네트워크 게임 프로그래밍 (01)

Term Project 추진계획서

3팀

2020184015 박가현

2020184025 이승희

2020184032 정다민

목차

1. **프로그램 개요**
2. **게임의 특징**
3. 게임 진행
4. 장애물
5. 아이템
6. 조작법
7. **개발 환경**
8. **High Level Design**
9. 클라이언트
10. 서버
11. **Low Level Design**
12. 클라이언트
13. 서버
14. **역할 분담 및 개발 일정**
15. **프로그램 개요**

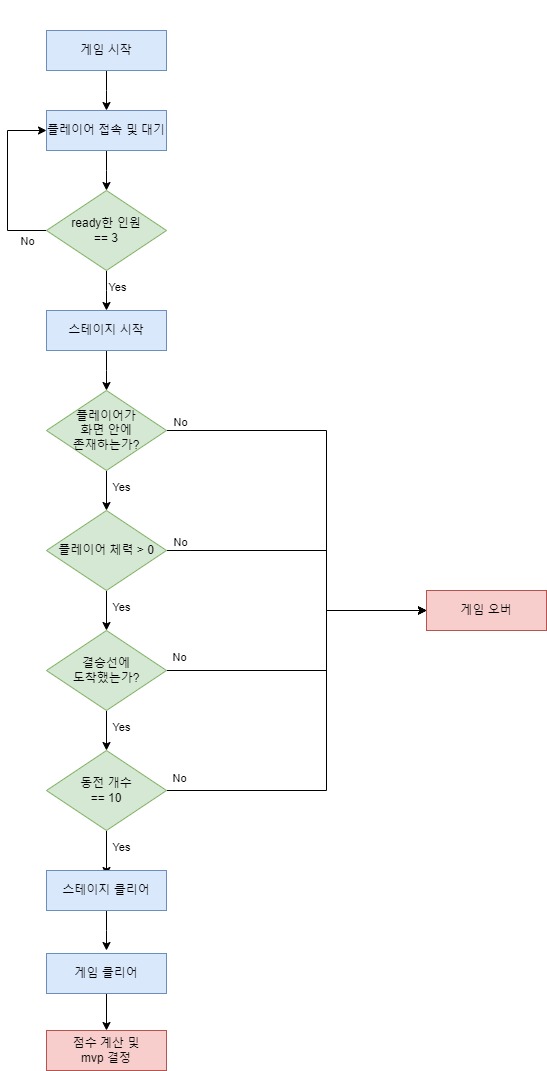
|  |  |
| --- | --- |
| 게임명 | To Be a Butterfly |
| 교과목 | 윈도우프로그래밍 |
| 작업자 | 2020184025 이승희 |
| 장르 | 캐주얼 아케이드 게임 |
| 플레이어 수 | 3명 |

****

****

****

1. **게임의 특징**

****

1. 게임 진행

- 게임의 스테이지는 총 1개이고, 캐릭터는 총 3개이다.

- 캐릭터는 중력의 영향을 받아서 아래로 떨어진다.

- 캐릭터의 체력은 100이다.

- 화면에 본인의 체력 UI와 팀원들의 체력 UI가 존재한다.

- 스크롤이 자동으로 진행되고, 캐릭터가 화면 아래로 벗어나면 게임 오버된다.

- 스테이지 클리어 후에 동전을 가장 많이 모은 플레이어는 체력이 상승된다.

- 모든 스테이지 클리어 후에 동전을 가장 많이 모은 플레이어는 MVP가 된다.

- 이벤트를 사용하여 스레드 동기화를 지원한다.

1. 장애물
2. 일반 구름

- 스테이지를 이루는 기본 장애물이다.

- 캐릭터가 통과할 수 있지만, 그동안 체력과 속도가 감소한다.

1. 먹구름

- 구름의 일정 범위 아래까지 비를 내린다.

- 캐릭터가 비를 맞으면 체력이 감소한다.

- 캐릭터가 비를 맞으면 아래로 떨어지고 일정 시간동안 점프할 수 없다.

1. 번개 구름

- 구름의 일정 범위 아래까지 번개를 내리친다.

- 캐릭터가 번개에 맞으면 체력이 감소한다.

- 캐릭터가 번개에 맞으면 일정 시간동안 캐릭터가 멈춘다.

1. 아이템
2. 동전

- 세 플레이어의 총 획득 수가 일정 개수 이상이어야 클리어 가능하다.

1. 하트

- 획득 시 체력이 5 회복된다.

1. 조작법

- 키보드의 방향키를 이용하여 상, 하, 좌, 우로 이동한다.

1. **개발 환경**

- 운영체제: Windows 10

- 통신 프로토콜: TCP/IP

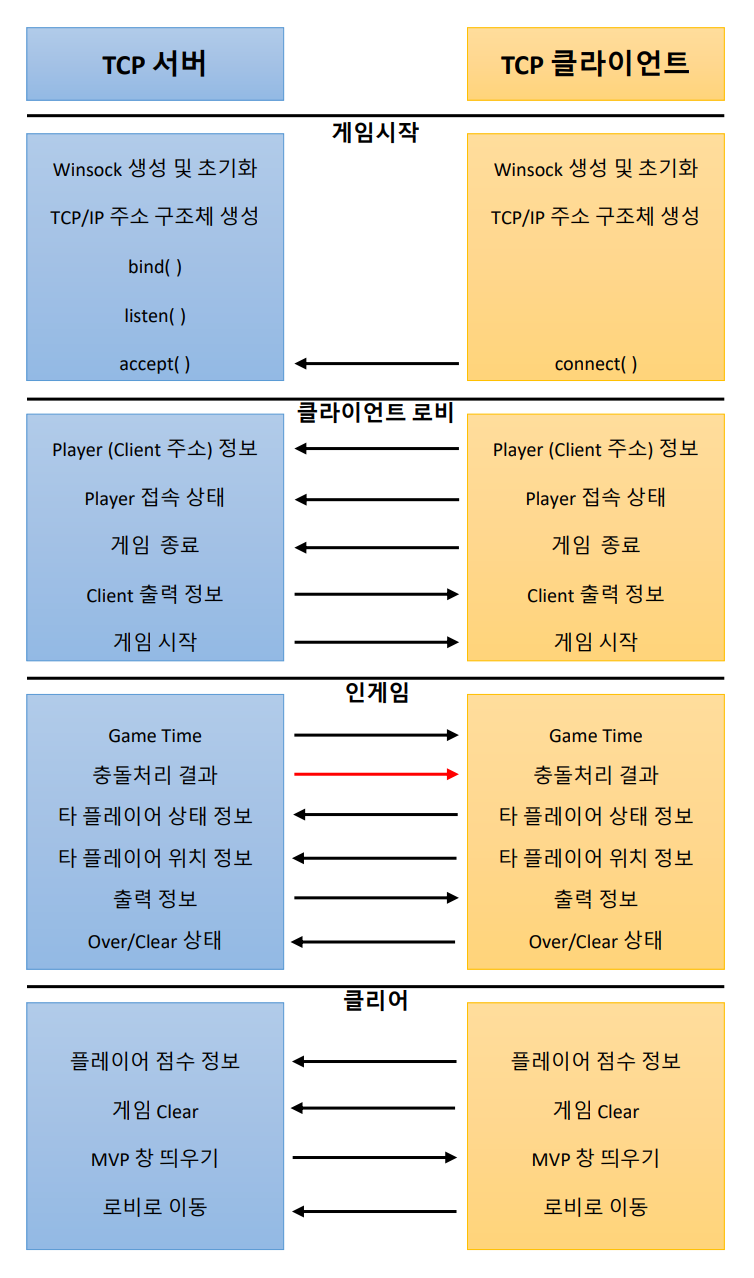
- 사용 언어: C/C++

- 컴파일러: Visual Studio 2022

- 라이브러리: OpenGL

- 버전 관리: github

1. **High Level Design**

(서버 – 클라이언트 흐름도)

|  |  |
| --- | --- |
| 서버 | 클라이언트 |
|  |  |

1. 클라이언트

- 프로그램 실행 후 서버에 접속한다.

- 서버에서 접속 인원을 받는다.

- 접속 인원이 3명이 되면 게임을 시작한다.

- 게임 진행 중에는 서버로부터 게임 데이터를 받아 게임 화면을 렌더링한다.

- 플레이어의 ~~충돌 및~~ 상태 정보를 서버로 보낸다.

- 게임 오버 시 게임 오버 창을 렌더링하고, 다른 클라이언트들이 게임을 클리어할 때까지 대기한다.

- 게임 클리어 시 점수 및 MVP 결과를 받아 결과 창을 렌더링한다.

- 프로그램을 종료한다.

1. 서버
2. 서버 실행

- WinSock을 초기화하고 대기 소켓을 생성하고 bind 함수와 listen 함수를 통해 지역 IP 주소와 지역 포트 번호를 결정하여 클라이언트의 접속을 기다리는 대기 소켓 상태로 만든다.

- 클라이언트가 connect 함수로 서버에 접근하면 accept된 클라이언트를 위한 소켓을 생성하고 로비 화면으로 전환하는 메시지 정보를 송신한다.

1. 플레이어 별 스레드를 이용한 데이터 통신

- 프로그램을 실행한다.

- 3개의 클라이언트가 접속할 때까지 대기한다.

- 클라이언트가 접속하면 접속 인원을 보낸다.

- 접속한 플레이어 별로 스레드를 할당한다.

- 게임이 시작되면 event를 비신호 상태로 생성한다. 접근 순서가 중요해 이벤트를 사용하여 동기화를 진행한다.

- Wait\*() 함수를 호출해 이벤트가 신호 상태가 될 때까지 대기한다.

- 동전과 충돌이 있을 때 이벤트 신호 상태를 비신호 상태로 바꾼다.

- 이벤트 신호 상태를 확인하여 신호 상태일 때 게임 데이터를 갱신한다.

- 통신 소켓을 생성하고 스레드 내부에서 IP 주소 정보를 받는다.

- 스레드 내부에서 클라이언트가 보낸 플레이어의 ~~충돌 및~~ 상태 정보를 수신 받는다.

- 각각의 클라이언트들에게 다른 클라이언트의 위치 정보, 상태 정보 및 서버 시간 정보를 지속적으로 송신한다.

- 게임 종료 혹은 클리어 조건에 부합하는 순간 클라이언트 별로 갱신된 상태를 전송 후 반영한다.

- 게임 종료 시에는 게임 종료 신호를 클라이언트에 전송하여 결과 화면으로 넘어간다.

- 결과 화면에서는 MVP 플레이어 정보를 클라이언트에 보내 결과를 출력할 수 있게 한다.

- 이벤트가 필요하지 않으면 CloseHandle() 함수를 호출하여 이벤트를 제거한다.

- 프로그램 종료 메시지를 받게 되면 클라이언트에 해당하는 스레드의 소켓을 종료하고 스레드를 제거한다.

1. **Low Level Design**
2. 클라이언트

|  |  |  |
| --- | --- | --- |
| class Player | | int id,  int px, py,  int status,  int jumpForce,  int hp,  int coinCount,  int animIndex |
| class Cloud | | int cx, cy,  int type,  int index |
| void | SendClientInfo() | 클라이언트 주소 정보 송신 |
| void | RecvMapData() | 맵 정보 수신 |
| void | RecvGameStartEvent() | 게임 시작 이벤트 수신 |
| void | RecvGameTime() | 게임 시간 수신 |
| void | SendClientPos() | 위치 정보 송신 |
| void | RecvOtherClientPos() | 다른 플레이어 위치 정보 수신 |
| void | SendClientStatus() | 플레이어 상태 정보 송신  - 구름 충돌 시, 사망 시, 목표 지점 도달 시 |
| void | RecvOtherClientStatus() | 다른 플레이어 상태 정보 수신 |
| void | RecvGameOverEvent() | 게임 오버 이벤트 수신 |
| void | RecvGameClearEvent() | 게임 클리어 이벤트 수신 |

1. 서버

|  |  |  |
| --- | --- | --- |
| DWORD WINAPI | ClientThread() | 각 클라이언트와 통신을 담당할 스레드 함수 |
| void | RecvClientInfo() | 클라이언트 주소 정보 수신 |
| void | SendMapData() | 맵 정보 송신 |
| void | SendGameStartEvent() | 게임 시작 이벤트 송신  - 접속 인원 충족 시 |
| void | SendGameTime() | 게임 시간 송신 |
| void | RecvClientPos() | 클라이언트 위치 정보 수신 |
| void | SendOtherClientPos() | 다른 클라이언트 위치 정보 송신 |
| void | MapCollision() | 맵과 플레이어 간 충돌 체크 |
| void | ItemCollision() | 아이템과 플레이어 간 충돌 체크 |
| void | RecvClientStatus() | 클라이언트 상태 정보 수신  - 사망 시 다른 두 플레이어의 status 검사  - 목표 지점 도달 시 동전 개수 확인 및 다른 두 플레이어의 status 검사 |
| void | SendOtherClientStatus() | 다른 클라이언트 상태 정보 송신 |
| void | SendGameOverEvent() | 게임 오버 이벤트 송신  - 클라이언트 상태 정보 수신 후 모든 플레이어가 사망일 경우 |
| void | SendGameClearEvent() | 게임 클리어 이벤트 송신  - 클라이언트 상태 정보 수신 후 모든 플레이어가 목표 지점 도달일 경우 |

1. 역할 분담 및 개발 일정



