|  |
| --- |
| 네트워크 게임 프로그래밍 |
| 개구리 : 비상 |
| [문서 부제 입력] |

|  |
| --- |
|  |

목차

1. 애플리케이션 기획
   1. 게임 컨셉
   2. 플레이 방식
   3. 게임 소개
   4. 플레이 방법
2. High-Level 디자인
3. Low-level 디자인
4. 역할 분담
5. 개발 일정
6. 개발 환경

애플리케이션 기획

게임 컨셉: 발판을 이용해 적들의 공격을 피해가며 두 명의 플레이어가 모두 버튼을 밟아 스테이지를 클리어 하는 협동 게임

플레이 방식: 2인 협동 멀티 플레이 게임

게임소개: 맵의 특정 지역에 버튼 2개가 배치되어 있다. 맵에는 원거리 공격 몬스터와 지정된 형식으로 돌아다니는 몬스터가 있다. 플레이어는 마우스 클릭을 통해 원하는 위치에 발판을 생성할 수 있으며 생성 가능 개수에 제한이 있다.

생성된 발판과 기존 발판을 이용하여 2개의 버튼에 도달하여 누르고 있을 시 스테이지가 클리어 된다. 스테이지는 {}개 있으며 진행할수록 난이도가 상승한다. 플레이어가 몬스터와 충돌하거나 몬스터의 공격에 맞을 시 생성된 발판은 초기화되며 모든 플레이어가 출발 지점으로 돌아간다.

플레이 방법:

1. 게임 접속

2. 아이디 입력을 통한 로그인

3. 팀원 입장 시 게임 시작

4. 방향키와 스페이스바를 이용해 움직인다.

5. 마우스 클릭을 통해 발판을 생성한다.

6. 두 명의 플레이어가 모두 발판을 밟을 시 스테이지 클리어 or 게임 승리

7. 플레이어가 몬스터와 혹은 공격에 충돌 시 발판 및 플레이어 위치 초기화

High-Level디자인

플로우 차트

Login

<server> <client>

Socket()

Socket()

bind()

listen()

connect()

accept()

(ID) Send()

(ID) recv()

(checkID) Send()

(checkID)

recv()

.

saveID()

GameStart()

GameStart()

클라이언트가 서버에 접속을 요청하고 서버는 요청을 수락한다.

각 클라이언트는 자신이 사용할 ID를 입력하고 서버에 전송한다.

서버는 클라이언트가 입력한 ID를 받아서 이미 존재하는 ID인지를 파악해 그 결과를 클라이언트에 전송한다.

이미 존재하는 ID라면 ID입력 후 전송을 반복한다.

ID가 존재하지 않으면 서버에 저장하고 게임을 실행한다.

Lobby

Ingame

<server> <client>

GameStart()

GameStart()

InputKey()

(key) recv()

(key) send()

collision()

(position) recv()

(position) send()

render()

<server> <client>

GameStart()

GameStart()

Inputmouse()

(x,y) recv()

(x,y) send()

collision() ???

(collision) recv()

(collision) send()

saveblock()

render()

클라이언트는 키보드 입력으로 맵을 이동하고 마우스 클릭으로 블록을 생성한다.

클라이언트의 키보드 값 혹은 마우스의 X, Y좌표를 서버에 전송 한다.

서버는 클라이언트에게서 받은 정보로 충돌체크를 진행하고 연산을 실행한다.

충돌 여부와 연산 결과를 모든 클라이언트에 전송한다.

클라이언트는 전송 받은 결과를 가지고 렌더링을 실행한다.

또한 몬스터 혹은 몬스터의 공격이 플레이어와 충돌했을 경우에도 서버에서 충돌 검사를 수행하여 그 결과를 클라이언트에게 전송하면 모든 블록과 클라이언트의 위치가 초기화 된다.

Low-Level 디자인

역할 분담

|  |  |  |
| --- | --- | --- |
| 담당 | Client | Server |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

개발 일정

|  |  |  |  |
| --- | --- | --- | --- |
|  | 강승준 | 유영준 | 오성훈 |
| 10/28 목 | 게임 선정 및 추진 계획서 작성 | | |
| 29 금 |
| 30 토 |
| 31 일 |
| 11/1 월 |
| 2 화 | 추진 계획서 제출 | | |
| 3 수 |  |  |  |
| 4 목 |  |  |  |
| 5 금 |  |  |  |
| 6 토 |  |  |  |
| 7일 |  |  |  |
| 8 월 |  |  |  |
| 9화 |  |  |  |
| 10 수 |  |  |  |
| 11 목 |  |  |  |
| 12 금 |  |  |  |
| 13 토 |  |  |  |
| 14 일 |  |  |  |
| 15 월 |  |  |  |
| 16 화 |  |  |  |
| 17 수 |  |  |  |
| 18 목 |  |  |  |
| 19 금 |  |  |  |
| 20 토 |  |  |  |
| 21 일 |  |  |  |
| 22 월 |  |  |  |
| 23 화 |  |  |  |
| 24 수 |  |  |  |
| 25 목 |  |  |  |
| 26 금 |  |  |  |
| 27 토 |  |  |  |
| 28 일 |  |  |  |
| 29 월 |  |  |  |
| 30 화 |  |  |  |
| 12/1 수 |  |  |  |
| 2 목 |  |  |  |
| 3 금 |  |  |  |
| 4 토 |  |  |  |
| 5 일 |  |  |  |
| 6 월 |  |  |  |
| 7 화 |  |  |  |
| 8 수 | 최종 본 제출 | | |

개발 환경

OS: Windows 10

IDE: Visual Studio 2019

API: Windows Socket API(WSA)

개발 언어: C/C++