

# 소프트웨어 프로젝트

## 프로젝트

- 명칭 : Dungeon Story
- 개발 인원 : 4(개발 2명(저포함), 디자인 2명)
- 참여기간: 2017.05. ~ 2018.12.
- 개발 환경: Java(1.8) / Android Studio / AndEngine

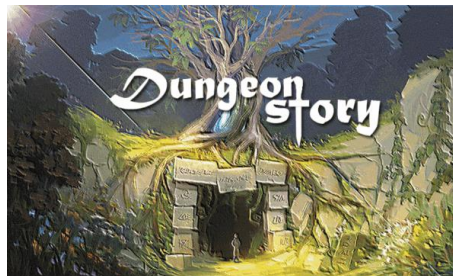
## 소개

Dungeon Story는 횡스크롤 모바일게임과 로그라이크에서 아이디어를 얻어 만든 게임입니다. 한번 게임 캐릭터가 죽으면 이어하는게 아니라 처음부터 하는 로그라이크라는 방식에다 휴식시에나 던전 진입시 각종 이벤트를 확률적으로 발생하게 하여서 여러가지 재미요소를 추가했습니다.

### **DungeonStory 시연 동영상 링크**

<https://youtu.be/0HToa5o5SYQ>

타격감을 위해 효과음과 배경화면도 따로 넣었습니다.



[사진자료 1] 차례로 진입 화면, 메인화면 게임화면

게임을 좋아하는 학생들이 모여서 여러가지 요소들을 넣어서 만든 게임입니다.

DungeonStory 개발에는 AndEngine이라는 라이브러리를 사용했습니다. AndEngine은 메뉴터치를 통한 객체이동 센서를 통한 이동, 가속도 물리엔진, 배경스크롤, 맵배경에서의 스프라이트 이동과 충돌 등 2D게임 개발에 필요한 대부분의 기능을 갖춘 라이브러리입니다.

## 소스 코드 일부분

### Stage.class

아래의 소스 코드는 DungeonStory에서 일반 공격시 불러오는 함수 입니다. 플레이어가 바라보는 방향이나 공속 업그레이드 상태등 여러가지 요소들의 고려해서 만들었습니다.

```
public void showSword(AnimatedSprite _playerSprite, int _playerDir, int number) {
    /**
     * 프레임 100 당 0.1 초
     * 기본공속 100 기준
     * 기본공격 프레임 총 800 / 0.8 초
     * 공속 2당 공격속도 1% , 프레임 8 차감(각 2씩 차감)
     */
    int sword_location = 30, index_sword_x, index_sword_y;
    swordCount++;
    if(_playerDir == 1) {
        index_sword_x = 0; index_sword_y = 3;
        player_sprite.animate(new long[] {sword_speed, sword_speed, sword_speed, sword_speed},
26, 29, false);
    }else{
        index_sword_x = 4; index_sword_y = 7;
        player_sprite.animate(new long[] {sword_speed, sword_speed, sword_speed, sword_speed},
37, 40, false);
    }

    AnimatedSprite mSwordSprite = new AnimatedSprite(_playerSprite.getX() + (_playerDir *
sword_location), _playerSprite.getY(),
        mSwordTextureRegion, this.getVertexBufferObjectManager());
    mSwordSprite.animate(new long[] {sword_speed, sword_speed, sword_speed, sword_speed},
index_sword_x, index_sword_y, false);
    mSwordSprite.setScale(1.5f);
    mSwordSprite.setUserData("sword"+number);
    mEntityList.add(mSwordSprite);
    mScene.attachChild(mSwordSprite);

    mSound_sword2.play();
    mSound_sword2.play();
    //간격마다 타격이 필요한 경우 아래 코드를 핸들러를 이용한다.
    if(findShape("sword"+number) != null){
        int i = 0; //엔티티의 번지수를 체크
        for (IEntity pEntity : mEntityList) {
            if (pEntity.getUserData() != null &&
(pEntity.getUserData().toString().contains("enemy") ||
                pEntity.getUserData().toString().contains("middle") ||
                pEntity.getUserData().toString().contains("high"))) {
                AnimatedSprite bodySprite = (AnimatedSprite) mEntityList.get(i);
                if (mSwordSprite.collidesWith(bodySprite)) {
                    fix3_name = pEntity.getUserData().toString();
                    int targetNumber = battleEnemy1.indexOf(new battle_enemy(fix3_name));
                    if (targetNumber >= 0) {
                        Debug.d(fix3_name + " Sword Hit!~~ ");
                        mSound_sword1.play();
                        int damage = damageCalculation.getDamageCalculation(false,
                            player_info.myLevel, player_info.myATK,
```

```

        player_info.myF_Atk, player_info.myHit,
        battleEnemy1.get(targetNumber).lv,
        battleEnemy1.get(targetNumber).def,
        battleEnemy1.get(targetNumber).p_def);

        float tmp_xDistance = 4f;
        float levelbalance = battleEnemy1.get(targetNumber).lv -
player_info.myLeve/;
        if(levelbalance >=5 &&
(pEntity.getUserData().toString().contains("enemy") ||
        pEntity.getUserData().toString().contains("middle"))){
            tmp_xDistance = 3f - ( (levelbalance-4) * 0.3f);
        }
        else if(levelbalance >=3 &&
pEntity.getUserData().toString().contains("high")){
            tmp_xDistance = 2f - ( (levelbalance-2) * 0.1f);
        }
        damageCollision(fix3_name, 1, 1, damage, 0f, tmp_xDistance, -0.4f);
    }
}
}
i++;
}
}

destroySword("sword"+number);
}

```