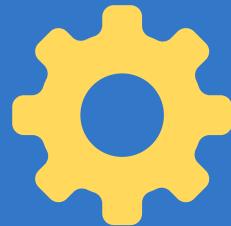




한국정보교육원

아수라(ASURA)



TEAM

세얼간이

김기찬, 조유빈, 이영호

[멘토] 김승기

목 차

-
- 01 프로젝트 개요
 - 02 프로젝트 팀 구성 및 역할
 - 03 프로젝트 수행 절차 및 방법
 - 04 프로젝트 수행 경과
 - 05 자체 평가 의견

프로젝트 개요

1

프로젝트 주제 및 선정 배경, 기획의도

간편하고 빠르게 즐길 수 있는 웹 기반 PvP 게임 수요 증가

누구나 쉽게 접근 가능한
빠른 템포 전투
다양한 맵과 무기
설치 없이 실행 가능한 웹
플랫폼

2

프로젝트 내용

실시간 멀티플레이 전투
(2~8인)

무기 쟁탈 시스템, 점프맵,
레이드 등 다양한 모드
짧고 빠른 라운드 중심의
PvP + 전략성

3

활용 장비 및 재료

개발 툴/플랫폼:
Visual Studio Code
GitHub, Render (배포)

사용 기술 스택:
HTML, CSS, JavaScript,
Node.js, Express.js
Three.js, Socket.io

4

프로젝트 구조

별도 설치 없이 웹에서 즉시
실행 가능한 PvP 게임 제공

다양한 모드·무기·캐릭터 추
가를 통한 지속적인 콘텐츠
확장 가능

실시간 멀티플레이, 웹 3D
그래픽, 서버 통신 등 실무
적용 기술 반영

5

활용방안 및 기대 효과

캐주얼 PvP 게임 플랫폼으로
확장 가능
커뮤니티 기반 배틀 콘텐츠
적용

접근성 높은 웹 PvP 콘텐츠
로 사용자 유입 기대
개인 성장 시스템, 커스터마
이징 도입 시 수익화 가능

프로젝트 팀 구성 및 역할

훈련생	역할	담당 업무		
이영호	팀장	아이템 시스템 구현	전투 시스템 구현	
김기찬	팀원	서버 구축	코드 안정화	캐릭터 모션 시스템 구축
조유빈	팀원	HP 시스템	맵 로직 및 렌더링 구현	
김승기	멘토	UI 설계 및 구현	맵 로직 및 렌더링 구현	
		주제 선정 피드백, 프로젝트 질의응답		

03

K-Digital Training

프로젝트 수행 절차 및 방법

구분	기간	활동	비고
사전 기획	7/18(금) ~ 7/18(금)	▶ 프로젝트 기획 및 주제 선정 ▶ 기획안 작성	아이디어 선정
기반 환경 구축	7/21(월) ~ 7/22(화)	▶ 기본 환경 구성 ▶ 게임 에셋 정리 ▶ 핵심 아키텍처 설계	
게임 데이터 설계	7/22(화) ~ 7/23(수)	▶ 데이터 관리 유틸리티 구현	
핵심 기능 구현	7/23(수) ~ 7/24(목)	▶ 핵심 게임 로직 구현 ▶ 실시간 서버 기본 로직 구현	팀별 중간보고 실시
시스템 통합 및 안정화	7/24(목) ~ 7/28(월)	▶ 클라이언트-서버 기능 연동 ▶ UI 구현 및 통합 테스트	최적화, 오류 수정
총 개발기간	8/11(금) ~ 8/22(월)(총 10일)		

04 프로젝트 수행 경과

▶ 아수라(ASURA) 요구사항 분석 및 기획

기획 배경 및 문제 정의

설치 없이 웹에서 빠르게 즐길 수 있는 PvP 게임 수요 증가에 따라, 기존 웹 기반 게임의 실시간성 및 콘텐츠 부족 문제를 해결하고자 했습니다.

기획 목표

설치 없는 웹 기반 PvP 게임 제공
빠른 템포의 멀티플레이 전투 구현
직관적 조작 및 쉬운 학습 곡선 설계
향후 확장 가능한 모듈식 구조 확보

핵심 요구사항

기능 : 실시간 전투 시스템, 무기/피격 처리, 플레이어 동기화
비기능 : 웹 브라우저 호환성, 네트워크 딜레이 최소화, 확장성 및 재배포 용이성

프로젝트 수행 경과

기술 스택 선정 및 구조 설계

- **HTML, CSS, JavaScript:** 웹 인터페이스 구현
- **Node.js:** 플레이어 간 매칭, 상태 관리, 데이터 처리
- **Three.js:** 3D 그래픽 렌더링
- **Express.js:** 서버 구축 및 게임 매칭, 상태 관리
- **Socket.IO:** 실시간 양방향 통신으로 플레이어 동기화
- **Render:** 클라우드 배포로 웹 브라우저에서 즉시 접속 가능

핵심 기능 구현

- 26종의 캐릭터 에셋
- 다채로운 무기 선택
- 맵 구현 (도시, 무인도, 당구대, 우주)
- 게임 모드 개발 (대전, 팀전, 점프맵, 레이드, 1인칭)

핵심 기능 구현

- 실시간 플레이어 동기화
- 게임 루프 및 렌더링 파이프라인
- 투사체(Projectile) 구현
- 프레임 단위 모션 및 충돌 처리
- 데이터 기반 콘텐츠 관리

프로젝트 수행 경과

테스트 및 개선 반복

- 실시간 위치 동기화와 함께 맵과 캐릭터 간 충돌 로직을 더욱 정교하게 수정해 플레이어가 벽이나 장애물에 자연스럽게 막히도록 개선.
- 무기 관련 로직을 재검토하여 공격 범위, 데미지 판정, 공격 발동 타이밍을 세밀하게 조정, 공격 시 애니메이션과 투사체가 정확히 맞물리도록 동기화 문제를 해결
- 킬과 데스 정보가 서버에 실시간으로 연동되어 게임 통계에 즉시 반영되도록 구현

04

K-Digital Training

프로젝트 수행 경과

시연 동영상



프로젝트 결과물에 대한 완성도 평가 - 8점

프로젝트의 핵심 기능 구현과 기술 적용은 성공적이었으나, 코드의 유지보수성 및 코드 모듈화 부족으로 팀 협업 과정에서 경험의 부족함을 느낌

추후 개선점이나 보완할 점

React/Next.js로 마이그레이션: 컴포넌트 기반 구조와 효율적인 상태 관리를 도입 -> 코드의 재사용성과 유지보수성을 높일 것
Three.js 이외에도 Babylon.js, Ammo.js 등과 같은 라이브러리를 이용하여 물리 연산 및 충돌의 개선

잘한 부분과 아쉬운 점

실시간 멀티플레이, 웹 3D 그래픽(Three.js), 서버 통신(Socket.IO) 등 실제 서비스에 필요한 핵심 기술들을 적용, 정기적인 소통을 통한 명확한 역할 분담 -> 높은 생산성 유지

상태 관리가 복잡해지면서 코드 구조가 비대해지고, 디버깅이 어려워졌다. React 같은 프레임워크의 필요성을 느낌

팀원들끼리 각자 개발한 코드를 병합하는 과정의 어려움을 느낌

느낀 점이나 경험한 성과

상태 관리의 중요성을 깨달았고, 왜 현대적인 프론트엔드 프레임워크가 필요한지 체감.
기획부터 개발, 배포까지 전 과정을 경험해 보며 실제 서비스 개발 프로세스에 대한 실무 감각을 익힐 수 있었음
팀원 간의 역할 분담과 협업 과정을 통해, 원활한 커뮤니케이션의 중요성을 느낌

세**얼간이**