캡스톤디자인 개발 보고서

<방탈출W.CWNU>

* 메타버스 방 탈출 게임 -

|  |  |
| --- | --- |
| 지도 교수 | 강동현 |
| 학 과 | 국어국문학과, 컴퓨터 공학과 |
| 조 원 | 우지연, 김지현 |
| 제 출 일 | 2022.06.21 |



**목차**

[1.서론 1](#_Toc106726547)

[1.1 개발 동기 2](#_Toc106726548)

[1.2 기대 효과 2](#_Toc106726549)

[1.3 개발 환경 및 기기 2](#_Toc106726550)

[2.배경지식 1](#_Toc106726551)

[2.1 Metaverse 1](#_Toc106726552)

[2.2 Unity 1](#_Toc106726553)

[2.3 C# 1](#_Toc106726554)

[2.4 Photon 1](#_Toc106726555)

[3.유사서비스 2](#_Toc106726556)

[3.1 The Room VR: A Dark Matter 2](#_Toc106726557)

[3.2 분석 2](#_Toc106726558)

[4.개발내용 3](#_Toc106726559)

[4.1 서버 3](#_Toc106726560)

[4.3 플레이어 5](#_Toc106726561)

[4.3.1 플레이어 조작 5](#_Toc106726562)

[4.3.2 인벤토리 5](#_Toc106726563)

[4.3.3 보이스 챗 6](#_Toc106726564)

[4.4 오브젝트 7](#_Toc106726565)

[4.4.1 퀴즈 오브젝트 7](#_Toc106726566)

[4.4.2 타이머 기능 7](#_Toc106726567)

[5. 개발 Issue 9](#_Toc106726568)

[5.1 주제 변경 9](#_Toc106726569)

[5.2 유니티 사용 미숙 9](#_Toc106726570)

[6.개발 일정 10](#_Toc106726571)

[7. 참고 문헌 11](#_Toc106726572)

**[그림 목차]**

[[그림 1] The Room VR 화면 2](#_Toc106726664)

[[그림 2] PhotonNetwork의 플레이어 및 서버 동기화 부분 3](#_Toc106726665)

[[그림 3] 첫 시작화면 4](#_Toc106726666)

[[그림 4] 서버 접속 후 화면 4](#_Toc106726667)

[[그림 5] 방 리스트 갱신 화면 4](#_Toc106726668)

[[그림 6] 방 입장 화면 4](#_Toc106726669)

[[그림 7] 사용자 정보 갱신 화면 4](#_Toc106726670)

[[그림 8] 플레이어 조작 스크립트 일부 5](#_Toc106726671)

[[그림 9] 아이템 동기화 화면 5](#_Toc106726672)

[[그림 10] 아이템 동기화 스크립트 일부 6](#_Toc106726673)

[[그림 11] Voice Chat 함수 일부분 발췌 6](#_Toc106726674)

[[그림 12] Stage1 퀴즈 7](#_Toc106726675)

[[그림 13] Stage 1 퀴즈 결과 7](#_Toc106726676)

[[그림 14] Stage 2 퀴즈 기능 7](#_Toc106726677)

[[그림 15] Timer 동기화 스크립트 일부 발췌 8](#_Toc106726678)

[[그림 16] 게임 종료 시 타이머를 띄워주는 화면 8](#_Toc106726679)

# 1.서론

서론에서는 해당 프로그램 개발 동기에 대해 서술한다.

## 1.1 개발 동기

오프라인에서 방탈출 게임을 진행하는 경우 세가지의 단점이 존재한다. 첫번째, 게임에 한 번 참여할 때마다 약 2만원의 입장료가 발생한다. 두번째, 하나의 테마는 한 팀만 이용 가능하며, 테마의 분배가 고르지 않다는 공간적 제약성이 존재한다. 마지막 시간적 제약성이다. 방탈출의 게임 특성 상 제한 시간이 존재하며, 해당 시간이 지나면 게임이 남아있더라도 진행할 수 없다. 이러한 단점들을 보완하고 개선하고자 메타버스 플랫폼을 통해 방탈출 게임을 만들기로 하였다. 따라서, 시공간적 제약없이 다인원이 같은 테마의 방탈출 게임을 할 수 있도록 한다.

## 1.2 기대 효과

사용자들이 동시에 동일한 테마 방탈출 게임을 할 수 있으며, 메타버스 플랫폼을 활용하여 오프라인에서 진행하는 게임과 유사한 환경을 만들어 즐길 수 있다. 더불어, 게임에 국한되는 것이 아닌 사용자 간의 의사소통이 가능하게 하여 SNS의 기능을 더할 수 있다.

## 1.3 개발 환경 및 기기

하드웨어는 [표1]의 DELL 인스피론15 7501-WH06KR, [표 2]의 LG그램 15 Z90Q-GA56K 두 개의 개발 기기를 이용하였다. 소프트웨어는 Rider, Unity 2019.4.36 ver, C#을 사용하였다.

[표 1] Dell 인스피론 사양

|  |  |  |  |
| --- | --- | --- | --- |
| 제조사 | Dell | 메모리 | 16GB |
| CPU | 코어i7-10세대 | 저장 용량 | 512GB |
| OS | Window10 | GPU | GTX1650 |

[표 2]LG gram 사양

|  |  |  |  |
| --- | --- | --- | --- |
| 제조사 | LG | 메모리 | 16GB |
| CPU | 코어i5-11세대 | 저장 용량 | 256GB |
| OS | Window10 | GPU | Intel Iris Xe |

# 2.배경지식

## 2.1 Metaverse

메타버스(metaverse) 또는 확장 가상 세계는 가상, 초월을 의미하는 '메타'(meta)와 세계, 우주를 의미하는 '유니버스'(universe)를 합성한 신조어다. 가상현실, 증강현실의 상위 개념으로서 현실을 디지털 기반의 가상 세계로 확장해 가상 공간에서 모든 활동을 할 수 있게 만드는 시스템이다.

## 2.2 Unity

유니티(Unity)는 3D 및 2D 비디오 게임의 개발 환경을 제공하는 게임 엔진이자, 3D 애니메이션과 건축 시각화, 가상현실(VR) 등 인터랙티브 콘텐츠 제작을 위한 통합 제작 도구이다. 게임 개발에 사용하는 스크립트 언어는 C#와 자바스크립트를 지원한다.

## 2.3 C#

마이크로소프트에서 개발한 객체 지향 프로그래밍 언어로, 닷넷 프레임워크의 한 부분으로 만들어졌으며 나중에 ECMA (ECMA-334)와 ISO (ISO/IEC/23270)의 표준으로 자리잡았다. C++와 자바의 문법과 비슷한 문법을 가지고 있다. C#의 기본 자료형은 닷넷의 객체 모델을 따르고 있고, 런타임 차원에서 쓰레기 수집(garbage collection)이 되며 또한 클래스, 인터페이스, 위임, 예외와 같이 객체 지향 언어로서 가져야 할 모든 요소들이 포함되어 있다. C#에는 전역 변수 및 전역 함수가 존재하지 않으며, 클래스 안에 선언되어야 한다. C#의 bool은 오직 true와 false의 논리값만을 가질 수 있으며, 상수 또는 정수형 변수에서 암시적으로 변환이 불가능하다. 직접 대입을 위해서는 변환 명령을 이용해야 한다. 반면 C++의 bool은 정수값을 대입할 수 있다. 또한 C#에서는 if나 while문 등의 비교문에서 이용하는 값도 bool 형태로 제한되는 반면, C++에서는 상수 또는 변수를 이용하여 '0이 아닌 값' 또는 '0'의 여부로 비교할 수 있다. C#에서는 static 키워드를 오직 한 번만 초기화를 수행한다는 의미로 이용할 수 없다.

## 2.4 Photon

실시간 클라우드, 멀티 플레이어, 매치메이킹 API 서버 간 통신이 가능한 엔진이다.

Photon PUN은 Unity 3D를 이용하는 모든 타입의 룸 기반 멀티플레이어 게임을 위한 백엔드를 지니고 있다. PUN을 통해 개발된 게임은 Photon Cloud를 통해 자동으로 스케일링 된다.

랜덤, 또는 검색 조건으로 플레이어의 매칭이 가능하며, UDP, TCP, HTTP 또는 Websocket 등, 어떤 환경일지라도 펀치스루 문제가 발생하지 않는다.

# 3.유사서비스

## 3.1 The Room VR: A Dark Matter

[그림 1]은 앞으로 개발할 프로그램과 유사한 특징을 지닌 VR방탈출 게임이다.

[그림 1] The Room VR 화면

텍스트이(가) 표시된 사진

자동 생성된 설명[[1]](#footnote-1)

이는 VR기기 전용 게임이며, 약 3만원대의 가격으로 구성 되어있다. 게임을 시작할 때, 서있는 상태로 진행할 지, 앉은 상태로 진행할 지 선택할 수 있다. 전용 VR기기를 착용하고 양손에 조이스틱을 쥐면 게임이 시작된다. 해당 게임은 4개의 스테이지로 구성되어 있으며, 각 스테이지가 다른 테마로 이루어져 있다. 게임의 진행은 정해진 위치에서만 이동 가능하다. 해당 위치에서 상호작용하며 문제를 풀거나 힌트를 얻어서 진행할 수 있다.

## 3.2 분석

위 게임은 VR기기를 사용하여 실감나게 방탈출을 할 수 있다는 장점이 있다. 스테이지 별 확장성으로 게임에 지루함을 덜어준다. 그러나, 싱글 플레이로 진행되고, 그에 따라 메타버스의 환경을 유지하지 못한다는 아쉬움이 존재했다. 따라서, 이번 프로그램에서는 인게임 보이스 채팅 및 대기실 채팅을 할 수 있으며, 멀티 플레이로 게임을 할 수 있도록 제작하였다.

# 4.개발내용

## 4.1 서버

socket 서버중 하나인 포톤 서버의 기능을 사용할 수 있는 애플리케이션인 포톤클라우드를 사용하였다. 포톤외에도 프라우드넷, cloud bread와 같은 클라우드가 존재하지만, 포톤서버가 c++로 설계되어 타 게임서버보다 정교하고 빠르다는 이점이 있어 포톤으로 진행하게 되었다.

Photon과의 서버 연결 및 동기화를 진행하기 위해 Network Manager라는 스크립트를 생성하였다. 이를 통해 구현한 기능은 서버 연결, 게임 내 여러 개의 방 생성, 실시간 사용자 정보 갱신, 채팅 4가지이다.

먼저, 플레이어와 서버를 연결하기 위해 PhotonNetwork와 NetworkManager를 작성하였다. [그림 2]은 PhotonNetwork 중 플레이어가 게임에 접속하게 되면, 즉시 서버(PhotonCloud)와 연결되도록 하는 함수이다.

[그림 2] PhotonNetwork의 플레이어 및 서버 동기화 부분

텍스트이(가) 표시된 사진

자동 생성된 설명

[그림 3]과 [그림 4]는 서버에 접속할 때의 화면을 보여준다. 해당 화면에서는 유저가 원하는 캐릭터와 닉네임을 설정할 수 있으며, 게임을 하기 위한 룸을 생성할 수 있도록 구현하였다.

|  |  |
| --- | --- |
| [그림 3] 첫 시작화면 | [그림 4] 서버 접속 후 화면  텍스트이(가) 표시된 사진  자동 생성된 설명 |

한명의 플레이어가 방을 생성하면 [그림 5]와 같이 접속된 모든 플레이어의 화면 오른쪽에 방 리스트가 갱신된다.

[그림 5] 방 리스트 갱신 화면

텍스트이(가) 표시된 사진

자동 생성된 설명

[그림 6]과 [그림 7]은 방에 참여한 뒤의 화면을 보여준다. 방에 다른 사용자 참여시 방의 인원, 알림 등 방의 정보를 갱신해준다. 또한 사용자들의 상호작용을 위하여 채팅기능을 추가하였다. 본 화면에서는 4명이 그룹을 이뤄 게임을 진행하기 때문에 방의 최대 인원도 4명으로 제한되었다.

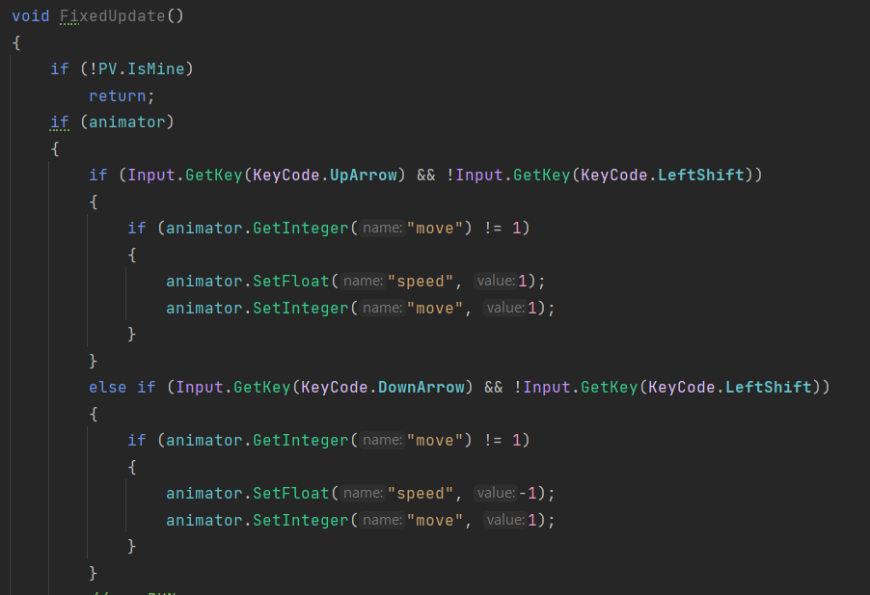
|  |  |
| --- | --- |
| [그림 6] 방 입장 화면  텍스트이(가) 표시된 사진  자동 생성된 설명 | [그림 7] 사용자 정보 갱신 화면  텍스트이(가) 표시된 사진  자동 생성된 설명 |

## 4.3 플레이어

### 4.3.1 플레이어 조작

[그림 8]은 방향키를 이용하여 플레이어를 조작할 수 있게 하기위한 스크립트의 일부분이다. PhotonView를 이용하여 각각의 플레이어를 독립적으로 구분하며 특정 키보드 입력 시 플레이어가 이동하거나 게임내 오브젝트들과 상호작용하도록 구현하였다.

[그림 8] 플레이어 조작 스크립트 일부



### 4.3.2 인벤토리

[그림 9]은 플레이어의 인벤토리 구현 화면이다. 플레이어들 간의 획득한 아이템을 공유할 수 있게 하기 위해 구현하였다. [그림 9]과 같이 아이템을 습득하면, 모든 플레이어들의 인벤토리에 업데이트 되도록 하였다. 플레이어가 소모하는 아이템은 1회성을 띄고 있어 사용 시 인벤토리에서 삭제된다. [그림 10]은 플레이어간 획득한 아이템을 동기화하기 위한 함수이며, 해당 코드는 아이템을 획득 시 인벤토리에 아이템이 추가되도록 구현되었다.

[그림 9] 아이템 동기화 화면



[그림 10] 아이템 동기화 스크립트 일부

텍스트이(가) 표시된 사진

자동 생성된 설명

### 4.3.3 Voice Chat

메타버스 플랫폼의 특성을 활용하기 위해 구현한 기능이다. 인게임에서 플레이어들은 서로의 목소리를 들으며 함께 문제를 풀어나갈 수 있다. [그림 11]은 해당 기능을 구현하기 위한 함수 일부분을 발췌하였다.

[그림 11] Voice Chat 함수 일부분 발췌

텍스트이(가) 표시된 사진

자동 생성된 설명

## 4.4 오브젝트

게임 내 퀴즈와 오브젝트들의 기능을 서술하였다. 먼저, 게임 내의 퀴즈 오브젝트부터 살펴보도록 한다.

### 4.4.1 퀴즈 오브젝트

해당 기능은 스테이지가 진행될 수 있는 힌트 혹은 아이템을 얻기 위해 구현한 기능이다. [그림 12]는 stage 1에서 등장하는 퀴즈 오브젝트이다. 해당 퀴즈를 해결하게 되면 [그림 13]과 같이 다음 퀴즈로 넘어가는 힌트가 등장하게 된다.

|  |  |
| --- | --- |
| [그림 12] Stage1 퀴즈  텍스트, 칠판이(가) 표시된 사진  자동 생성된 설명 | [그림 13] Stage 1 퀴즈 결과 |

[그림 14]는 Stage 2]의 퀴즈 기능이다. Stage 내에서 힌트를 찾고, 해당되는 위치에서 정답을 맞출 경우, 다음 Stage로 넘어가는 열쇠가 나타나도록 구현하였다.

[그림 14] Stage 2 퀴즈 기능

텍스트이(가) 표시된 사진

자동 생성된 설명

### 4.4.2 타이머 기능

플레이어들이 게임을 진행하는 시간을 보여주기 위한 기능이다. [그림 15]는 Timer 스크립트 중 일부를 발췌한 것으로 해당 타이머 기능은 모든 플레이어간 동일한 시간을 공유하도록 설정하였다.

[그림 15] Timer 동기화 스크립트 일부 발췌

텍스트이(가) 표시된 사진

자동 생성된 설명

또한, 게임이 끝난 경우, [그림 16]과 같이 소요 시간을 모든 플레이어의 화면에 띄워주도록 하였다.

[그림 16] 게임 종료 시 타이머를 띄워주는 화면



# 5. 개발 Issue

해당 프로젝트를 개발하는 도중 생겼던 문제점들에 대해 나열하였다.

## 5.1 주제 변경

가장 큰 문제점은 개발 도중 주제를 변경한 것이다.

티볼리는 기존 메타버스 오픈소스 SW 분석 및 최적화를 위해 선정되었던 플랫폼이다. 해당 플랫폼을 통해 PC에서 빌드하기 위해 AWS 구입과 Linux환경을 구축하였고, 이 과정에서 사양에 관한 에러가 발생하여 Docker를 통해 빌드를 해야 했다. 해당 문제점을 해결하기까지 상당한 시간이 소요되었다. 이후, 빌드된 티볼리를 통해 창원대학교 내의 중고장터를 개발하고 있었으나, 2월 중순경 해당 플랫폼이 서비스를 종료함에 따라 메타버스와 관련된 새로운 주제를 찾아야 했다.

## 5.2 유니티 사용 미숙

새로운 주제로 메타버스 플랫폼 내의 방탈출 게임을 만들기로 결정을 하게 되었다. 그러나, 게임 개발을 경험해 본 팀원이 없을뿐더러, C#언어를 접해 본 적이 없었기 때문에, 해당 플랫폼과 언어를 습득하는 데 많은 시간을 들여야 했다. 유니티에서 3D로 멀티 게임을 구현하는 것에 대한 정보가 2D에 비해 현저히 적었기 때문에, 유니티와 관련된 다양한 서적을 참고했다.

멀티플레이와 동기화를 구현하기 위해 포톤 클라우드를 사용하였다. 포톤클라우드가 제공하는 메서드 및 기능을 분석하고 최적화하는 것에 어려움을 느꼈으며, 해당 과정과 관련된 자료들은 채팅을 하지 않거나, 게임 방이 존재하지 않고 바로 게임을 시작하는 예시밖에 존재하지 않아 참고 자료 없이 팀원끼리 자체적으로 연구하고 해결해야 했다.

초반에는 플레이어와 오브젝트를 동기화하기 위해서 실시간으로 정보를 업데이트 해주었다. 그러나, 이 구현은 많은 리소스 낭비와 그에 따라 데이터 처리가 느린 결과를 낳았다. 따라서, 플레이어 위치나 정보 같은 실시간 업데이트가 필요한 정보와 인벤토리 업데이트, 아이템 드랍 같은 간헐적인 업데이트가 필요한 경우를 구별하여 각 다른 함수로 구현하여 최적화하였다.

# 6.개발 일정

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **순서** | **기간**  **작업내용** | **12** | | | | **1** | | | | **2** | | | | **3** | | | | **4** | | | | **5** | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| **1** | **프로젝트 일정 조정** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2** | **메타버스 플랫폼 조사** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3** | **Open Source 설치** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **4** | **방탈출 스토리 구상** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **5** | **플레이어 기능 설계** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **6** | **플레이어 기능 구현** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **7** | **Stage 1 구현** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **8** | **Stage 2 구현** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **9** | **Stage 3 구현** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **10** | **Photon 설계 및 구현** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **11** | **플레이어 동기화** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **12** | **스테이지 별**  **오브젝트 추가** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **13** | **1차 버그 수정** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **14** | **오브젝트 동기화** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **15** | **2차 버그 수정 및 업데이트** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **16** | **최종 발표 준비 및 발표** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**: 공통 : 우지연 : 김지현**

# 7. 참고 문헌

C#, 위키백과, <https://ko.wikipedia.org/wiki/C_%EC%83%A4%ED%94%84>

Unity, 위키백과,

<https://ko.wikipedia.org/wiki/%EC%9C%A0%EB%8B%88%ED%8B%B0_(%EA%B2%8C%EC%9E%84_%EC%97%94%EC%A7%84)>

포톤, <https://www.photonengine.com/ko-KR/>

이종원(2021), 게임 제작으로 배우는 유니티, 한빛아카데미

김민석(2019), 가장 쉬운 유니티 게임 제작, 정보문화사

가토마사키(2017), 유니티로 게임을 만드는 10가지 방법, 한빛미디어

조상철(2019), 너도 만들수있어! 유니티 모바일 게임, 비제이퍼블릭

1. 출처=스팀(steam) [↑](#footnote-ref-1)