터를 성장시키게 된다. 해당 턴이 10턴 반복된 이후 보스전을 거치게 되며, 보스전에서 패배하는 등의 패배 조건을 만족하게 되거나 혹은 최종 보스까지 처치하는 승리 조건을 만족하게 되면 결과 발표의 과정을 거친 후 게임이 종료된다. 유저는 해당 회차의 결과로 해금된 새로운 장비등의 이점을 가지고 다시 게임을 진행할 수 있다.

텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 직사각형, 도표이(가) 표시된 사진

자동 생성된 설명

그림 4, 5. 메인 화면과 전투 화면의 예상 UI

메인 화면에서 유저는 해당 턴을 어떻게 소모할지 결정하게 된다. 예측 가능하고 비교적 적을 수 있는 이익을 주는 육성과 불확실하지만 많은 이익을 얻을 수 있는 전투에서 선택할 수 있으며, 각 결정은 한 개의 턴을 소모하게 된다. 전투 혹은 육성이 끝나면, 다음 턴으로 넘어가기 전 이벤트가 발생하는데, 이는 각각 유저에게 이익 혹은 불이익을 주며 아이템, 스탯 변화 등의 다양한 효과로 이를 실행한다.

전투의 경우 유저의 장비와 스탯에 맞게 공격이 진행되고, 적 또한 장비와 스탯에 맞게 공격이 진행된다. 각 공격은 최종 승리 확률에 영향을 주며, 양측 모두 가능한 공격이 전부 끝난 경우의 최종 승리 확률에 따라 승패를 결정하게 된다. 육성 측면의 즐거움을 강조하고 극대화하기 위해 전투 과정에서 유저의 선택 등의 능동성이 들어가는 요소는 특정 장비나 특성으로 한정하여 최소화할 것이다.

게임이 종료되면 자신의 최종 점수를 표시하는 화면이 나와 유저에게 해당 회차의 수적 가치를 보여줄 것이다. 이는 최종 보스를 클리어하는 것을 유일한 목적으로 삼지 않고 클리어 이후에도 유저가 높은 점수를 목표하여 게임을 진행하도록 하는 자극제의 역할을 한다. 또한 유저가 자신의 플레이에서 보완 사항과 개선점을 알아갈 수 있는 지표의 역할도 할 예정이다.

* 1. 구현

폰트, 로고, 그래픽, 상징이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 폰트, 스크린샷, 로고이(가) 표시된 사진

자동 생성된 설명

그림 6, 7, 8. Unity 프로젝트의 구조

프로젝트 내에서 앞서 설명한 요구 사항의 화면들을 개별 Scene으로 제작하는 과정이 우선시 되었다. 이후 해당 Scene의 내부에 각 버튼과 이미지 등을 삽입하며 해당 object를 관리하는 방식으로 구조가 짜여졌다.

Object는 각 Scene별로 필요한 버튼과 이미지 외에도 ButtonController, EventHandler등 해당 Scene에서 발생할 수 있는 행동을 관리하는 Object도 존재한다. 이들은 서로 상호작용하며 각 행위에 필요한 코드를 요청하게 된다. 예를 들어 StartEventButtonController의 선택 버튼이 눌렸을 때 call 되는 OnOptionSelection 함수는 StartEventEventHandler의 특정 액션을 실행하는 OnOptionSelected 함수를 호출하게 되는 방식. 이를 통해 프로젝트의 각 시스템을 담당하는 코드를 분할하여 관리하고 있다.

1. 실행 결과

스크린샷, 도표, 직사각형, 디자인이(가) 표시된 사진

자동 생성된 설명스크린샷, 원, 직사각형, 도표이(가) 표시된 사진

자동 생성된 설명

그림 9, 10. MainMenu Scene

* MainMenu에는 New Game, Load Game, Settings, Quit등의 선택지가 있다.
* New Game등 파일에 변화를 주는 선택지는 실행 되기 전 팝업 창을 띄워 해당 사항이 유저의 결정이 맞는지를 묻는다.

스케치, 인간의 얼굴, 예술, 사진 액자이(가) 표시된 사진

자동 생성된 설명인간의 얼굴, 스케치, 스크린샷, 만화 영화이(가) 표시된 사진

자동 생성된 설명

그림 11, 12. StartEvent Scene

* StartEvent등의 이벤트는 이벤트의 목록에서 dialogue와 option을 추출하여 표시한다.
* 해당 화면에는 데모에서 이해를 돕기 위한 AI 생성 이미지가 삽입되었고, 개발 과정에선 이를 직접 제작한 Sprite로 수정할 예정이다.
* 첫 시작시엔 Option들이 나오지 않고, NPC의 Dialogue가 끝난 이후 가능한 선택지가 등장한다.
* Dialogue의 진행은 플레이의 단순화를 위해 버튼을 제외한 화면의 어느 곳을 눌러도 가능하게끔 개발되었다.

인간의 얼굴, 스케치, 스크린샷, 만화 영화이(가) 표시된 사진

자동 생성된 설명

그림 13. Alternative StartEvent Case

* StartEvent 말고도 다른 이벤트의 대부분이 필요한 기능으로 다양한 가능 이벤트 중 랜덤하게 번호를 뽑아 이벤트를 출력한다.

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

그림 14. userData의 저장

* 게임에서 자주 일어날 userData의 저장은 대부분의 게임과 같게 Application.persistentDataPath에 일어난다.
* 이는 타 플랫폼 간의 구조의 차이에 의해 일어나는 저장 데이터 위치의 persistency 문제등을 해결해주는 고정적 저장 경로이다.

1. 개선 사항 및 향후 계획
   1. 개선 사항

* 아직 Scene 대부분이 기능적으로 구현되지 않은 상태이다. 만들어진 기초 시스템을 바탕으로, 각 Scene에 적용시켜 정상적인 플레이가 가능하게 하는 것이 가장 우선시 될 개선 사항이다.
* 게임 시스템의 기획은 거의 완료되었지만, 컨텐츠의 기획이 부족하다. 다양한 이벤트, 적, 장비의 예시들이 기획되어야 하며, 해당 아이템들을 프로젝트 내에 삽입하는 과정이 필요하다.
* 프로젝트에 활용되는 Sprite와 Animation의 제작이 필요하다. 그래픽적 퀄리티를 최소화 하는 것과는 별개로, 좋은 프로젝트가 되기 위해선 게임 내 그래픽의 컨셉이 일관적이고 연계되어 유저에게 만족감을 선사할 수 있어야 한다. 이를 위해선 다양한 아이템들의 Sprite와 UI의 Animation등의 시각적 요소의 제작이 필요하다.
  1. 향후 계획

|  |  |  |
| --- | --- | --- |
| 6월 | 전체 기틀 | * 타 Scene들의 개발 * 정상 작동 하도록 설계, 플레이 가능하도록 제작 |
| 7월 | 추가적 요소 | * 다양한 Event, Item의 추가 * 각 요소들의 작동 설계 |
| 2학기 | Sprite, Animation | * 작동하는 게임의 수정 단계 * 게임의 전체적인 퀄리티 향상 |

1. 참고 문헌

[1] ‘Steam의 Slay the Spire’, https://store.steampowered.com/app/646570/Slay\_the\_Spire/?l=koreana

[2] ’Unity Learn’, https://learn.unity.com/