창직종합설계프로젝트1 제안서

Do Mine

B611125유민호

B611064민정인

B611154이유림

목차

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

7. 참고문헌

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

* 게임 장르 : 2D 온라인 소셜 디덕션 게임 (A.K.A 마피아게임장르)
* 플레이 시간 : 8~15분
* 플레이 인원 : 3~10인
  1. 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
  1. 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
   1. 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

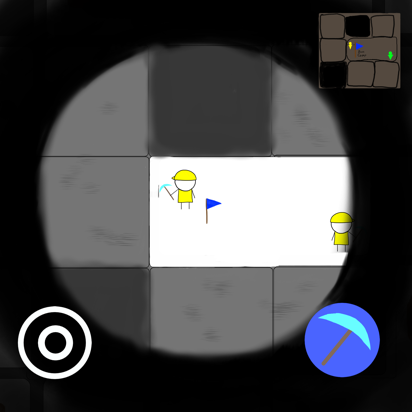
Back to Main : 방에서 나와 TitleScene으로 이동

* 1. 서버(PUN으로 개발, 기능 별로 구분)
     + 서버 – DB간 결합 : PUN만으로는 DB 저장에 어려움이 있어 FireBase 통해 연결함
     + 매치메이킹
       - 빠른 참가
         * 무작위 매치메이킹 기능을 사용 – PhotonNetwork.JoinRandomRoom()
       - Room 옵션 : Room을 생성한 호스트에게 권한을 부여하여 해당 Room의 인원 수, 공개/비공개 여부, 기타 게임 정보 등을 설정할 수 있게 함.
     + 인게임
       - 네트워크 객체 : 게임 내부에서 객체들의 변화를 각 플레이어들이 갱신받을 수 있게 함.
       - 애니메이션 : Mecanim 동기화 컴포넌(PUN 내장)를 통해 캐릭터들의 상태 동기화
  2. DB(FireBase로 개발, 저장할 데이터별로 구분, 예시)
     + 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
       - Bool open : 공개/비공개 방 여부
     + 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

1. 인게임 플레이 진행단계
   1. 시작단계 : 플레이어의 직업 결정, 맵 생성과 금괴배치
      * 사보타지 : 33퍼센트내외 랜덤하게 배정(ex 3~4 0~1인, 5~7 1~2인, 8~10 2~3인)
      * 금괴수 : 인원대비 40~50프로 (34인 2개, 567인 3개, 8910인 4개)
      * 금괴위치 : 중앙의 안전지대를 제외한 나머지 공간에 랜덤 배치
      * 맵 크기 : 미정, 최대 한계는 존재
   2. 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 : 어딘가 갇히거나 길을 잃었을시 사용가능, 모든 소지품을 잃고 시작위치로 귀환
      * 보물상자 : 여러 개 생성되며 금이나 아이템을 보관하거나 빈 상자로 존재
   3. 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
   4. 게임플레이 후반단계(후반 1/3) : 제한시간이 다가오면서 맵이 끝에서부터 좁아짐
      * 승리에 필요한 금이 입금되야 승리하기 때문에 성공한 광부가 타 광부에 협력함
      * 맵이 끝에서부터 파괴되는데 해당 위치의 플레이어는 자동 귀환 되고 해당 위치의 금은 파괴
   5. 게임이 끝나는 조건
      * 광부가 승리에 필요한 금을 모두 입금 (입금한 광부 승리)
      * 15분이 지나거나 금이 파괴되어 승리에 필요한 양보다 금이 부족한 경우 (사보타지 승리)
2. 게임 세부 정보
   1. 맵 생성
      * 기본적인 n \* m 형식의 직사각형 형태의 맵
      * 해당 칸과 1대1대응이되도록 배열을 만들고 랜덤으로 값을 배정
      * 단, 맞닿은 칸들 끼리는 맞닿은 부분에 장애물 유무가 동일해야 함.
      * 각 장애물들은 곡괭이를 통해 제거 가능 및 바리케이드를 통해 생성 가능
      * 금의 경우, 일정 위치에 생성
   2. 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
   3. 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
   4. 통신 및 게임화면 업데이트
      * 맵 정보
        + 배열 형태로 서버가 DB에 저장 후 클라이언트에 실시간으로 동기화
        + 클라이언트는 Object들과 상호작용시 서버에 신호를 보내고 서버는 이에 맞춰 DB를 수정
      * 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
   5. 시야
      * 캐릭터 주변 일정 범위 내 시야 확보
      * 시야 범위 밖은 보이지 않고, 시야 범위 내라도 벽 존재 시 보이지 않음
   6. UI 정보

 <- UI 예시

* 미니맵
  + 현재 캐릭터의 위치, 보물상자의 위치를 표시(나침반형식)
* 가상패드
  + 캐릭터를 이동시킬 때 사용
* 상호작용키
  + 현재 플레이어에게 활성화된 활동을 하는 키(경로개척 금 줍기 등)
  1. 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 분업
   * 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 설명
   * Unity
     + 게임 엔진 플랫폼
     + 다양한 플랫폼지원(모바일)
     + 통합 개발 환경 제공
   * PUN(Photon Unity Network)
     + Photon에서 제공하는 서버 시스템 서비스 중 하나
     + Unity 내의 UnityAssetStore를 통해 설치하여 Unity와 Photon을 적절하게 연결할 수 있음.
   * FireBase
     + DB시스템
     + PUN과의 결합을 통해 Unity – PUN – FireBase 방식으로 클라이언트 – 서버 – DB 결합을 하고자 함.
     + NoSQL 형식을 차용함.

* 셋 모두 무료라는 점에서 프로젝트 진행에 있어 무료라는 점으로 인해 비용적인 부담을 줄일 수 있음.

1. 참고문헌

* Unity Manual : <https://docs.unity3d.com/kr/current/Manual/UnityManual.html>
* PUN Tutorial : <https://doc.photonengine.com/ko-kr/pun/v2/demos-and-tutorials/pun-basics-tutorial/intro>
* FireBase Docs : <https://firebase.google.com/docs/database?hl=ko>