종합 설계 B1 TEAM AAA

Code-Eat

컴퓨팅 사고력 향상을 위한 멀티플레이 게임

김민정, 김시완, 김하연, 문다인, 홍주완

목차

1. 서론

- 1.1 제안 배경
- 1.2 과제 개요

2. 본론

- 2.1 시스템 구성
- 2.2 사용 사례
- 2.3 상세 설계
- 2.4 적용 기법 및 기술
- 2.5 핵심 연구/ 개발 과제
- 2.6 업무 분담/ 일정 계획

3. 결론

- 3.1 달성 목표
- 3.2 프로젝트 의의

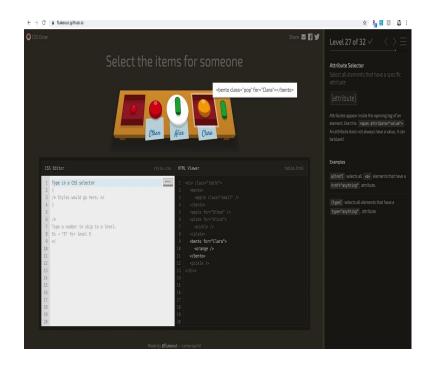


1.1 제안배경 1.2 과제 개요

제안 배경

- 1. 코딩 교육 관심 상승
 - 코딩 관련 교육 기관, 프로그램 등장 중
- 2. 컴퓨팅 사고력 향상을 도와주는 게임 제작 결정

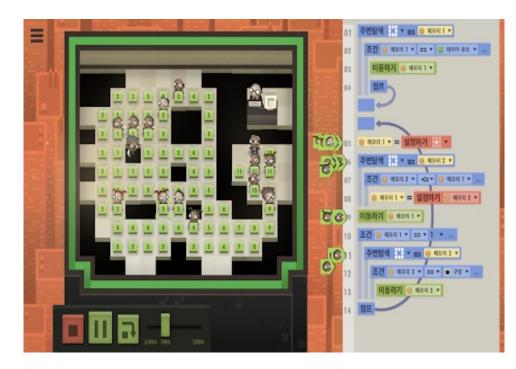
디지털 인재 양성 주요 내용 자료: 교육부		
	내용	적용 시기
초·중학교 디지털 교육 강화	코딩 교육 의무화 정보 수업 확대 ■ 초등학교 17시간→34시간 ■ 중학교 34시간→68시간	2025년
디지털 영재 육성	초중고 SW 영재학급 확대	2022년 40개 →2025년 70개
대학 학사운영 개선	학·석·박사 통합과정 (5년 6개월·11학기) 신설 졸업 전 집중 교육 프로그램 (부트캠프) 신설	2023년
	첨단분야 정원 확대 요건 완화	이르면 2024년부터



CSS Diner



Scratch



70억 명의 인간들

※ 기존 코딩 게임의 특징

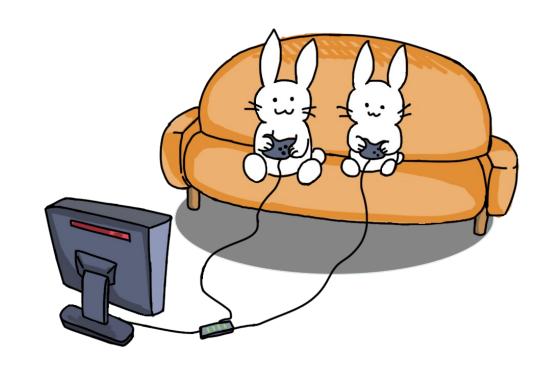
- 1. 정적인 요소(퀴즈, 퍼즐, 블록 등)만으로 진행
- 2. 1인 플레이 게임이 대다수

※ 목적

컴퓨팅 사고력 학습에 중점 전투 시스템과 멀티플레이 기능을 추가하여 재미를 더하는 것

※ 대상

프로그래밍을 처음 접하는 전 연령층





말랑말랑 두뇌학원

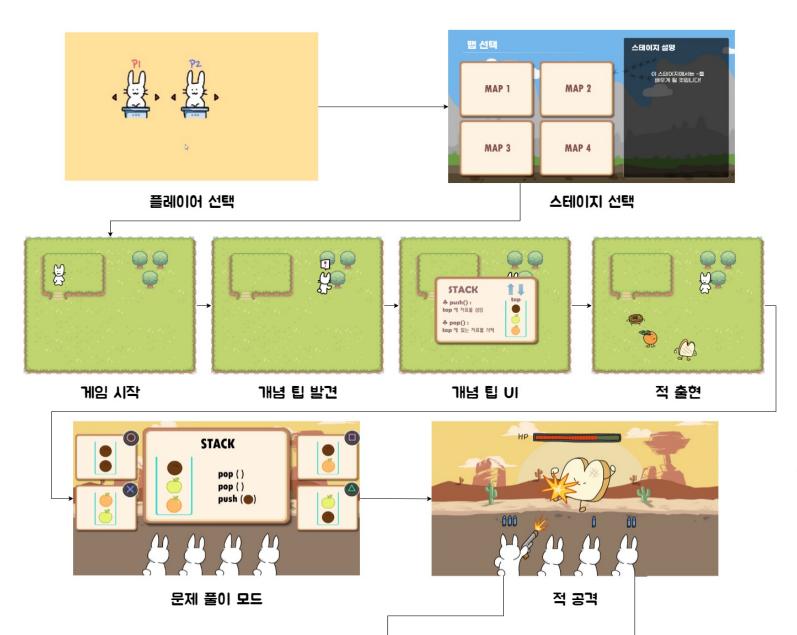
※ 참고한 기능성 게임의 특징

- 1. 오답을 선택했을 시 해답을 제공하지 않음
- 2. 개인 커서가 타 플레이어에게 보임



- 2.1 시스템 구성
- 2.2 사용 사례
- 2.3 상세 설계
- 2.4 적용 기법 및 기술
- 2.5 핵심 연구/ 개발 과제
- 2.6 업무 분담/ 일정 계획

시스템 구성도

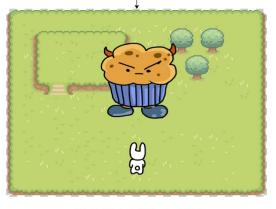


1. 개별 플레이어 선택 2. 스테이지 선택

3. 맵 탐색, 개념 팁 발견 4. 모든 개념 팁 학습 시 적 출현 5. 적 선택, 문제 풀이 모드 입장

6. 문제 풀이
7. 정답 -> 선택 순 공격 기회 차등 부여
오답 -> 체력 감소
8. 공격
9. 적 타격 데미지 순 점수 차등 부여
10. 적 모두 처치할 때까지 반복

10

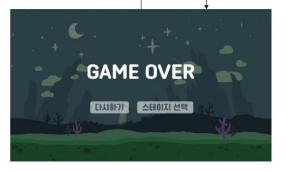


최종 보스 등장 & 처치



P10I 무승자입니다!
300pt 200pt 100pt 50pt

MVP & 최종 점수 발표

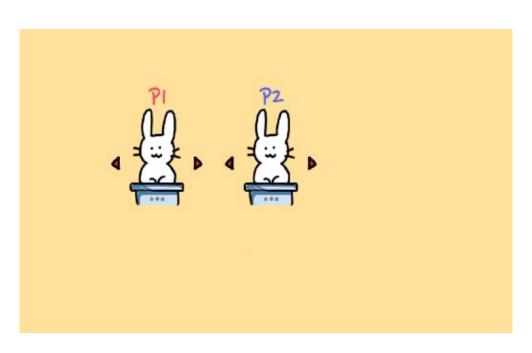


게임 오버

11. 모든 일반 적 처치 후 최종 보스 등장

12. 최종 보스 처치 완료 OR 모든 플레이어 사망 시 문제 해설 공개

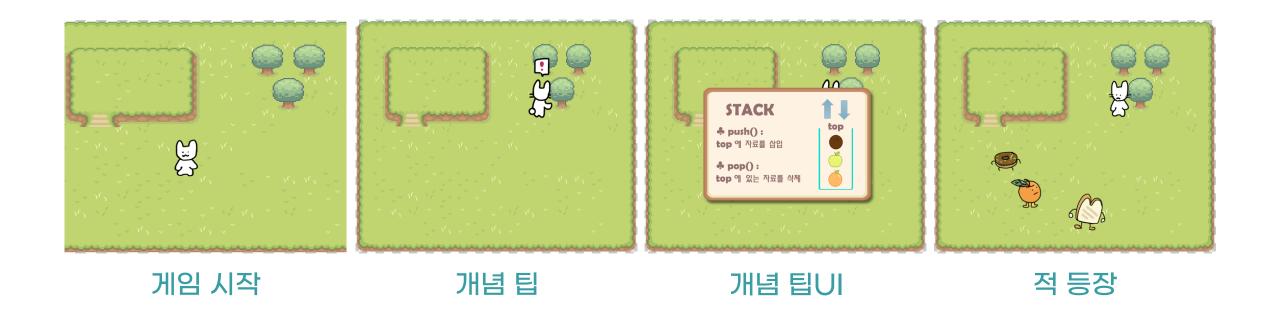
> 12. MVP 및 최종 점수 발표 13. 모든 플레이어 사망 시 게임 오버

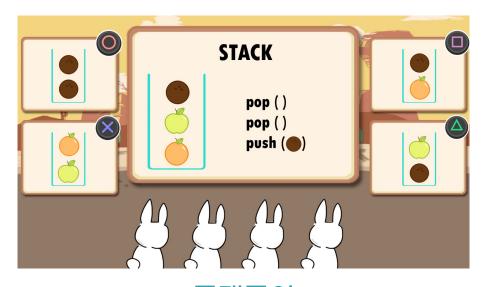


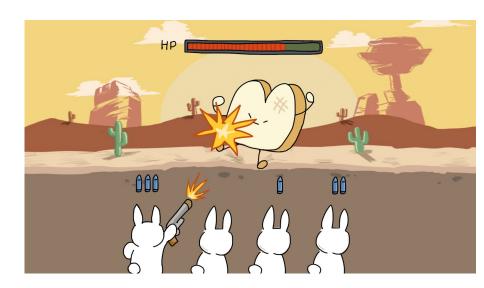
플레이어 선택



스테이지 선택

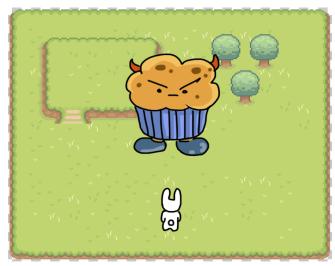






적 공격

문제풀이



최종 보스





게임오버

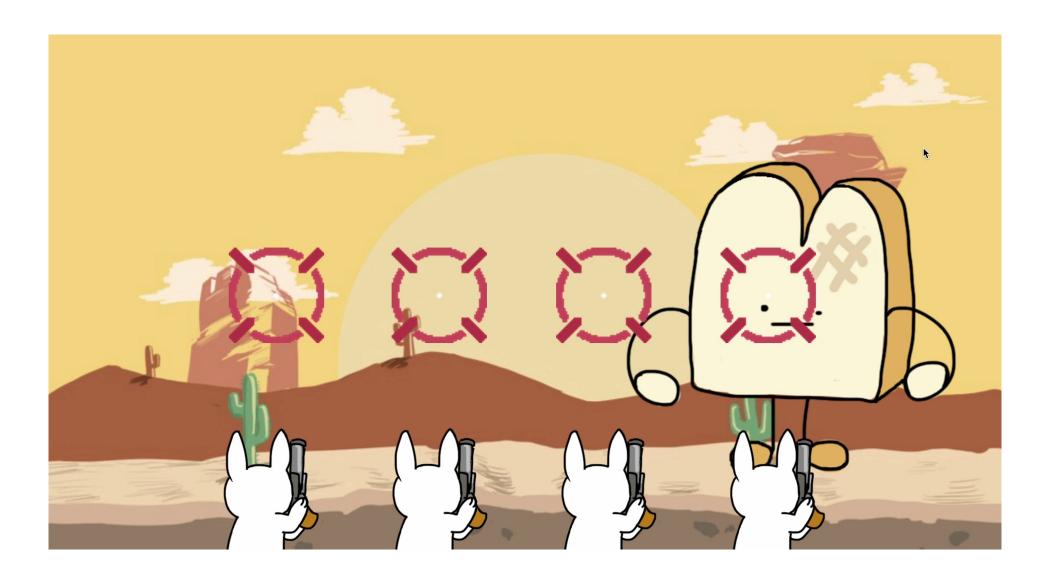
시스템 구성

- 1. 최대 5인 플레이어
- 2. 2D 탑다운 방식으로 플레이어가 맵을 이동하며 이벤트(적 발견) 발생 시 문제 풀이
- 3. 최종 보스 처치 시 플레이 종료, 모든 플레이어 전투불가 시 게임 실패
- 4. 게임 종료 시 MVP 및 플레이어 점수 결과 발표

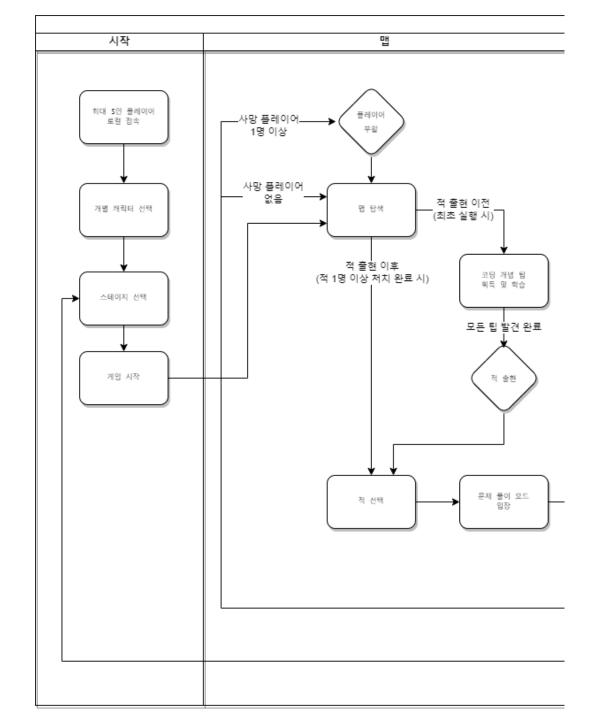
Code-Eat

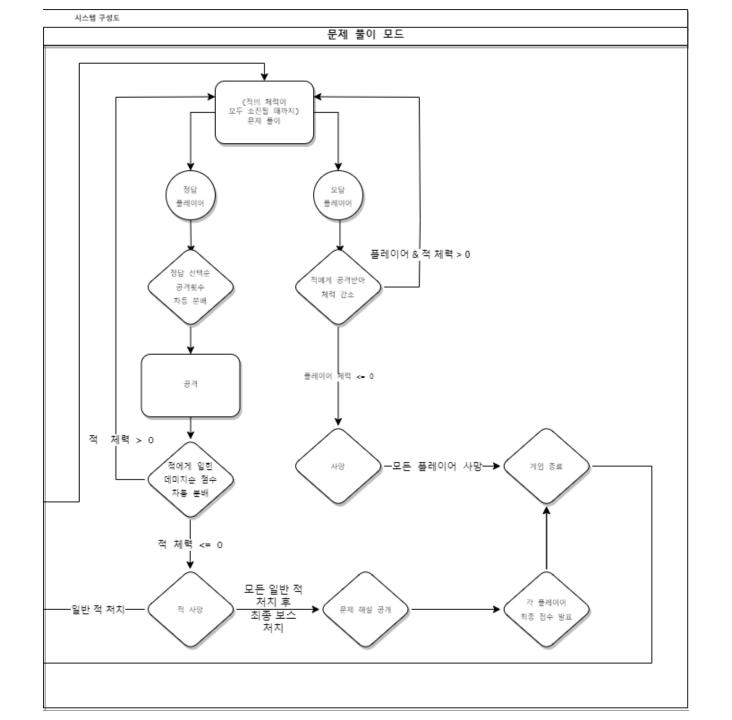


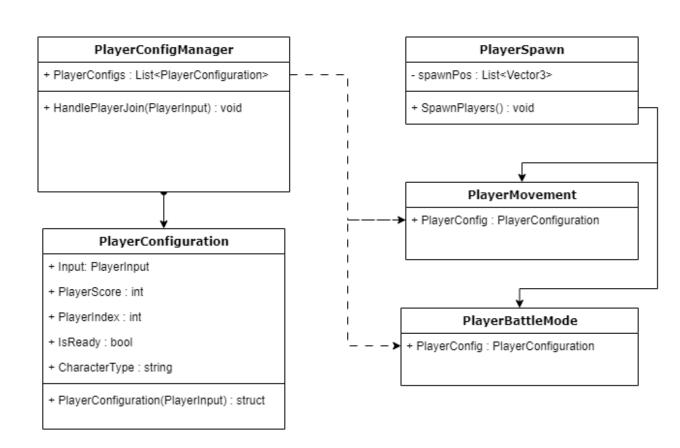
버튼을 눌러서 시작해주세요



상세 설계





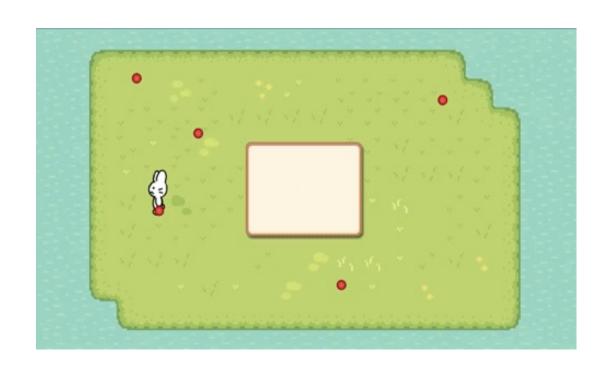






- Configmanager
 - 접속한 플레이어의 정보를 관리하는 매니저

- Playerconfiguration
 - 플레이어 정보를 기록하기 위한 클래스



- 개념 팁 랜덤 생성 및 랜덤 배치
 - 최대 5개의 개념 팁
 - 트리거: 플레이어와 상호작용
 - HashSet: 중복제거
- 개념 팁 확인
 - 플레이어가 상호작용 시 UI 활성화 후 트리거 비활성화
 - 모든 트리거 비활성화 시 적 생성

ProblemManager

- + problemType : enum
- + enemyType : EnemyType
- + SetProblemType(enum) : void
- + SetEnemyType(EnemyType) : void
- NextProblem(): void
- ExitProblem(): void

AnswerManager

- + PlayerList : List<PlayerBattleMode>
- + answerRank : int
- + GetAnswerIndex(): int
- MarkPlayerAnswer(): void
- DistributeBullet(): void
- GetWinner(): PlayerBattleMode
- ShowResult(): void

PlayerBattleMode

- + PlayerConfig : PlayerConfiguration
- + selectedAnswerIndex : int
- + playerHealth : int
- + bulletCount : int
- + IncreaseScore(): void

PlayerBattleMode

- + PlayerConfig : PlayerConfiguration
- + selectedAnswerIndex : int
- + playerHealth : int
- + bulletCount : int
- + IncreaseScore(): void

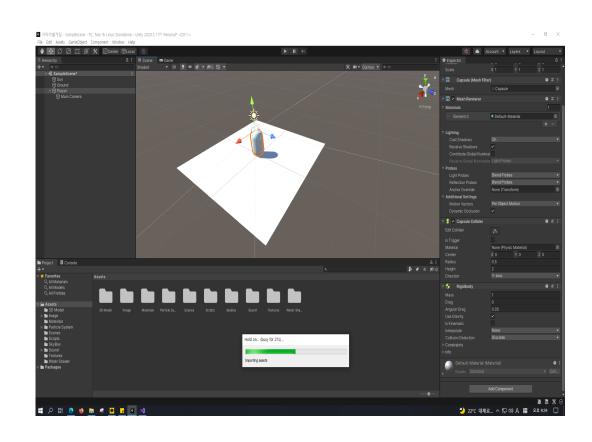
PlayerBattleMode

000

- + PlayerConfig: PlayerConfiguration
- + selectedAnswerIndex : int
- + playerHealth : int
- + bulletCount : int
- + IncreaseScore(): void

- ProblemManager
 - 스테이지에 맞는 문제 유형 설정 & 반복 생성
- AnswerManager
 - 답안 채점 & 결과에 따른 총알 부여
- PlayerBattleMode
 - 공격 모드 관련 요소의 변동 제어

적용 기법 및 기술



- 유니티
 - 게임 개발 엔진
- 에셋 스토어, itch.io
 - 무료/상업용 에셋 라이브러리

핵심과제

- 1. 게임 성능 측면
 - 문제 풀이 시스템
 - 공격 시스템
 - 스코어링 시스템

2. 제작 의도 측면

- 학습도
- 재미

일정 계획





3.1 프로젝트 의의

성능 측면 달성 목표

1. 문제 풀이 시스템

- 한 유형 내에서 최소 2개 이상의 중복되지 않는 문제를 자동적으로 생성한다.
- 최종 결과 화면에 개인 오답률을 표시한다.

2. 공격 시스템

- 플레이어 답안 선택 순서를 정확하게 파악하고, 정답 선택 순서에 따라 공격 횟수를 차등적으로 부여한다.
- 적의 피격 부위 설정에 따라 점수를 차등 부여한다.

3. 스코어링 시스템

- 특정한 유형 내에서 최소 2개 이상의 문제를 자동적으로 생성한다.

제작 의도 측면 달성 목표

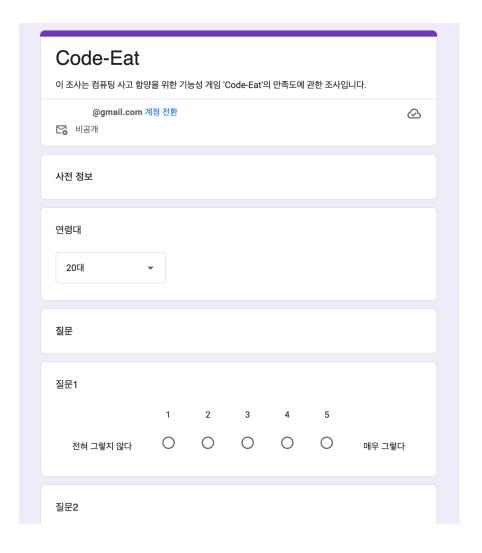
외부 테스팅(10~50대 대상)

1. 학습도

 같은 스테이지를 2번 플레이 했을 때 오답율 감소

2. 재미

- 구글폼을 배포하여 익명 만족도 조사
- & 만족도(리커트 5점 척도 기준) 3.5이상



의의

사용자 관점)

- 사람들과 즐기면서 코딩에 대한 두려움 완화
- 스테이지 시스템을 통한 개인별 취약 개념 학습 가능
- 각 플레이어 선택답안 숨기기 → 공정성 보완한 경쟁 학습

개발자 관점)

- 로컬 멀티플레이 시스템의 이해
- 문제 자동 생성 시스템 구현 시도
- 개발 로직의 이해, 게임 개발 사이클 경험