

REPORT

2025 네트워크 게임 프로그래밍 팀프로젝트 추진계획서



한국공학대학교
TECH UNIVERSITY OF KOREA

학과	게임공학과
과목명	네트워크 게임 프로그래밍 (03)
학생	2023182041 이용재 2020184029 이현석 2020180045 황태규
제출일	2025.10.25

목차

I. 애플리케이션 기획

1. 게임 설명
2. 월드
3. 플레이어
4. 아이템
5. 몬스터
6. 조작
7. 상호작용
8. 멀티플레이 추가 기능

II. High-Level 디자인

III. Low-Level 디자인

IV. 팀원 별 역할분담

V. 개발 환경

VI. 개발 일정

1. 이용재
2. 이현석
3. 황태규

I. 애플리케이션 기획

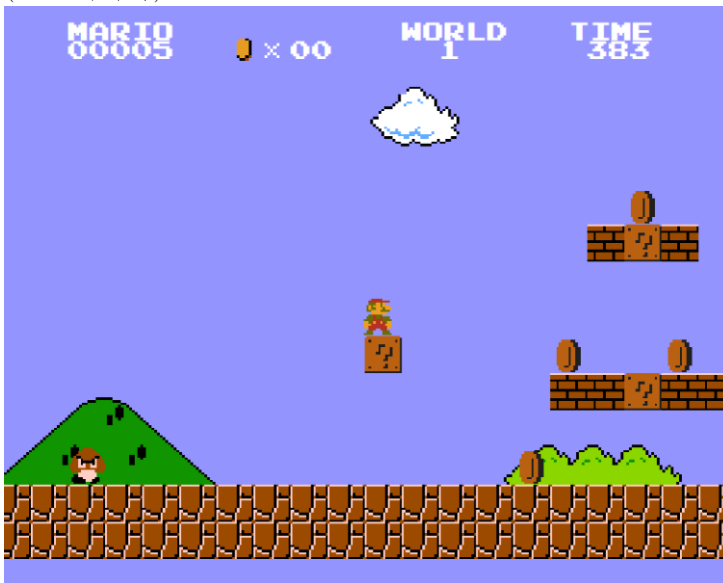
1. 게임 설명

2025-1 윈도우 프로그래밍 서 WINAPI 를 통하여 개발 | 이용재

본 게임은 'SUPER MARIO BROS'의 모작으로, 2D 사이드 스크롤 플랫폼 액션 게임이며 점프, 이동, 몬스터 공격 등을 통하여 각 스테이지를 클리어하며, 최종 구간까지 도달하는 것이 목표입니다.

2. 월드

(1 스테이지)



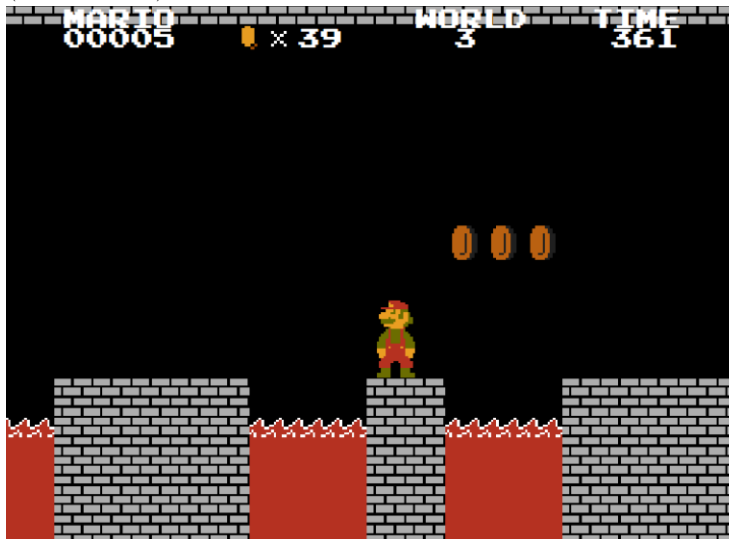
테마: 낮, 숲

(2 스테이지)



테마: 낮, 하늘, 버섯

(3 스테이지)



테마: 쿠파 성

3. 플레이어



1P : 마리오

2P : 루이지

3P : 와리오

플레이어의 상태:

- Normal mario: 게임의 디폴트 마리오 상태
- Star Mario: 10 초간 무적이 되며, 플레이어와 몬스터 충돌 시 몬스터가 한 방에 사망
- ☞ Special Mario: Normal mario 보다 상단으로 한칸 더 크며, 데미지를 받을 시 Normal mario 상태로 변함
 - Big Mario:
 - Flower Mario: SPACE 입력 시 불꽃이 플레이어가 바라보는 방향으로 떨어짐
 - 불꽃: 중력의 영향을 받으며, 바닥에 충돌 시 위로 튀어오름, 몬스터 및 블록에 충돌 시 사라짐
 - Tino Mario: SPACE 입력 시 불꽃이 플레이어가 바라보는 방향으로 일직선으로 날아감, Z 입력 시 플레이어가 바라보는 방향으로 물기
 - 티노불꽃: 중력의 영향을 받지 않으며, 몬스터 및 블록을 관통함
쿨타임: 5 초
 - 물기: 잠시동안 무적이 되며 일정 부분에 몬스터가 있다면 데미지를 입힘
쿨타임: 15 초

4. 아이템

버섯:

- 등장 시 우측으로 이동하며, 블록을 만날 시 방향을 전환함
- 효과: 플레이어가 상태가 Small mario 상태라면 Big mario로 전환

초록버섯:

- 효과: 플레이어의 목숨 + 1

꽃:

- 해당 아이템은 움직이지 않음
- 효과: Flower mario 상태로 전환

티노:

- 해당 아이템은 움직이지 않음
- 효과: Tino mario 상태로 전환

스타:

- 등장 시 위로 점프를 하며, 블록과 만날 시 방향을 전환하고, 바닥에 충돌할 시 튀어오름
- 효과: Star mario 상태로 전환

5. 몬스터

갈색 굼바:

- 플레이어에게 밟힐 시 찌그러져 죽음
- 좌우로 이동하며, 벽을 만날 시 방향을 전환

초록 굼바, 회색 굼바:

- 플레이어에게 밟힐 시 찌그러져 죽음
- 좌우로 이동하며, 벽을 만날 시 방향을 전환, 낭떠러지에서 떨어지지 않음

거북이 1, 2:

- 플레이어에게 밟힐 시 등껍질에 들어감
- 좌우로 이동하며, 벽을 만날 시 방향을 전환, 낭떠러지에서 떨어지지 않음
- 플레이어가 등껍질을 밟을 시 등껍질의 중앙을 기준으로 좌우로 나뉘며, 밟힌 방향으로 날아감, 등껍질에 맞을 시 어떤 오브젝트던 체력 -1

천사 거북이:

- 날개가 달려있어 공중에서 위아래로 천천히 떠있음
- 플레이어가 한 번 밟을 시 날개가 떨어져 일반 거북이 1로 변함
- 나머지 위 거북이1,2 와 동일

쿠과:

- 랜덤으로 좌우로 이동 및 점프
- 랜덤으로 불꽃을 발사 / 불꽃은 일직선으로 중력 및 블럭의 영향없이 좌측으로 날아감
- 플레이어가 불꽃에 충돌할 시 플레이어 데미지 1
- 체력: 100
- 쿠과 사망 시 우측 엔딩으로 가는 블럭 제거

6. 조작

- 방향키: 이동
- SPACE: 아이템 스킬
- Z: 아이템 스킬 2

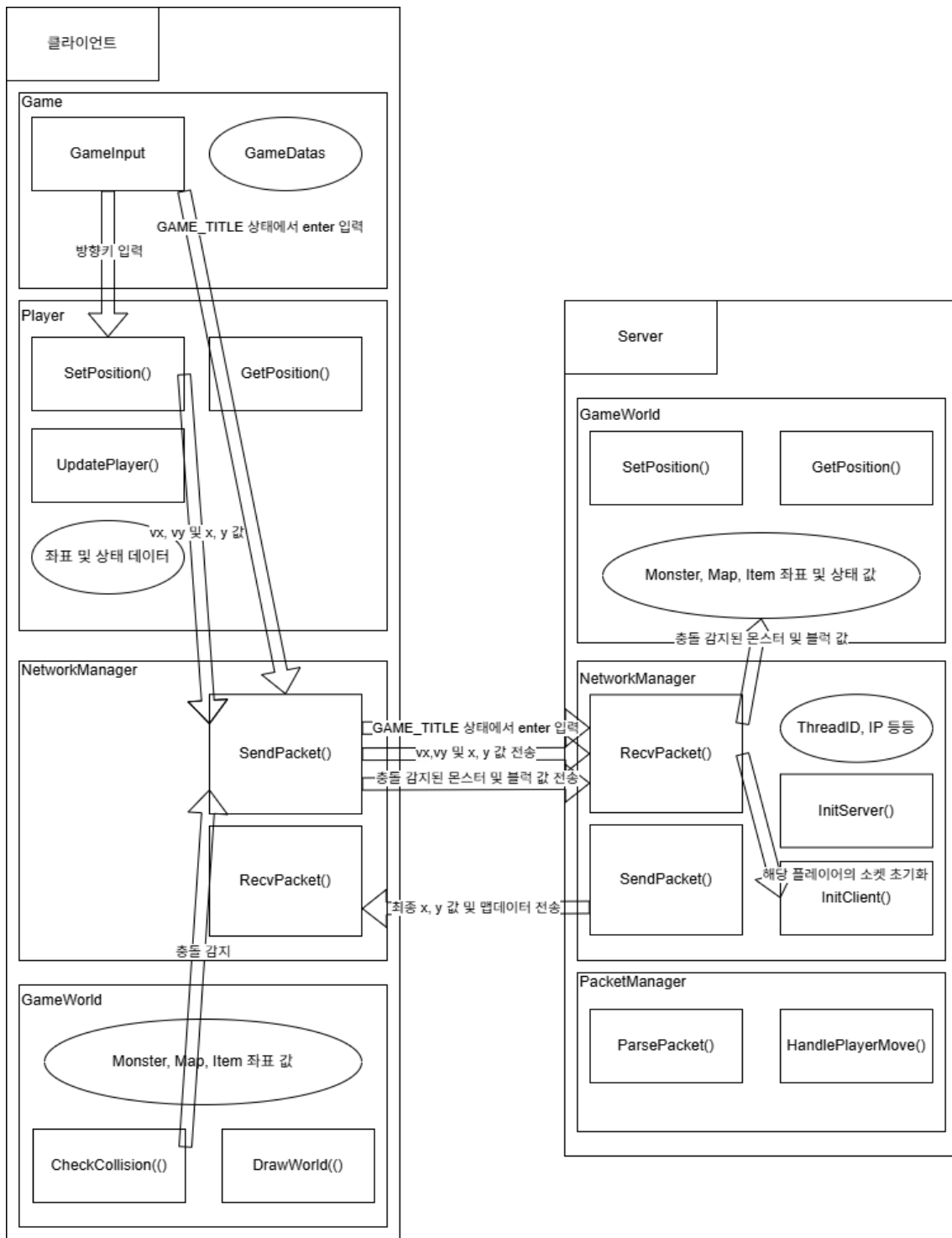
7. 상호작용

- 플레이어가 점프를 하여 플레이어 상단이 아이템블록에 충돌할 시 해당 블록의 아이템이 해당블록의 상단으로 생성
- 플레이어가 몬스터의 상단쪽을 밟을 시 몬스터의 체력 -1 (쿠파는 해당되지 않음)
- 플레이어가 버섯, 꽃, 티노, 스타 등의 아이템에 충돌할 시 해당 아이템의 효과를 얻음

8. 멀티플레이 추가 기능

- 각각의 플레이어가 각각의 카메라 좌표를 가지고있어, 각각의 화면이 다름
- 게임 클리어 조건 변경: 게임에 참여하고있는 플레이어 모두가 깃발에 충돌 시 게임 클리어
- 목숨 공용으로 변경
- 플레이어간 상호작용: 각각의 머리를 밟아 더 높게 뛰어 오를 수 있음

II. High-Level 디자인



Ⅲ. Low-Level 디자인

클라이언트	
GameWorld	void GameInput() void UpdateGame()
Player	void SetPosition() int GetPosition() void UpdatePlayer()
NetworkManager	bool SendPacket() int RecvPacket() bool InitClient()
PacketHandler	bool ParsePacket() bool HandlePlayerMove()
GameRender	void Render() void RenderTitle() void RenderGame()
서버	
NetworkManager	bool SendPacket() int RecvPacket() bool InitServer()
PacketHandler	bool ParsePacket() bool HandlePlayerMove()
GameWorld	void SetPosition() int GetPosition()

IV. 팀원 별 역할분담

팀원	주요 역할	세부 담당 내용
이용재	클라이언트 내 데이터 및 NetworkManager 작용 담당	<ul style="list-style-type: none">- 클라이언트에서 NetworkManager 초기화 (InitClient())- 패킷 송수신 (SendPacket(), RecvPacket())- 서버와의 연결 테스트 및 오류 처리- 클라이언트 쪽 네트워크 통신에 집중- 로컬 입력과 패킷 전송 연동 가능
이현석	서버/클라이언트 PacketHandler 담당	<ul style="list-style-type: none">- 패킷 파싱 (ParsePacket())- 플레이어 이동/상태 처리 (HandlePlayerMove())- 서버와 클라이언트 모두의 패킷 처리 로직을 통합- 패킷 포맷, 타입 정의와 검증 책임
황태규	서버 내 데이터 및 NetworkManager 작용 담당	<ul style="list-style-type: none">- 서버에서 NetworkManager 초기화 (InitServer()), 클라이언트 접속 관리- 서버 상태 관리 (GameWorld, Player 등)- 동기화 상태 전송, 충돌/게임 로직 처리 후 패킷 전송- 데이터 관리와 네트워크 전송을 동시에 담당

V. 개발 환경

개발 도구: Visual Studio 2022

API: Windows API (WinAPI)

통신 프로토콜: TCP/IP

버전 관리: Git, GitHub

VI. 개발 일정

1. 이용재 (클라이언트 네트워크)

일	월	화	수	목	금	토
	10/27	10/28	10/29	10/30	10/31	11/1
	추진 계획 서 제출		프로젝트 리팩토링 완성		프로젝트 구조 파악 클래스 정 의 확인 Network Manager 클래스 틀 작성	InitClient () 작성
11/2	11/3	11/4	11/5	11/6	11/7	11/8
테스트용 로컬 서버 코드 작성		연결 시도 시 재시도 로직 작성	로컬서버 accept() connect() 테스트		테스트용 패킷 정의	SendPacke t() 구현
11/9	11/10	11/11	11/12	11/13	11/14	11/15
SendPacke t() 구현		SendPacke t() 구현 간단한 패 킷 전송 로그 확인 및 테스 트	RecvPack et() 구현	RecvPack et() 구현		
11/16	11/17	11/18	11/19	11/20	11/21	11/22
		RecvPack et() 구현 테스트용 패킷 수신 확인	RecvPack et() 구현 서버에서 보내는 패 킷 수신 테스트	프로젝트 싱글플레 이 + 멀티플 레이스 단 계	Recv + Send 통합 테스 트 및 PacketHa ndler 연동	Recv + Send 통합 테스 트 및 ParsePack et(), Han dlePlayer Move() 호출 테스 트
11/23	11/24	11/25	11/26	11/27	11/28	11/29

서버로부터 수신받은 데이터 클라이언트 관리		서버로부터 수신받은 데이터 클라이언트 관리	여러 클라이언트 동시 접속, 위치 동기화 확인	패킷 손실 / 지연 테스트		최종 서버 / 클라이언트 연동 테스트
11/30	12/1	12/2	12/3	12/4	12/5	12/6
				조기 제출		
12/7	12/8	12/9	12/10	12/11		
				최종 제출		

2. 이현석 (패킷 핸들러)

일	월	화	수	목	금	토
	10/27	10/28	10/29	10/30	10/31	11/1
	추진계획 서 제출					
11/2	11/3	11/4	11/5	11/6	11/7	11/8
패킷 목록 초안작성			패킷 기본 구조 설계			패킷 구조 체 테스트 코 드
11/9	11/10	11/11	11/12	11/13	11/14	11/15
		PacketMa nager 클 래스 기본 구조				패킷 등록
11/16	11/17	11/18	11/19	11/20	11/21	11/22
			서버/클라 이언트 연동 준비			
11/23	11/24	11/25	11/26	11/27	11/28	11/29
서버와 클 라이언트 공통으로 사용되는 Packet 코 드 통합			다중 클라이언 트 대응			
11/30	12/1	12/2	12/3	12/4	12/5	12/6
예외 처리 및 안정화						서버/클라 이언트 통합 테스 트
12/7	12/8	12/9	12/10	12/11		

				최종 제출		
--	--	--	--	-------	--	--

3. 황태규 (서버 로직)

일	월	화	수	목	금	토
	10/27	10/28	10/29	10/30	10/31	11/1
	추진계획 서 제출				클라이언트 클래스 구조 파악 GameWorld() 클래스 작 성	SetPosition() GetPosition() 작성
11/2	11/3	11/4	11/5	11/6	11/7	11/8
몬스터 맵, 아이템 좌 표 및 상태 값 작성					Network Manager () 클래스 작성	InitServer () 작성
11/9	11/10	11/11	11/12	11/13	11/14	11/15
RecvPacket(), Send Packet() 작성 및 구현					initClient () 작성	서버 / 클라 이언트 연 동
11/16	11/17	11/18	11/19	11/20	11/21	11/22
멀티쓰레 드 생성 및 구현						다중 클라 이언트 접속 및 허용 테스트
11/23	11/24	11/25	11/26	11/27	11/28	11/29
예외 분석 및 처리						
11/30	12/1	12/2	12/3	12/4	12/5	12/6
						서버/클라 이언트 전 체 통합
12/7	12/8	12/9	12/10	12/11		

오류 분석				최종 제출		
-------	--	--	--	-------	--	--