

2D 포트폴리오

Binding of Isaac 모작

경일게임아카데미

이승준

1. 게임 소개
2. 기본 기능 구현
3. 핵심 기능 구현

1. 게임 소개



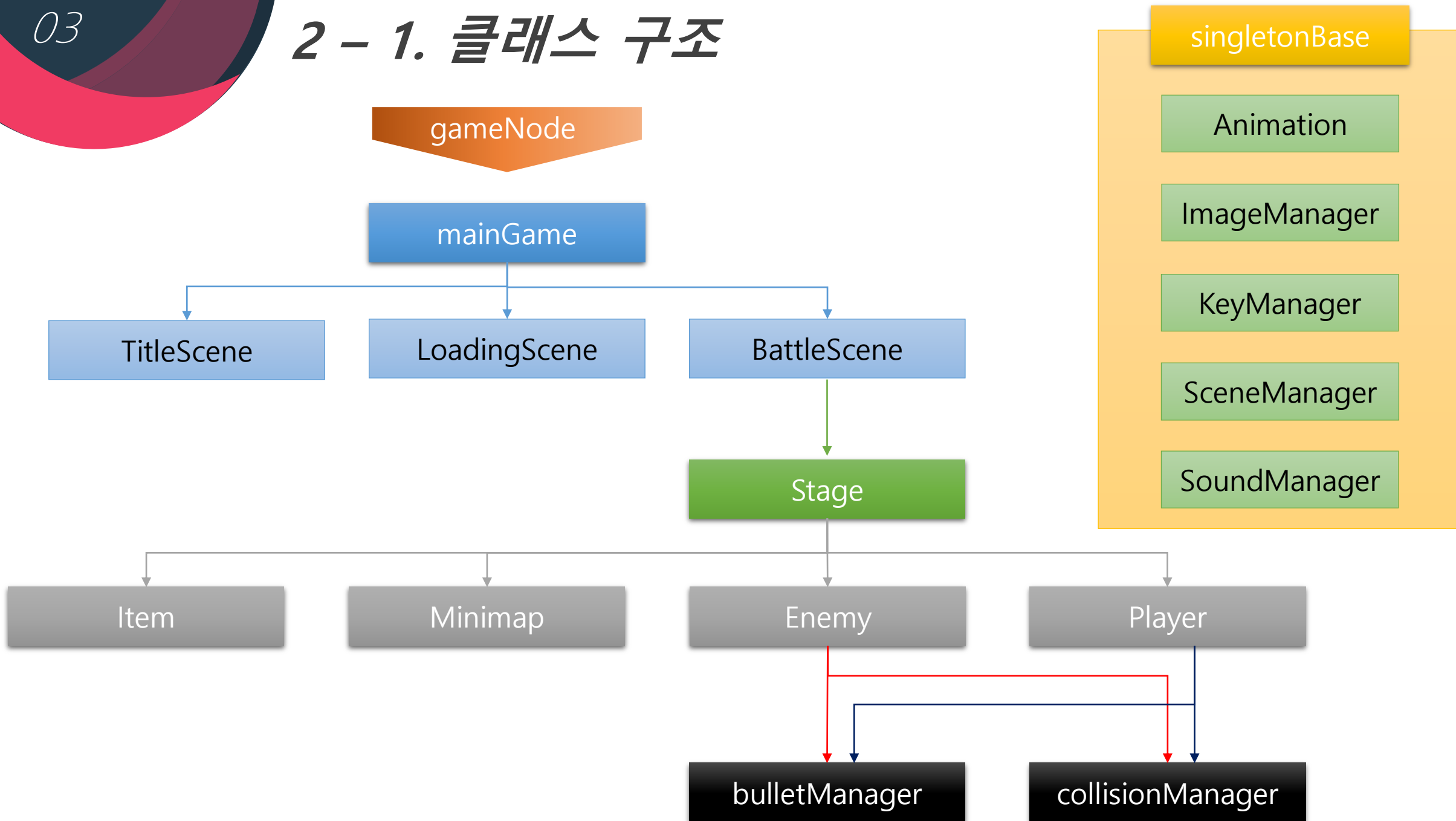
1. 게임이름 : 아이작의 구속
2. 게임 장르 : 어드벤처, 로그라이크
3. 개발 환경 : WINAPI/C++
4. 개발 기간 : 19.11.25 ~ 19.12.20일 (4주)
5. 영상 소개 :

<https://youtu.be/ZnTHRfhSRxM>

1 - 1. 개발 일정

월	화	수	목	금	토	일
1일	2일	3일	4일	5일	6일	7일
클래스 구조설계 및 역할정리	플레이어 및 적 리소스작업	맵툴 및 스테이지 관련 리소스 작업	플레이어 이동과 공격 구현	플레이어 공격력, HP 등 스텟구현	미니맵 시스템 구 현	디버깅 및 1주차 마무리 작업
8일	9일	10일	11일	12일	13일	14일
맵 랜덤 배치 알고리즘 제작 및 미니맵 시스템 구현			맵툴 구현 및 타일 오브젝트 구현		몬스터 오브젝트 와 아이템 오브젝 트 구현	디버깅 및 2주차 마무리 작업
15일	16일	17일	18일	19일	20일	21일
플레이어와 타일 오브젝트 충돌처 리	추가 리소스 작업	일반 몬스터 스텟 구현 및 공격패턴 구현		몬스터 충돌처리	폭탄 구현 및 타일오브젝트 파 괴 구현	디버깅 및 3주차 마무리 작업
22일	23일	24일	25일	26일	27일	28일
보스몹 패턴 및 보스 스테이지 구 현	픽업 아이템 구현	컬렉티블 아이템 구현	상점 구현, 샘플맵 제작	샘플맵 세이브 로 드 기능 구현	디버깅 및 빌드 테스트	

2 - 1. 클래스 구조



2 - 2. 적과 총알

< 적과 총알 객체 생성 >



1. 자료 구조 (Vector)

: vector를 이용해 각각의 속성을 담은 bullet과 enemy 형식으로 객체를 미리 생성해 저장했다.

2. 충돌

: 충돌은 렉트 충돌 방식을 사용하여 간단하게 구현하였다.

3. 메모리 관리

: vector를 이용해 객체들을 미리 생성했기 때문에 사용되지 않는 객체들도 렉트 충돌 등을 검사하는 메모리 누수를 고려했다.

리스트를 사용해 사용되고 있는 객체들끼리만 비교하도록 해서 메모리 효율을 높였다



< 맵틀을 이용한 오브젝트관리 >

게임 내 오브젝트를 표시하기 위해 타일맵 방식을 사용했다. 오브젝트의 종류는 1. 아이템, 2. 적, 3. 장애물 이다.

그림처럼 인게임에서는 보이지 않는 선으로 이루어진 렉트 타일들을 구성했다. 그리고 맵틀을 사용해 오브젝트들을 겹치기 않게 적절하게 배치하였다.

<이미지의 애니메이션 함수>



게임에는 이미지의 변화가 없는 정적인 오브젝트와 이미지를 변화시켜 움직임 등의 시각효과를 주는 동적인 오브젝트가 있다. 움직이는 오브젝트의 경우 연속된 이미지를 매 프레임마다 출력하는 방식으로 표현되지만 이미지의 개수가 다 다르고 이미지의 연결 순서가 아닌 불규칙한 애니메이션을 가진 오브젝트도 존재한다.

모든 경우의 수마다 다른 변수를 사용하고 함수를 사용해 코드가 길어지고 난잡해 지는 것을 피하기 위해 애니메이션을 처리하는 다양한 함수를 만들고 클래스를 따로 관리하여 간결하고 직관적이고 효율적으로 이미지 렌더링을 했다.

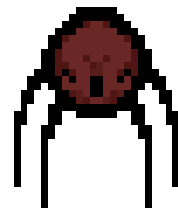
2-4. 적 패턴 구현

<공격파리>



: 플레이어를 추적한다.

<롱다리>



- : 1. 플레이어를 추적한다.
- 2. 십자방향으로 총알 발사
- 3. 휴식

<박쥐>



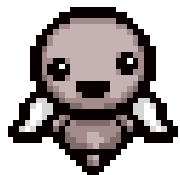
: 플레이어를 추적하고 일정거리 안에 있으면 이동속도가 증가한다.

<애벌레>



- : 1. 십자방향으로 일직선 이동
- 2. 벽을 만날시 90도로 방향전환
- 3. 진행 방향에 적 발견시 전방으로 돌진

<아기천사>



- : 1. 플레이어를 느린 속도로 추적
- 2. 적 방향으로 총알 3발 발사
- 3. 적 주변 위치로 순간이동

<빨간 폭탄 파리>



- : 1. 대각선 이동을 하고 벽에 부딪히면 반사되어 날아감
- 2. 죽을시 360도로 총알 6발 발사

2-4. 적 패턴 구현

<Maw>

<파리>



: 적을 매우 느리게 추격하고 적 방향으로 총알을 한발 발사

<boss : 몬스트로>



1. 공통 패턴

: 적 방향으로 총알 수십발을 흩뿌린다.

2. 상황별 패턴

- 플레이어가 범위 안에 있는 경우

: 적 방향으로 점프를 하고 착지시 총알을 사방으로 퍼트린다.

- 범위 밖인 경우

: 점프로 화면 밖으로 벗어난 후 플레이어 머리 위로 착지, 총알을 사방으로 퍼트린다.

2 - 5. 아이템 구현

<픽업 아이템>



열쇠 : 잠긴문을 여는데 사용



폭탄 : 설치시 몇 초후에 폭발하여 데미지를 주거나 오브젝트를 파괴한다.



동전 : 상점에서 아이템 구입시 사용

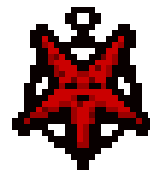


하트 : 체력 회복



소울 하트 : 체력 대신 소모되는 쉴드

<컬렉티블 아이템>



: 획득시 플레이어의 체력이나 공격력, 공격속도 등을 올려주는 아이템들. 총알의 외형이나 플레이어의 외형을 바꾸는 경우도 있다.

3 - 1. 랜덤 맵 알고리즘



아이작 게임의 메인 콘텐츠는 로그라이크 특유의 랜덤 맵 생성 시스템이다. 원작에는 다음과 같은 규칙이 존재한다.

1. 보물방, 보스방, 상점방은 반드시 하나씩 생성된다.
2. 보물방, 보스방, 상점방과 인접한 방은 1개이다.
3. 모든 방에 접근할 수 있어야 한다.
4. 보스방은 시작위치에서 일정거리 밖에 생성된다.

그래서 위의 규칙을 적용해 알고리즘을 만들었다.

1. 보물방, 보스방, 상점방을 생성한다.
2. 생성된 특수방과 인접한 방은 1개씩 생성한다.
3. 시작 지점으로 방을 연결한다.
4. 규칙들을 검사해 맞지 않으면 재생성한다.

3 - 2. 맵파일 입출력



랜덤한 위치에 맵을 생성하더라도 해당 방에 오브젝트가 어떤 식으로 배치되어 있는지가 게임 난이도와 재미에 영향을 준다. 따라서 코드를 이용한 완전 랜덤 방식이 아닌 원작 게임처럼 맵틀을 이용해 샘플 맵을 미리 제작해 놓고 로드하는 방식을 택했다.

1. 수십개의 샘플 저장
2. 맵생성 후 방노드별로 샘플에서 복사해서 저장
3. 인게임에서 현재 방노드의 맵파일 로드
4. 방 이동시 현재 노드의 파일을 저장하고 이동하는 방 노드의 맵파일 로드

다음과 같은 방식으로 난이도를 조절하고 필요한 정보만 출력하여 메모리 낭비를 최소화 할 수 있게 했다.



감사합니다.