창직종합설계프로젝트1 제안서

Do Mine

B611125유민호, B611064민정인,B611154이유림

1 개요와 전체적 게임 컨셉

2 프로그램 구조(클라이언트, 서버, DB)

3 인게임 플레이 진행과정

4 게임 플레이 세부 정보

5 개발 분업

6 사용 플랫폼 설명과 관련논문

1. 개요와 게임 컨셉
   * 개요
     + 게임 제목 : Do Mine

* 게임 장르 : 2D 온라인 소셜 디덕션 게임 (A.K.A 마피아게임장르)
* 플레이 시간 : 8~15분
* 플레이 인원 : 3~10인
  + 게임 시스템 컨셉
* 소모임규모 인원이 친목목적으로 즐길 수 있는 마피아게임
* 사보타지와 광부 두 팀으로 진행
* 광부는 시간 내로 보물상자에 있는 금을 찾아 시작지점으로 가져와야 함
* 사보타지는 광부사이에 숨어 광부들의 활동을 방해해야 함
* 동종 장르의 대표적인 게임들과는 다르게 사망을 통한 게임배제는 없음
* 광부들도 금을 입금한 광부만 승리하므로 광부 간의 치킨게임 유도
  + 게임 배경 컨셉
    - 깊은 지하 용암이 차오르는 금광이 배경
    - 욕심이 많은 광부들이 서로 먼저 금을 차지하기위해 경쟁함
    - 광부들이 금을 찾지 못하도록 사보타지들이 광부인 척 하면서 방해함

1. 프로그램 구조
   * 클라이언트프로그램 (Unity로 개발, Scene별로 구분)
     + Title Scene
       - 게임 실행
         * 서버와 연결
         * (첫 실행시)유저 고유코드 부여 및 닉네임설정
       - 메인 화면
         * Join Game : 이미 존재하는 게임 방에 입장

현재 활성화된 방 목록을 출력 후 클릭하여 입장

방 고유코드 입력을 통한 직접 입장

* + - * + Host Game : 새로운 게임 Room 생성

방 이름, 게임시간, 인원 수 등을 설정 가능

* + - * + How to Play : 게임에 대한 설명
        + Option : 게임 내 설정

볼륨조절, 컨트롤 관련 설정들

* + - * + Nickname Change : 닉네임을 변경

유저 고유키와는 별개의 닉네임

* + - * + Quit Game : 게임종료
    - Ingame Scene
      * 로비 화면 : 현재 방정보와 참가유저목록을 실시간으로 업데이트
        + Game Start : 현재 방 인원들과 게임 시작(Host만 사용가능)
        + Room Settings : 현재 방 정보 변경 (방 이름, 게임시간, 인원 수 등을 설정 가능)
        + Back to Main : 메인 화면으로 이동
      * 인게임 화면 : 게임 플레이하는 공간
        + 게임 시작

1. 서버에서 무작위로 생성한 맵 정보, 역할을 받아옴
2. 서버에서 받은 정보대로 오브젝트생성
   * + - * 인게임 플레이 : 플레이 진행단계에서 설명

Quit Game : 게임을 포기하고 게임 종료로 이동

* + - * + 게임 종료 : 게임 종료 후 승리,패배 정보를 출력

Back to Lobby : 현재 게임 방의 로비로 돌아감

Back to Main : 방에서 나와 TitleScene으로 이동

* 1. 서버
  2. DB
     + Room
       - Bool isActive : 현재 방 활성화 상태
       - Int capacity : 방 최대 인원
       - Int gameTime : 게임 시간
       - String gameKey : 방 고유 키
       - UserList []
         * String nickname : 닉네임 정보
         * String userId : 유저 고유 키
         * Bool isOnline : 유저 현재 접속상태 (장기 미응답시 강퇴)
         * Bool isHost : 호스트인지를 나타냄
     + Ingame
       - String gameKey : 방 고유 키
       - Int map[][] : 맵의 구조를 저장
       - Int timeLast : 남은 시간
       - Int goldAmount : 게임 내 금의 개수
       - playerStatus[] : 플레이어의 상태를 저장
         * String userId : 유저 고유 키
         * Int locationX, locationY : 유저의 xy좌표값
         * Bool isSabotage : 현재 유저가 사보타지인지 확인
         * Int playerActing : 현재 유저가 하고있는 행동 저장(중립,걷기,채굴,사보타지활동등)
         * bool isPlaying : 현재 유저가 활동 하고 있는지 저장
         * inventory : 보유한 아이템 일람 (추후 추가가능)

bool gold : 금

int barricade : 바리케이드

##서버는 포톤 기능 몇 개 들면서 어캐쓸지 간략한 설명

##DB구조 적어야함

1. 인게임 플레이 진행단계
   * 시작단계 : 플레이어의 직업 결정, 맵 생성과 금괴배치
     + 사보타지 : 33퍼센트를 넘지않게 랜덤하게 배정(ex 3~4 0~1인, 5~7 1~2인, 8~10 2~3인)
     + 금괴수 : 인원대비 40~50프로 승리조건은 여기서 -1 (34인 2개, 567인 3개, 8910인 4개)
     + 금괴위치 : 중앙의 안전지대를 제외한 나머지 공간에 랜덤 배치
     + 맵 크기 : 미정, 최대 한계는 존재
   * 게임 플레이 초반단계 (초반 1/3) : 광부와 사보타지는 길을 개척하면서 금을 찾으면서 각자의 임무를 한다.
     + 경로 개척 : 맵 한계 이내에선 자유롭게 벽을 제거 후 경로개척가능
     + 비상탈출기능 : 유저들의 방해로 어딘가 갇히거나 길을 잃었을 시 사용 가능하고 금과 모든 소지품을 잃고 시작위치로 귀환함
     + 보물상자 : 한 게임에 여러 개 생성되며 금이나 아이템을 보관하고 있음 빈 상자도 있을 수 있음
   * 게임플레이 중반단계(중반 1/3) : 서로 간의 견제와 방해를 할 수 있고 금을 차지한 광부는 안전지대의 금고에 입금 후 광부승리가 되면 승리하게 됨.
     + 사보타지들의 특수능력이 사용가능해짐
     + 금을 발견하면 금을 들고 이동하여 시작지점에 입금해야 함
   * 게임플레이 후반단계(후반 1/3) : 제한시간이 다가오면서 맵이 끝에서부터 좁아지기 시작함 이미 성공한 광부는 다른 광부가 금을 차지하도록 돕고, 사보타지는 시간을 지연 시켜야 함
     + 승리에 필요한 금이 입금되야 승리하기 때문에 성공한 광부가 타 광부에 협력함
     + 맵이 끝에서부터 파괴되는데 해당 위치에 있는 플레이어는 거점으로 자동 귀환 되고 해당위치의 금은 파괴됨.
   * 게임이 끝나는 조건
     + 광부가 승리에 필요한 금을 모두 입금 (입금한 광부 승리)
     + 금이 파괴되어 승리에 필요한 양보다 금이 부족한 경우 (사보타지 승리)
     + 15분 시간종료시 (사보타지 승리)
2. 게임 세부 정보
   * 맵 생성
     + 기본적인 n \* m 형식의 직사각형 형태의 맵
     + 해당 칸과 1대1대응이되도록 배열을 만들고 랜덤으로 값을 배정
     + 단, 맞닿은 칸들끼리는 맞닿은 부분에 장애물 유무가 동일해야 함.
     + 각 장애물들은 곡괭이를 통해 제거 가능 및 바리케이트를 통해 생성 가능
     + 금의 경우, 일정 위치에 생성
   * 기본 조작
     + 이동
       - 이동방식 : 가상 패드 형식
       - 카메라이동 : 탑뷰, 캐릭터에 고정
     + 경로 개척
       - 캐릭터에서 가장 가까운 벽 중, 캐릭터가 바라보고 있는 방향의 벽에 표시 생성 및 키 활성화
       - 키 입력 시 일정 시간 대기시간 후 벽 파괴(대기시간 중엔 행동불능)
       - 벽 파괴 횟수 한정, 베이스캠프로 귀환 시 횟수 회복 가능
   * 금괴 관련 상호작용
     + 금괴 발견 및 범위 내에 도달할 시 버튼 활성화후 버튼 누를 시 획득
     + 금괴 획득 시 캐릭터의 이동속도 감소
     + 금괴 내려놓기 가능
     + 베이스캠프 도착 시 자동으로 회수 및 점수로 환산
   * 통신 및 게임화면 업데이트
     + 맵 정보
       - 배열 형태로 서버가 DB에 저장 후 클라이언트에 실시간으로 동기화
       - 클라이언트는 맵의 Object들과 상호작용할 때마다 서버에 신호를 보내고 서버는 이에 맞춰 DB를 수정함
     + 이동 정보
       - X,Y좌표값으로 지속적으로 클라이언트에서 DB로 전달함 (사용할 서버 플랫폼에서 내장 된 기능 사용 예정)
     + 게임 방 정보
       - 유저의 고유 키 값과 닉네임 값, 접속 상태 등을 리스트로 저장
       - 입장과 퇴장은 해당 클라이언트에서 신호를 내고 서버가 처리
       - 강제종료한 플레이어는 서버가 찾아내어 퇴장
       - 게임 시작시에 방을 비활성화 상태로 만들고 게임이 끝난 후에 다시 활성화
       - 방에 남은 인원이 없으면 방 자동 삭제
       - 호스트가 떠날 시 배열의 가장 처음 플레이어가 호스트가 됨
   * 시야
     + 캐릭터 주변 일정 범위 내 시야 확보
     + 시야 범위 밖은 보이지 않고, 시야 범위 내라도 벽 존재 시 보이지 않음
   * UI 정보

* 미니맵
  + 현재 캐릭터의 위치, 보물상자의 위치를 표시(나침반형식)
* 가상패드
  + 캐릭터를 이동시킬 때 사용
* 상호작용키
  + 현재 플레이어에게 활성화된 활동을 하는 키(경로개척 금 줍기 등)
  + 사보타지
    - 목표
      * 시간 내에 광부가 필요한 개수만큼 금을 얻지 못하게 막아야 한다.
* 특수 능력
  + - * 기본적으로 광부가 갖는 기능은 모두 사용가능
      * 바리케이트 소환으로 광부의 이동을 막을 수 있음
      * 광부보다 넓은 시야를 가짐
      * 사보타지가 입금을 하고 광부 승리시 광부 전원 승리
      * 은신이나 이동속도 증가 등의 다른 기능 생각 중

1. 개발 분업과 규칙
   * 개발 분업
     + 게임 플레이 구현
       - 민호 : 인게임 플레이
       - 정인 : 로비 등 씬 연결
       - 유림 : 인게임 UI
     + 게임 서버 구현
       - 인게임 통신 : 민호, 정인
       - DB : 유림
2. 사용 플랫폼 설명과 관련논문
   * Unity
     + 게임 엔진 플랫폼

##이번에 찾은거 써봅시다

1. 추가 가능기능