**보고서**

**컴퓨터 종합설계 001**

**Team. GameinhAi (한예원, 박지훈, 윤병철)**

목차

**요약**

* 프로젝트 개요
* 주요 결과
* 결론

**서론**

* 프로젝트 배경 및 목적
* 목표 및 기대 효과
* 범위 및 한계

**프로젝트 수행 계획 및 방법론**

* 프로젝트 일정 및 단계
* 수행 방법론
* 사용된 도구 및 기술

**프로젝트 수행 내용**

* 작업 내용 및 세부 내역
* 각 단계별 주요 활동
* 참여자 역할 및 책임

**결과 및 성과**

* 프로젝트 산출물
* 주요 성과 및 결과물 설명

**문제점 및 해결방안**

* 프로젝트 진행 중 발생한 문제
* 문제 해결 방법 및 과정
* 교훈 및 개선점

**결론**

* 프로젝트 결과 요약
* 프로젝트의 의의 및 활용 방안
* 향후 과제 및 제안

**부록**

* 참고자료
* 추가 산출물

**요약**

**프로젝트 개요**

"**For the Granada**"는 플레이어가 악당에게 납치된 여주인공을 구하는 2D 싱글 플레이 액션 어드벤처 게임이다. 다양한 스테이지를 탐험하고 퍼즐과 보스를 공략하며 긴장감 넘치는 플레이를 제공한다.

**주요 결과**

- Unity 엔진을 활용하여 2D 탑뷰 스타일 게임 완성

- BFS/DFS 알고리즘 기반 NPC 및 맵 생성 로직 구현

- LLM API를 이용한 미니게임 퀴즈 구현

- 보스 AI 및 미니게임 시스템 성공적 개발

**결론**

본 프로젝트를 완성하면서 Unity와 알고리즘, AI 기술을 결합한 게임 개발 경험을 축적했다.

**서론**

**프로젝트 배경 및 목적**

게임 개발 전반적인 과정을 팀 단위로 직접 경험하며 게임 개발 경험을 쌓고, 실질적인 프로젝트 협업 경험을 확보하고자 한다.

**목표 및 기대 효과**

- Unity 기반 2D 게임 제작 능력 함양

- 알고리즘 및 AI 기반 게임 기능 구현

- 팀 프로젝트를 통한 협업 능력 배양

**범위 및 한계**

- 플랫폼: PC

- AI 적용 제한: NPC 탐색 및 미니게임 생성, 보스 행동 패턴  
 - 그래픽 품질과 게임 난이도는 제한된 자원 내에서 구현

**프로젝트 수행 계획 및 방법론**

**프로젝트 일정 및 단계**

기본 설계 → 기능 개발 → 테스트 및 디버깅 → 최종 발표

**수행 방법론**

- Agile 방식을 적용하여 주간 스프린트로 개발 진행

- 매주 정해진 요일에 팀 회의를 통해 진행 상황 점검 및 계획 수립

**사용된 도구 및 기술**

- 언어 : C#

- 프레임워크 : Unity

- 라이브러리 : UnityEngine API, Nuget

- 엔진 : Unity 2D

- 버전 관리 : Git, Github

- UI : Unity UI

- NPC의 플레이어 추격 : DFS 알고리즘

- 랜덤 스테이지 생성 : BFS 알고리즘

- 미니게임 퀴즈 : LLM API(Google Gemini 1.5 Flash Model)

**프로젝트 수행 내용**

**작업 내용 및 세부 내역**

1. UI/UX

- 유니티 에디터 UI 컴포넌트 및 디자인 에셋 활용

- Leonardo AI 및 그림판 3D로 제작한 아이콘 사용

- Unity 이벤트 시스템으로 버튼 상호작용 구현

2. 음향

- Unity AudioSource로 배경음 및 효과음 구현

- UI와 연동된 음량 조절 기능 추가

3. 미니게임 퀴즈

UnityWebRequest 메소드를 사용하여 네트워크 연결을 구현했다.

이미지는 미리 Leonardo Ai를 통해 생성된 이미지 100장을 리소스에 저장하였으며, 그 중 랜덤으로 중복되지 않게 3개를 선택하여 사용했다.

Gemini 모델의 API를 이용하기 위해 전송할 데이터를 JSON 형식으로 작성해 Header, Body를 포함해 POST 방식으로 전송하였다.

프롬프트는 String 데이터로 저장했고, 이미지는 Byte형 배열에 담아 Base64 인코딩 후 인라인 데이터 형식으로 전송했다.

이후 모델의 응답을 newtonsoft.json 패키지를 통해 파싱해서 퀴즈의 정답으로 사용했다.

나머지 보기는 미리 저장된 String 배열에서 랜덤으로 중복되지 않는 3개를 가져오도록 했다.

4. NPC의 플레이어 추격

DFS 알고리즘으로 플레이어를 추격하는 기능을 구현했다.

NPC는 정해진 경로를 순찰하다가 시야범위 내에 플레이어가 들어오면 추격을 시작하도록 구현했다. NPC의 위치를 시작 노드로, 플레이어의 위치를 목표 노드로 설정하여 최적 경로를 찾도록 설정했다. 0.1초 간격으로 시작 노드와 목표 노드를 재설정하여 재탐색하도록 했다.

5. 스테이지 랜덤 생성

스테이지 진입시 시작방과 끝방을 랜덤으로 생성헸다.(시작방의 경우 모서리의 생성되지 않도록 범위 조정)

BFS 알고리즘을 통해 시작방과 끝방사이에 최단거리를 구하고 백트레이싱을 통해 경로를 저장했다.

경로에 저장된 방 중 하나를 무작위로 골라 무작위 방향 중 하나의 방이 생성가능 하면 생성했다.(정해진 방의 개수가 충족될 때까지 반복)

생성된 방들은 미리 생성된 방들의 프리팹 중 하나로 랜덤하게 저장했다.

방과 접해있는 방향의 문 활성화했다.

6. 보스 AI

유니티 애니메이터와 C# 스크립트를 이용해 애니메이션 흐름을 제어했다.

코루틴을 이용해 일정 주기마다 보스가 랜덤 패턴을 실행하도록 구현했다(플레이어 추격 돌진 / 플레이어 추격 점프).

HP가 ½ 이하가 되면 강화 패턴이 발동되도록 구현했다(맵 상에 불 오브젝트 생성).

**각 단계별 주요 활동**

- 설계: 상세설계서 및 요구사항 명세서 작성

- 개발: NPC AI, 보스 로직, 맵 랜덤 생성, 미니게임 시스템 구현

- 테스트: QA 및 디버깅 수행, 테스트 플레이

**참여자 역할 및 책임**

- 한예원(팀장): 시스템 설계 및 플레이어 기능 구현, NPC AI 구현, 맵 이동, 상자 상호 작용 구현

- 박지훈(팀원): UI/UX 디자인 및 아이템 구현, 미니게임 로직 개발, 보스 AI 구현, 음향 구현

- 윤병철(팀원): 게임 시스템 전반 구현, 맵 디자인 및 랜덤 맵 생성 로직 개발, 아이템 구조 구현

**결과 및 성과**

**프로젝트 산출물**

- 플레이 가능한 2D 탑뷰 어드벤처 게임

- 소스 코드 및 게임 매뉴얼 제공

**주요 성과 및 결과물 설명**

- NPC가 상황마다 최단 거리를 탐색해 플레이어를 추격

- 매 스테이지 진입 시 맵을 랜덤으로 조합하여 새로운 스테이지 생성

- 상자 오브젝트와 상호작용 시 LLM의 응답을 바탕으로 퀴즈 생성

- 응답 오류나 네트워크 오류 시 미리 준비된 퀴즈 세팅

- 오브젝트 충돌 시 HP가 감소하며, 랜덤 패턴 및 강화 패턴을 구사하는 보스 NPC 구현

**문제점 및 해결방안**

**프로젝트 진행 중 발생한 문제**

- NPC AI의 경로 탐색 오류

- 맵 이동 중 끼임 문제

- 미니게임의 난이도 밸런싱

- 과부하 문제

- 깃허브 머지 중 충돌 문제

- 게임 플레이 난이도 조절 필요

**문제 해결 방법 및 과정**

- 문제가 되는 코드 수정

- 각종 알고리즘 수정 및 최적화

- 난이도 관련 테스트 및 조정

**교훈 및 개선점**

- 협업 도구 활용법 및 고려 사항 경험

- 초반 기획 단계에서 더 정교한 설계 필요

- 코드 품질 관리를 위한 정기적 코드 리뷰 필요

**결론**

**프로젝트 결과 요약**

Unity 엔진과 알고리즘을 활용한 2D 어드벤처 게임 개발을 성공적으로 완료했다.

**프로젝트의 의의 및 활용 방안**

- 게임 엔진 활용 기술 및 AI API 사용 기술 강화

- 게임 설계 능력 강화

- 깃허브 사용을 통한 협업능력 강화

**향후 과제 및 제안**

- 멀티플레이 모드 추가

- 도전과제 추가

- 다회차 플레이를 위한 요소 추가

- 모바일 플랫폼 확장

**부록**

**참고자료**

- Unity 공식 문서

- 관련 블로그 게시글

**추가 산출물**

- 상세 설계서

- 요구사항 명세서

- 주간업무 보고 및 계획서

- 결과 보고서