멀티 플레이 디펜스 게임 구현

방종혁, 김명규, 장규현
한국공학대학교 게임공학과
airjong@tukorea.ac.kr, {derisan, jangdae0}@naver.com

요 약

본 연구에서는 2021년 모바일 게임 시장에서 급성장을 한 하이퍼 캐주얼 게임 장르를 PC 플랫폼으로 변경하여 설계한 사항과 구현 내용을 소개한다. 제안된 게임은 하이퍼 캐주얼의 특징인 단순함과 귀여움을 바탕으로, 이용자들 선호도에 따라 개인이 아닌 멀티 플레이 방식을 채택하여 PC 플랫폼의 장점을 살리며, 디펜스 형식의 게임 플레이를 통해 액션성을 추가하였다. 게임 서버는 Windows IOCP를 이용하여 여러 개의 방에서 동시에 게임을 진행할 수 있도록 하였다. 이에 관한 내용을 서술한다.

1. 서 론

모바일 게임 시장에서 깜짝 등장하여 그 특유의 분위기와 디자인으로 인기를 끌고 게임 순위 1위까지 차지했던 '어몽어스'[1]는 하이퍼 캐주얼 게임 장르라고 불리며 남녀노소 상관없이 게임 이용자들의 마음을 사로잡으며 선풍적인 인기를 끌었으며, 이는 PC 기반의 온라인 게임에서도 마찬가지였는데, 대표적인 예로 '폴 가이즈'[2]가 있다.

하이퍼 캐주얼 장르는 쉬운 접근성과 빠른 진행으로 한 게임에 짧으면 5분에서 길어도 30분 이내의 시간밖에 소요하지 않아 짧은 시간에 간단하게 즐길 수 있다는 장점이 있고, 단순하고 귀여운 디자인으로 호불호가 적어 이용자들에게 쉬운 접근성으로 플레이를 유도한다고 볼 수 있다.

본 게임은 윈도우 PC 플랫폼의 게임으로서, 기존 하이퍼 캐주얼 장르에 디펜스 게임의 플레이 방식을 기반으로 삼아 액션성을 추가하였으며, 멀티플레이 방식으로 팀원간의 협력을 유도하여 게임성을 향상 시켰다.

2. 개발 내용

2.1 게임 기획

제안되는 게임은 하이퍼 캐주얼 게임 장르를 기본으로 한다. 하지만 '캐주얼'이라는 것은 '최대한 단순하게 하여 진입 장벽을 낮춘 게임'들을 통틀어 말하는 것으로 플레이 방식이 딱 정해져 있지 않다.[3] 그렇기에 디펜스 장르를 기본으로 하였다. 또한 게임의 주 이용자 층인 10대, 20대가 여럿이서 하는 게임을 더 선호하는 경향(각각 80.9%, 66.2%)[4]이 있기 때문에 협력 멀티 플레이 게임으로 방향성을 잡았다.

방에 접속하여 최소 1명에서 최대 3명의 플레이어가 각자 딜러, 서포터, 힐러의 역할 중 하나를 중복이 가능하도록 선택한 후 팀을 이루어 게임을 시작할 수 있다. 팀은 거점을 지

키면서 무사히 목적지까지 이동하는 것이 승리 조건이다. 거점 자체가 움직이도록 설정하여 플레이어들은 고정되어 있는 것이 아니라 계속 이동하면서 행동을 취해야 한다. 게임이 시작되면 플레이어는 맵의 시작점에서 나와 횡 방향으로 계속해서 이동하는 거점을 따라가며, 거점의 이동경로 상에 있는 장애물, 플레이어를 방해하고 거점을 공격하는 몬스터를 제거해야 하며, 중간에 등장하는 대량의 몬스터가 발생하는 특수 이벤트를 돌파하고, 목적지 부근에 위치한 보스 몬스터를 쓰러트리면 게임에서 승리한다.

디펜스 게임의 경우 플레이한 팀들의 쓰러트린 몬스터나 클리어한 시간을 기준으로 점수를 측정한다. 하지만 이 게임의 특성상 움직이는 거점의 이동속도와 맵의 길이는 고정이기에 시간으로 점수를 나눌 수 없으며, 몬스터 처치의 경우도 크게 차이가 나지 않는 것이 단점이다.

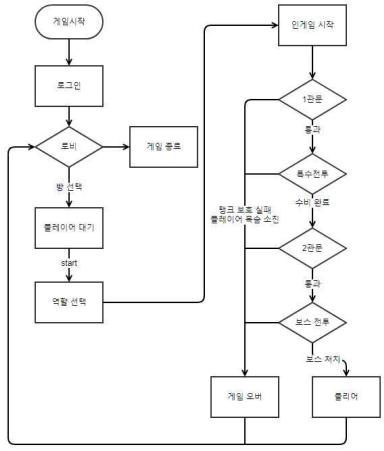
이 점을 보완하기 위하여, 점수를 획득해 오는 아군 NPC를 설정하여, 보호해야 할 대상으로 역할을 부여하여, 플레이어들은 좀 더 다양한 요소를 플레이하면서 판단해야 하고, 몬스터들이 NPC를 공격하여 고득점을 노릴 수 없게 하는 변수 요소로서 작동하도록 하여 게임의 재미를 더하였다.

또한 게임을 반복적으로 플레이할 경우 금방 지루해 질 수 있기 때문에 역할군을 나누었으며, 중복으로 선택을 할 수 있도록 하여 플레이 할 때 마다 다양한 전술과 조합을 만들 수 있도록 하였다.

디자인적 요소로는 게임에 하나의 공통된 주제를 주기 위하여 몸 속의 세포가 되어서 싸운다는 설정을 기반으로 하였고, 하이퍼 캐주얼 장르의 특징을 살리도록 모든 오브젝트들의 디자인은 단순하고 사각형, 구체 등 기본 도형에 가깝게 디자인 하였다.

2.2 게임의 전체 흐름

게임의 전체적인 흐름은 다음과 같다.



[그림 1] 플레이 흐름도

2.3. 게임 진행

2.3.1 준비단계

플레이어는 게임 메인 화면에서 닉네임을 입력 후 'LOGIN' 버튼을 클릭하여 자신의 캐릭터를 생성한다.

[그림 2] 로그인

3개의 대기실 중 비어있는 방을 클릭하여 입장한다. (비어있는 방은 'EMPTY'로 표시되

며 입장 시 가장 처음으로 입장한 플레이어가 ‘방장’으로 자동 설정된다. 다른 플레이어가 대기 중인 방은 ‘WAITING’으로 표시된다. 방에 입장한 플레이어가 3명일 경우 더 이상 입장할 수 없다. 이미 게임이 진행 중인 방은 ‘PLAYING’으로 표시되며 클릭하여도 입장할 수 없음)

[그림 3] 로비

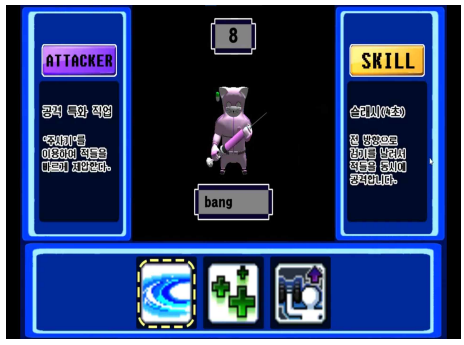
방에 입장한 순서대로 자동으로 캐릭터가 배정된다. (해당 방에서 나가고 싶은 경우 ‘BACK’ 버튼 클릭 시 로비로 돌아감)

[그림 4] 게임 대기실

2.3.2. 게임 플레이

대기 중인 상태에서 방장이 ‘START’ 버튼을 눌러 게임을 시작한다.

게임이 시작되면 10초의 시간이 주어지며 각 플레이어는 스킬 아이콘을 클릭하여 자신이 원하는 역할을 선택한다.(중복 선택이 가능하며 시간 내에서 제한 없이 변경이 가능)



[그림 5] 역할 선택

[그림 5]는 역할 선택화면으로 하단의 스킬 아이콘을 통해 역할을 선택할 수 있으며 좌측에선 선택한 역할의 특성, 우측에선 선택한 역할의 스킬 정보를 확인할 수 있다.

10초 후 게임 맵으로 이동하여 본격적으로 게임이 시작되며 플레이어는 조우하는 몬스터를 ‘기본 공격’과 ‘스킬’을 활용하여 격파하고 경로를 따라 이동하는 거점(탱크)을 방해하는 장애물(지방)을 제거하며 탱크를 목적지까지 무사히 옮기는 것을 목표로 한다.

[그림 6] 지방 제거

[그림 6]은 플레이 도중 지방을 제거하는 장면으로 좌측 상단에서 획득한 점수와 탱크의 체력, 하단에서 각 플레이어의 체력과 닉네임, 플레이 중인 본인의 캐릭터의 스킬과 스킬 재사용 대기시간을 확인할 수 있다.

게임을 진행하는 도중 2번의 이벤트가 발생

한다. 첫 번째 이벤트는 특수 전투로 세균 벽에 가로막힌 채 사방에서 적들이 출몰한다. 탱크가 세균 벽을 파괴할 때까지 적들의 공격으로부터 탱크를 지켜내야 한다.

[그림 7] 특수 전투

두 번째 이벤트는 맵의 마지막에 등장하는 보스 전투이다. 보스는 2가지의 일반 공격 패턴과 필살기를 갖는다. 공격 재사용 대기시간에 맞춰 일반 공격 중 한 가지 공격을 랜덤으로 사용한다. 보스의 체력이 일정 수준 이하로 데미지를 입으면 보스의 외형이 변화하며 필살기를 사용한다.

[그림 8] 보스 전투

2.3.3 게임 종료

앞에서 서술한 게임 플레이에 따라 탱크를 목적지까지 무사히 운반한 경우 게임을 승리한 것이 되며 플레이하면서 얻은 점수와 플레이 시간이 표시된다.

[그림 9] 클리어 화면

반대로 탱크가 목적지까지 운반되기 전에 파괴되거나, 플레이어가 전멸한 경우 게임 패배로 간주되며 마찬가지로 점수와 플레이 시간이 표시된다.

'BACK' 버튼을 누르면 다시 게임 방 선택화면으로 돌아간다.

2.4 클라이언트

2.4.1 클라이언트 구조

클라이언트는 하나의 스레드로 구성되어 있다. 이 스레드는 사용자 입력 처리, 패킷 처리, 게임 업데이트, 렌더링 순으로 작업을 실행한다. 게임 루프마다 논블로킹 소켓을 이용해 서버로부터 패킷을 받아 처리한다. 또한 여러 개의 씬(Scene)을 관리하기 위해 간단한 유한 상태 기계 모델을 사용했다.

2.4.2 클라이언트 구현

가장 먼저 수행되는 작업은 사용자의 입력을 처리하는 것이다. 이 단계에선 키보드 또는 마우스 입력 이벤트가 발생했는지를 조사하여 서버에 패킷을 보낸다. 클라이언트가 서버에게 보내는 패킷 대부분을 이동 요청 패킷이 차지하고 있는데, 효율을 위해 키 입력의 변화가 있을 때만 이동 요청 패킷을 보내도록 했다.

패킷 처리 단계에선 서버에서 보낸 패킷을 처리한다. TCP 연결 특성상 온전하지 않은 패킷이 올 가능성도 있기에 Ring buffer를 두어 패킷 조립을 진행한다.

게임 오브젝트 간의 충돌 감지는 AABB(Axis Aligned Bounding Box) 방식을 이용했다. 적, 벽 등 통과할 수 없는 오브젝트와 플레이어 캐릭터 간의 충돌을 감지하면, 충돌 박스가 겹친 만큼 캐릭터를 이동 방향의 반대 방향으로 밀어내 추가적인 충돌이 일어나지 않게끔 했다.

캐릭터 닉네임, 스킬 재사용 대기시간, 점수 등을 폰트를 사용해 표시했으며 폰트 렌더링은 Direct2D[5] API를 이용해 구현했다. 이외의 2D 스프라이트는 직교 투영 행렬을 사용해 화면 위에 그려내었다.

대부분의 3D 모델은 각자 여러 개의 애니메이션을 가지고 있다. 애니메이션을 부드럽게 전환하기 위해 애니메이션 블렌딩을 구현했다. 하나의 캐릭터가 역할에 따라 여러 장비를 착용할 수 있도록 3D 모델의 본에 다른 오브젝트를 부착하는 기능 또한 구현했다.

게임은 기본적으로 Top View로 진행되지만 중간 전투, 보스 전투 같은 이벤트 상황에서 시점을 바꾸거나 카메라를 흔들어서 몰입감을 높였다.

사운드 재생을 위해 FMOD 라이브러리[6]를 사용했다. 반복 재생, 볼륨 조절 기능을 추가해 배경 음악, 효과음을 재생할 때 이용했다.

2.5 서버

2.5.1 서버 구조

서버는 Windows에서 제공되는 IOCP(Input Output Completion Port) API를 이용해 구현했다. 패킷 송수신을 담당하는 다수의 Worker 스레드와 하나의 로직 스레드로 이루어진 멀티스레드 서버이며 동시에 3개의 방, 총 9명의 사용자에게 게임 플레이를 제공할 수 있다.

2.5.2 서버 구현

Worker 스레드는 클라이언트가 보낸 패킷을 받거나 로직 스레드가 만들어 낸 패킷을 보낸다. 이 과정에서 Worker 스레드와 로직 스레드 간 Data Race가 발생해 성능 저하의 우려가 있었다. 따라서 두 개의 패킷 Queue를 두었다. 로직 스레드는 Front Queue에서 Data Race가 발생할 우려 없이 패킷을 Pop하여 처리한다. 그동안 Worker 스레드들은 나머지 Back Queue에 클라이언트로부터 받은 패킷을 저장한다. 로직 스레드가 Queue에 담긴 패킷을 모두 처리하면, 신호를 보내 Front Queue와 Back Queue를 서로 바꾼다. 이것으로 Worker 스레드와 로직 스레드의 Data Race를 줄일 수 있었다.

Overlapped 확장 구조체의 빈번한 메모리 할당 및 해제의 오버헤드를 줄이기 위해 멀티스레드 환경에 최적화된 메모리 할당기인 Google의 tcmalloc[7]을 사용했다.

C++ 언어가 기본적으로 제공하는 Lock보다 Windows 환경에서 더 성능이 좋은 SRWLock(Slim Reader/Writer Lock)[8]을 사용했다. 서버 설계상 길게 Lock을 획득해야만 하는 부분은 없기에 Spin을 이용하는 SRWLock을 사용하는 게 성능에 좋다고 판단했다.

유저가 로그인에 성공하면 서버는 방 목록을 전송한다. 게임이 진행되고 있는 방으로의 입장 요청은 거부하며 다른 유저가 기다리고

있거나 비어있는 방에만 입장할 수 있도록 했다.

방에 입장하면 서버는 새로 입장한 유저에게 기존에 방에 있던 유저들의 닉네임, 캐릭터 정보를 보낸다. 기존 유저들에게 또한 마찬가지로 새 유저의 정보를 보낸다. 방장으로 설정된 유저가 시작을 요청하면 서버는 게임 스테이지를 초기화하는 작업을 수행한다.

서버는 A* 알고리즘을 사용해 길 찾기를 수행해야 하는 모든 오브젝트의 경로를 설정한다. A* 알고리즘을 통해 각 오브젝트는 목표에 도달하기 위한 최단 경로로 움직인다. 최상 우선 탐색(Best-first search) 또한 고려했으나 장애물의 유무에 따라 비효율적인 탐색을 보이기에 제외하였다.

2.6 리소스

2.6.1 모델링

게임 캐릭터는 3ds Max 2021을 이용하여 제작하였다. 플레이어 캐릭터는 3천개 전후의 폴리곤으로 제작하였으며, 몬스터의 경우에는 인간형 2천개, 동물형 3천개 전후, 보스 몬스터의 경우에는 약 8천개의 폴리곤을 사용하였다. NPC의 경우에는 약 1천개의 폴리곤이 사용되었다.

모든 오브젝트들은 1장의 텍스처만 사용하여 로딩 속도를 줄였다. 텍스처는 Adobe PhotoShop 2021로 제작하였으며 2048*2048의 크기를 가진다.

캐릭터 애니메이션과 보스 몬스터의 애니메이션의 경우 바이패드에 본을 붙여서 사용하였다.

2.6.2 파티클

따로 파티클 툴을 이용하여 제작되지 않았으며, 모델링과 마찬가지로 3ds Max 2021을 이용하여 제작하였다. 프로그램 자체 기능인 파티클 시스템을 이용하여 오브젝트 애니메이션으로 추출한 후 게임에 적용시켰다.

2.6.3 사운드

게임 내에서 사용된 효과음은 무료 효과음 배포 사이트 'FreeSound'[9]에서 파일을 다운로드 받고, 배경음의 경우에는 상업적으로도 이용 가능한 배경음 무료 배포 유튜브 'HeatleyBros'[10]에서 음원을 추출한 후 게임에 적용하였다.

3. 결 론

게임 '하트비트'는 호불호가 적은 디자인과 단순한 조작으로 하이퍼 캐주얼 게임의 특징을 살리며 디펜스 게임의 플레이 방식을 접목시켜 액션성을 부여하고 멀티 플레이를 지원하여 게임성을 높였다.

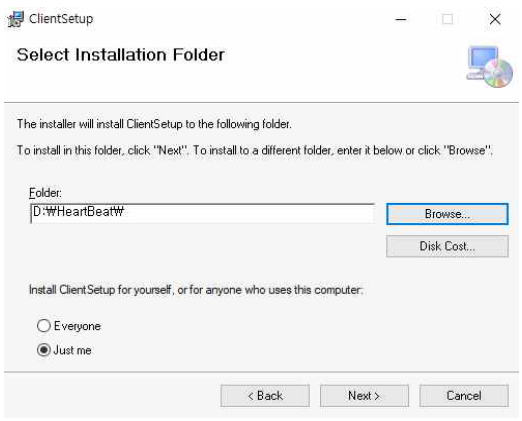
클라이언트에선 셰이더를 사용한 그림자를 구현하지 못했다. 향후 Shadow Mapping을 학습하여 자연스러운 그림자를 그릴 수 있도록 할 예정이다.

서버의 로직 스레드는 단 하나이기에 여러 개의 방에서 게임이 동시에 진행되면 효율적으로 동작하지 못할 수 있다. 방마다 로직 스레드를 두어 로직을 병렬적으로 처리하도록 보완할 필요가 있다. 또한 멀티스레드 환경에 최적화된 tcmalloc을 사용했다고 하지만 Overlapped 확장 구조체 때문에 메모리 할당과 해제가 계속 일어나고 있다. Microsoft PPL(Parallel Patterns Library)[11]의 lock-free queue를 이용한 Overlapped 확장 구조체 풀을 만들고 서버 초기화 시 사용할

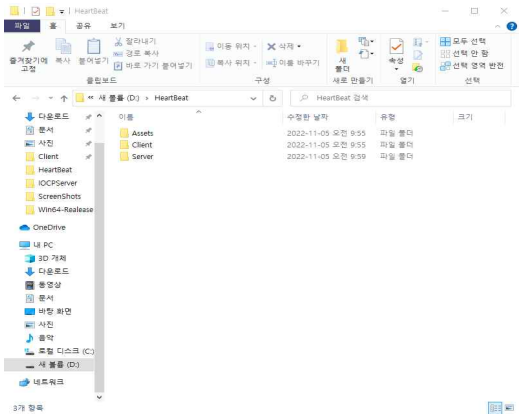
모든 메모리를 할당해 성능을 개선할 계획이다.

4. 부 록

4.1 게임 설치 및 실행



[그림 10] 클라이언트 설치 위치 지정



[그림 11] 설치 완료 후 폴더 모습

게임 설치를 위해 두 개의 파일을 실행시켜야 한다. ClientSetup.msi 파일로 클라이언트를, ServerSetup.msi 파일로 서버를 설치할 수 있다. [그림 10]은 msi 파일 실행 후 설치 경로를 지정하는 단계를 보여주고 있다. 설치 시 클라이언트와 서버를 반드시 같은 폴더에 설치하도록 주의한다. 폴더명이나 위치는 자유롭게 해도 상관없다. [그림 11]은 설치 완료 후 폴더의 모습이다. 클라이언트가 서버에 접속하기 위해선 서버 PC의 IP를 입력할 필요가 있다. Client 폴더 내부에 Client.xml 파일을 열어 <IP> 부분의 값을 서버가 실행되고 있는 PC의 IP로 고쳐주도록 한다.

4.2 문제 해결을 위한 연락처

① 게임 설치 및 실행
derisan@naver.com

② 기획 및 그래픽
airjong@tukorea.ac.kr

REFERENCES

[1] Among Us home, innersloth.com/games/among-us, Innersloth

[2] FallGuys home, fallguys.com/ko, Mediatonic

[3] 박수영, 한국 게임의 역사, 9장 1-1, 북코리아, 2012.12

[4] KOCCA, 2021 게임이용자 실태조사, P.57, 2021.09

[5] “Direct2D” Microsoft Learn, 2022년 5월 26일 수정, 2022년 11월 16일 접속, learn.microsoft.com/en-us/windows/win32/direct2d/direct2d-portal

[6] FMOD, fmod.com, Firelight Technologies

[7] tcmalloc, github.com/google/tcmalloc, Google

[8] “SRWLock 빠른 성능의 비결”
megayuchi 블로그,
2017년 6월 25일 수정,
2022년 11월 17일 접속,
megayuchi.com/2017/06/25/srwlock-%EB%B9%A0%EB%A5%B8-%EC%84%B1%EB%8A%A5%EC%9D%98-%EB%B9%84%EA%B2%B0
[9] FreeSound, freesound.org,
Music Technology Group
[10] HeatleyBros,
youtube.com/c/HeatleyBros
[11] “Parallel Patterns Library”
Microsoft Learn,
2021년 8월 4일 수정,
2022년 11월 17일 접속,
learn.microsoft.com/en-us/cpp/parallel/concrt/parallel-patterns-library-ppl?view=msvc-170