

로그라이크 서바이벌 게임

- 최초 Proposal -



소프트웨어학과 201720721 양성호
디지털미디어학과 201821122 이현섭
디지털미디어학과 202021126 김태유

1

로그라이크(Rogue-like)

1. 최초의 던전 탐색 RPG 'Rogue'라는 게임의 특징과 시스템을 따르는 게임들
2. 플레이어 임의로 이전 시점으로 로드 불가능, 한 번 죽으면 진행도를 모두 잃음
3. 순간순간의 판단과 선택을 통해 게임 진행
4. 반복되는 실패 및 재시작 속에서 플레이어의 경험이 쌓이며 멀리 진행해 나가는 방식



2

뱀파이어 서바이버즈 로그라이크 서바이벌 장르

1. 무한히 생성되어 플레이어를 향해 달려드는 적들
2. 스킬 탄막 등을 이용해 적들을 처치
3. 시간 흐름에 따라 강해지고, 불어나는 적들
4. 아이템 획득, 경험치 성장 등을 통해 스킬 획득 및 강화
5. 제한 시간 동안, 혹은 최대한 오래 생존
6. 실패 시, 다시 처음부터 시작



3

개발 스택 및 목표



개발 스택

Android Studio
Java
OpenGL ES



1차 목표

로그라이크 서바이벌 장르 게임의
필수적 기능을 구현한
프로토타입 결과물 제작

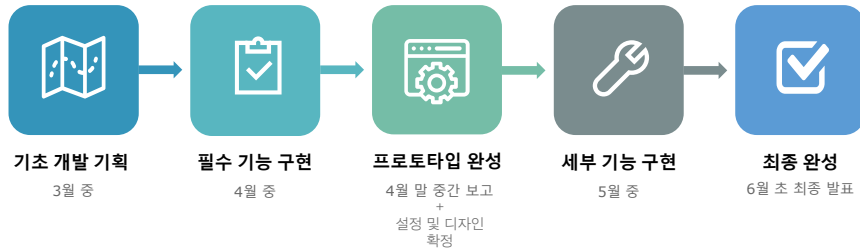


최종 목표

팀 만의 고유한 특색을 담은
로그라이크 서바이벌 게임 완성

4

개발 계획 타임라인



5

로그라이크 서바이벌 게임

- 중간 보고 -



소프트웨어학과 양성호
디지털미디어학과 이현섭
디지털미디어학과 김태유

6

Review...

개발 스택 및 목표



개발 스택

Android Studio
Java
OpenGL ES



1차 목표

로그라이크 서바이벌 장르 게임의
필수적 기능을 구현한
프로토타입 결과물 제작



최종 목표

팀 만의 고유한 특색을 담은
로그라이크 서바이벌 게임 완성

7

Review...

개발 계획 타임라인



기초 개발 기획

3월 중



필수 기능 구현

4월 중



프로토타입 완성

4월 말 중간 보고
+
설정 및 디자인
확정



세부 기능 구현

5월 중



최종 완성

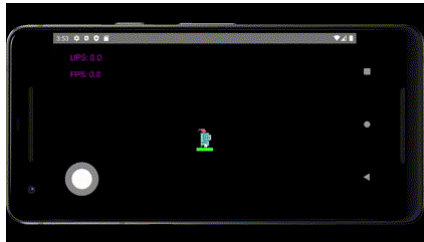
6월 초 최종 발표

8

프로토타입 구현 내용

캐릭터, 적 오브젝트 및 시스템 기초 구현

- 서피스뷰(SurfaceView)를 통한 게임 화면 구현
- 개발 중 성능 확인을 위한 UPS/FPS 표시
- 주인공 캐릭터의 움직임을 위한 조이스틱
- 카메라 시점 플레이어가 중앙에 오도록 추적 및 고정
- 적 오브젝트의 생성 및 주인공 캐릭터 추적
- 캐릭터-적 충돌 시 캐릭터의 HP 감소
- 임시 공격 수단 : 화면 클릭 방향으로 투사체 발사

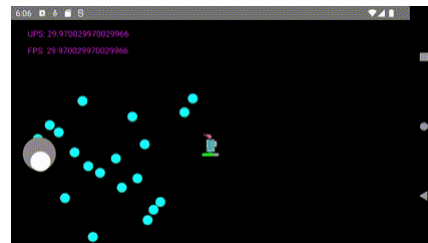
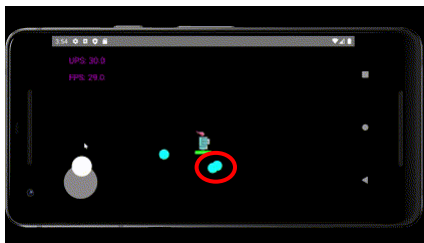


9

프로토타입 구현 내용

적 오브젝트 간 겹침 현상 제거

- Problem : 플레이어가 원형으로 회전 시 적 오브젝트가 하나로 겹쳐짐
- Problem : 겹쳐진 오브젝트들과 충돌 시 즉시
- Solution : Flock Behavior / Local Avoidance
- 프레임 업데이트 마다 각 적 오브젝트 이동 계산
- 이동 계산 과정에 타 오브젝트들의 위치 또한 고려하도록 수정
- Still Problem : 이동 속도 감소, 연산량 증가

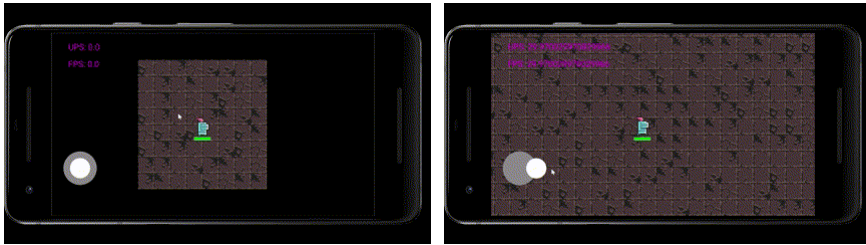


10

프로토타입 구현 내용

게임 배경 맵 무제한 반복

- Problem : 플레이어가 어느 방향으로 이동하든, 맵에는 **끝이 없어야** 함
- Solution :
 - InfiniteBackground 클래스에서 플레이어를 영점에 두고
각 사분면에 타일 bitmap이 랜덤으로 배치된 background 생성
 - update() 메소드에서 플레이어가 background의 특정 위치에 도달하면 background 이동



11

프로토타입 구현 내용

적 오브젝트 세부 구축

- 적 스프라이트 및 종류 추가 : 오크 / 진흙괴물 / 임프
- 적 체력 추가 : 일부 적 오브젝트는 투사체 다수 히트 시 사망하도록 수정
- 적 이동 애니메이션 구현 : 일정 프레임마다 스프라이트 교체
- 적 **피격 시 시인성**을 위한 피격 애니메이션 추가 : 직접 에셋 수정하여 적용

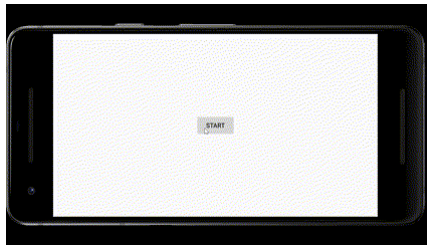


12

프로토타입 구현 내용

UI 기초 구현

- 제한 시간 표시 기능 : 화면 좌상단
- 현재 : 플레이 시간 표시 / 추후 변경 가능
- 게임 로비 화면 : 게임 시작 전 직업 선택란 표시
- 사망 시 Game Over 화면 → 메인 로비로 복귀

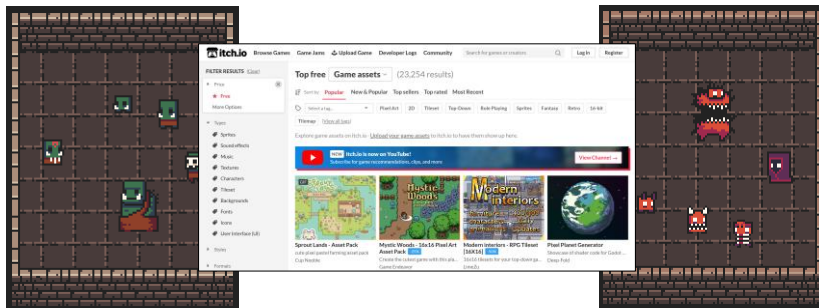


13

설정 및 디자인

게임 테마

- Assets : itch.io - Game Assets / Free
- Main Asset : [Dungeon Tileset II](https://0x72.itch.io/dungeontileset-ii) (<https://0x72.itch.io/dungeontileset-ii>) by 0x72(Robert)
- 다운로드 받은 파일을 게임 해상도에 맞게 크기 수정 후 적용
- 몬스터 웨이브 속에서 제한시간 동안(동일이 틀 때까지) 생존



14

세부 구현

최종 완성 빌드까지의 목표

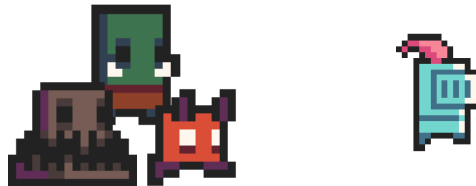
- 캐릭터 스킬 테마에 맞춰 구현
- 레벨 업 플레이어 선택지 설계
- 적 스폰 관련 레벨 설계
- (임시) 난이도 구현 : 노말, 하드, 나이트메어...
- 추가 아트 에셋 찾기
- BGM / SFX 추가
- 내부 저장소를 활용한 플레이 기록 Save & Load
- 게임 결과에 따른 성취도(업적) 보상 : 편의성, 캐릭터 성장 등...
- 최종 빌드 시연 : Git Release 혹은 Google Play Store 등록

15

로그라이크 서바이벌 게임

Dungeon Survival

- 최종 발표 -



소프트웨어학과 양성호
디지털미디어학과 이현섭
디지털미디어학과 김태유

16

지난 구현 내용

지난 중간 발표 내용 리뷰

Implemented...

- 게임 기초 시스템 구현
- 게임 배경 맵 반복을 통한 무제한 경계 구현
- 초기 레벨 디자인과 게임 테마에 부합하는 적 오브젝트 구축
- 기초적 UI 구현 및 테마 적용

Still problem & Target...

- Enemy 오브젝트 간 Avoidance 개선
- 레벨 디자인 구체화 및 테마 적용
- 기초 수준의 UI
- etc...

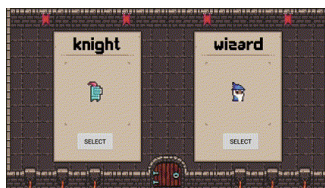


17

추가 구현 내용

배경 애니메이션 구현

- 다운로드 받은 오픈 에셋을 포토샵으로 가공하여 활용
- DrawableAnimation 기능을 통해 움직이는 배경 이미지 구현



```
<?xml version="1.0" encoding="utf-8"?>
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot="false">

    <item
        android:drawable="@drawable/title_1"
        android:duration="150"/>

    <item
        android:drawable="@drawable/title_2"
        android:duration="150"/>

    <item
        android:drawable="@drawable/title_3"
        android:duration="150"/>

</animation-list>
```

18

추가 구현 내용

스프라이트 애니메이션 구현

- 게임 곳곳에 존재하는 다양한 오브젝트들에게 테마에 맞는 **스프라이트 연결**
- 모든 투사체는 캐릭터 애니메이션 방향과 일치하도록 구현
- 기사 기본 공격 및 스킬 구현
 - 다중 공격 판정, 공격 애니메이션과 피격판정을 따로 구현
 - 스킬 사용 시 지속시간 10초 간 **무적**, 쿨타임 30초
- 마법사 기본 공격 및 스킬 구현
 - 단일 공격 판정, 기존에 만든 spell 클래스를 활용 파이어볼 애니메이션 구현
 - 스킬 사용 시 현재까지 소환된 적들을 10초 간 **얼림**, 쿨타임 30초



19

추가 구현 내용

게임 클리어 판정 시스템 구현

- 제한시간 동안 플레이어가 몬스터 웨이브에서 살아 남으면 **게임 클리어**
- 클리어 시점부터 몬스터 생성 중단
- 플레이어 방향으로 계속 접근하다가 클리어 시 반대방향으로 도망치는 애니메이션 구현
- 클리어 시점 플레이어 기록은 **Intent**를 통해 결과화면 Activity로 전달



```
Intent intent = new Intent(getContext(), GameClear.class);
intent.putExtra(name: "playtime_minute", GAME_CLEAR_TIME_MINUTE);
intent.putExtra(name: "playtime_second", value 0);
intent.putExtra(name: "play_killCount", getMonsterKillCount());
intent.putExtra(name: "play_levelCount", player.getLevel());

getContext().startActivity(intent);
((Activity) getContext()).finish();
```



```
Intent intent = getIntent();
playTime_minute_round = intent.getIntExtra(name: "playtime_minute", defaultValue 0);
playTime_second_round = intent.getIntExtra(name: "playtime_second", defaultValue 0);
killCount_round = intent.getIntExtra(name: "play_killCount", defaultValue 0);
playerLevel = intent.getIntExtra(name: "play_levelCount", defaultValue 0);
```

20

추가 구현 내용

플레이어 성장 시스템 구현 (1)

- 플레이 중 몬스터 처치 시 **경험치 획득** 및 획득 내용이 반영될 화면 상단 **경험치 바** 클래스 구현
- 정해진 수준의 경험치 획득에 따른 **플레이어 레벨** 및 **레벨 업** 클래스 구현
- 레벨 업 이벤트 발생 시 타이머 및 적 오브젝트 접근 등 **레벨 업 UI 제외 게임 전체 정지**
- 레벨 업 선택지: 공격력 증가 / 속도 증가 / 체력 증가 / 체력 회복 / 추가 스킬 ...
- **랜덤**으로 **3가지 선택지**를 제공, 배치 또한 **랜덤**
- 레벨 업 팝업은 직접 좌표 계산하여 이미지 배치 및 클릭 식별하는 방식으로 구현



21

추가 구현 내용

플레이어 성장 시스템 구현 (2)

- 플레이 중 몬스터 처치 시 **경험치 획득** 및 획득 내용이 반영될 화면 상단 **경험치 바** 클래스 구현
- 정해진 수준의 경험치 획득에 따른 **플레이어 레벨** 및 **레벨 업** 클래스 구현
- 레벨 업 이벤트 발생 시 타이머 및 적 오브젝트 접근 등 **레벨 업 UI 제외 게임 전체 정지**
- 레벨 업 선택지: 공격력 증가 / 속도 증가 / 체력 증가 / 체력 회복 / 추가 스킬 ...
- **랜덤**으로 **3가지 선택지**를 제공, 배치 또한 **랜덤**
- 레벨 업 팝업은 직접 좌표 계산하여 이미지 배치 및 클릭 식별하는 방식으로 구현



etc...

22

추가 구현 내용

게임플레이 기록 통계 기능 추가

- 액티비티 간 [Intent 전달](#)을 통해 게임플레이 기록 전달
- [SharedPreferences 저장소](#) 기능을 통해 게임플레이 기록 유지
- 저장소 기록과 비교하여 플레이어 신기록 수립 여부 확인
- 메인화면에서 전체 플레이 기록 확인 가능

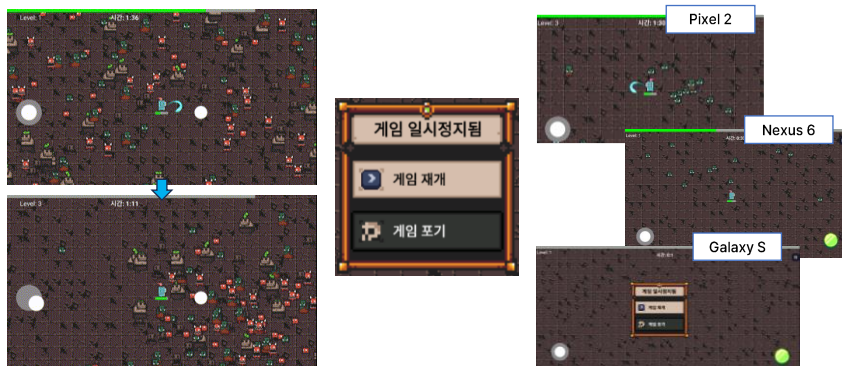


23

추가 구현 내용

기타 수정 내용

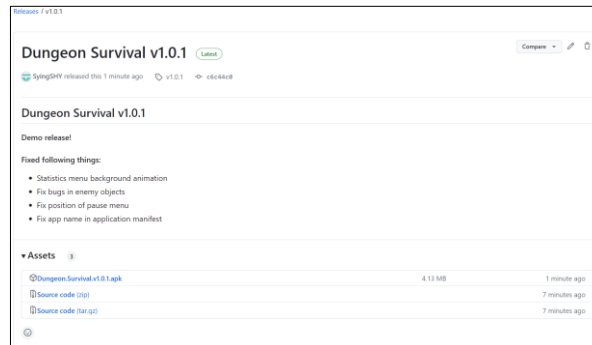
- 적 캐릭터 추가 구현: 고블린 / 초르트 / 오우거 / 빅 좀비
- [Avoidance 알고리즘 개선](#)
- 인게임 일시 정지 버튼 구현
- [DisplayMetrics](#)을 통해 가져온 기기 정보로 다양한 화면비 지원
- etc...



24

최종 구현 내용

Github Release



<https://github.com/SyngSHY/2301-MP-RS/releases>

< Github v1.0.0 기준 67 Commits, 16 Branches, 49 Java Codes, 8510 Lines >

25

최종 구현 내용

게임 시스템 플로우



26

최종 구현 내용

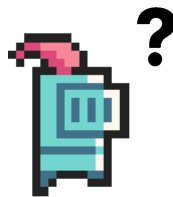
게임 레벨 디자인



27



시연 및 Q&A



28