

프로젝트 제안서

Open Source Software Project

Team04_Magic Blocker

2018111943 전재호

2018112515 류강현

2018111865 백성욱

목차

1. 프로젝트 개요.....	3
1.1 프로젝트 주제.....	3
1.2 선정 오픈소스 분석.....	3
1.2.1 선정 오픈소스 정보.....	3
1.2.2 SWOT 분석.....	3
1.2.3 선정 이유.....	4
2. 프로젝트 목표.....	5
2.1 개발환경 및 라이선스.....	5
2.2 개선 및 개발사항.....	5
3. 예상 결과물.....	6
3.1 프로젝트 시스템 구조.....	6
3.2 예상 시스템 결과.....	7
3.3 기대효과.....	7
4. 프로젝트 일정 및 규칙.....	8
4.1 개발 역할 분담.....	8
4.2 타임라인.....	8
4.3 Git 운영방식.....	9
4.3.1 Git 협업규칙.....	9
4.3.2 표기규칙.....	10
4.4 협업도구 활용방식.....	10
5. 참고문헌.....	11

1. 프로젝트 개요

1.1 프로젝트 주제

이번 프로젝트의 주제는 오픈소스 슈팅 게임인 Blocker.io를 개선한, Magic Blocker.io 를 개발하는 것이다. Blocker.io는 개발자인 jojoe가 개발한 게임으로, github에 그 소스코드가 전부 공개되어 있다. 해당 소스코드에 게임의 핵심 시스템은 이미 구현이 되어있지만, 완성된 게임의 필수요소인 메뉴, 설정, 채팅 시스템 등이 구현되어 있지 않다. 기존 OSSP 수업을 참고하여, 이러한 게임 시스템을 직접 개발하고, 소스코드를 개선하는 것이 개발 목표이다.

1.2 선정 오픈소스 분석

1.2.1 선정 오픈소스 정보

2D 탑 뷰 방식 오픈소스 슈팅 게임이다. 게임 서버는 Node.js이용하였으며, 게임 그래픽으로 Javascript canvas를 사용하며, 렌더링 엔진으로 Javascript 오픈소스 게임엔진인 Phaser3를 사용했으며, Socket.IO를 이용하여 온라인 멀티 게임 플레이를 구현하였다.

- License : MIT License
- Stack : Node.js socket.io Phaser3 gulp.js
- URL : <https://github.com/jojoe/blocker>

1.2.2 SWOT 분석

오픈소스를 정확히 분석하기 위해 SWOT기법을 사용하였다. Strength에는 Javascript의 DOM 방식으로 Component 가 쪼개져 있어서, 각각 개발하고 수정하기 간편한 점이 있으며, 중앙서버에서 게임의 주요 시스템이 실행되므로, 새로운 버전으로 게임의 소스코드를 수정하였을 때, 모든 클라이언트에게 변경사항이 간편하게 적용된다는 점이 있다. Weakness에는 모든 게임 사용자가 중앙 게임 서버를 통하여 통신하기 때문에, 서버 장애 또는 통신 장애가 생기면 게임 서비스를 이용할 수 없다는 점과 중앙서버에 게임 시스템이 구현되어 있기 때문에, 물리적으로 거리가 멀거나 네트워크 신호가 약한 환경에서는 원활한 게임 이용이 어렵다는 점이 있다. Opportunity에는 Web 기술을 기반으로 개발되었기 때문에, 프로그램 설치나 의존성 없이 간편하게 실행할 수 있는 점과 Web

기술을 기반으로 개발되었기 때문에, OS 별로 실행 방법을 따로 안내하는 등의 수고가 적다는 점, 모든 소스코드가 수정/배포 권한이 허용된 라이선스로 구성되어 있다는 점이 있다. 마지막으로, Threat에는 Phaser3 엔진의 작동방식이 내부적으로 복잡하므로, 게임상의 버그나 오류가 생겼을 때, 소스코드를 수정하기가 쉽지 않다는 점과 온라인 게임의 특성상, 게임 서비스를 상용화한다는 가정을 했을 때, 부정 행위 프로그램 사용자, 통신 매체 모욕 및 음란 행위 등을 관리하기 위한 추가 온라인 인력이 필요하다는 점이 있다.

S	<ul style="list-style-type: none"> • Javascript 의 DOM 방식으로 Component가 쪼개져 있어서, 모듈별로 각각 개발하고 수정하기 간편함 • 브라우저 환경에서 실행되기 때문에 OS 별로 실행 방법을 따로 안내해야하는 수고가 적음. • 중앙서버에서 게임의 주요 시스템이 실행되므로, 새로운 버전으로 게임의 소스코드를 수정하였을때, 모든 클라이언트에게 변경사항이 간편하게 적용됨.
W	<ul style="list-style-type: none"> • 모든 게임 사용자가 중앙 게임 서버를 통하여 통신하기 때문에, 서버 장애 또는 통신 장애가 생기면 게임 서비스를 이용할 수 없음. • 중앙서버에 게임 시스템이 구현되어있기때문에, 물리적으로 거리가 멀거나 네트워크 신호가 약한 환경에서는 원활한 게임 이용이 어려움.
O	<ul style="list-style-type: none"> • Web 기술을 기반으로 개발되었기 때문에, 프로그램 설치나 의존성 없이 간편하게 실행할 수 있어, 대중화에 용이하고 잠재적 서비스 이용자 수가 많음 • 소스코드가 전부 오픈소스 라이선스라는 점, 온라인게임이라는 점 때문에 GoogleAds, 온라인 쇼핑물 링크 등 온라인 비즈니스 수익 모델을 추가하기가 용이함.
T	<ul style="list-style-type: none"> • Phaser3 엔진의 작동방식이 내부적으로 복잡하므로, 게임상의 버그나 오류가 생겼을때, 소스코드를 수정하기가 쉽지 않을듯함. • 온라인 게임의 특성상, 게임 서비스를 상용화한다는 가정을 했을때, 부정 행위 프로그램 사용자, 통신 매체 모욕 및 음란 행위 등을 관리하기 위한 추가 온라인 인력이 필요할 것으로 예상됨.

Figure1. 오픈소스 분석을 위한 SWOT기법

1.2.3 선정 이유

CD 또는 패키지로 판매되던 비디오 게임이 게임 산업의 주 수익 모델이었던 과거와 달리, 온라인 게임과 스마트폰 게임 어플리케이션이 주요 플랫폼으로 바뀌었다. 따라서 기기와 운영체제 환경에 구애받지 않으며, 확장 가능하다는 점에서 Web 환경에서 구동되는 io 게임이 설치형 게임에 비해 확장성이 뛰어나다. Google, Naver 등 대기업의 광고 API를 이용하면 인-앱 광고, 링크 이동 등의 비즈니스 로직을 쉽게 구현할 수 있는 점에서 수익모델을 추가하기에도 용이하다는 장점이 있다. 따라서 io 게임의 핵심요소인, 브라우저로 구동하는 실시간 통신 방식을 구현한 해당 오픈소스를 프로젝트 대상으로 선정하였다.

2. 프로젝트 목표

2.1 개발환경 및 라이선스

개발 대상이 여러 멀티 미디어 작업이 필요하므로 Ubuntu 환경에서 개발하기로 하였다. Node.js 개발에 다양한 편의기능을 제공하며, Github 플러그인을 사용하여 협업을 간편하게 할 수 있는 무료 텍스트 에디터인 Microsoft 사의 Visual Studio Code를 개발툴로 선정하였다. 어도비사의 이미지 편집 및 일러스트레이션 도구를 이용하여 게임 캐릭터 이미지와 각종 그래픽을 작업한다. 웹언어인 자바스크립트를 이용한다. 수정/배포가 자유로운 MIT 라이선스로 배포하였다.

- OS: Ubuntu
- Code editor: vscode
- Image editor: Adobe Illustrator, Adobe Procreate
- Language: javascript
- Collaboration Tool: Notion, Github, Gantt Pro, Slack
- License : MIT License

2.2 개선 및 개발사항

Blocker.io는 게임 핵심 시스템이 이미 개발되어 있지만, 게임을 원활하게 플레이하기 위해 필요한 시스템 UI 등의 요소가 포함되어 있지 않다. 완성된 게임을 위해 필수적인 각종 요소를 구현하고, 다채로운 게임 플레이를 위해 필요한 각종 요소들을 개발하는 것이 주요 개선 사항이다.

Table1. 오픈소스 개선사항

문제점	개선사항
단조로운 캐릭터 종류	사용자가 선택할 수 있는 여러가지 캐릭터 클래스를 추가하여 게임성 향상
게임 내에서 플레이어끼리 소통할 수 있는 방법의 부재	채팅 기능 등의 커뮤니케이션 시스템 추가

게임의 승리 조건 또는 목표가 설정되어 있지 않음	랭킹과 게임 오버 등을 구현하여 플레이어의 목표를 설정하고 동기를 부여함
게임 플레이 방식이 단조로움	단순한 런앤건 방식을 개선하여 장애물, 랜덤 아이템 등의 흥미요소를 추가
게임 UI가 없어, 사용자가 게임에 접속하거나 설정하는데 어려움이 있음	편리한 사용자 UI를 구현하고 계정으로 로그인, 로비화면 등 상용 게임 소프트웨어가 포함된 클라이언트 요소를 추가

3. 예상 결과물

3.1 프로젝트 시스템 구조

게임 실행 후 로그인을 하고 메인 화면으로 진입하도록 구성했다. 메인 화면에 들어가면 3가지 선택지가 존재한다. '게임 시작', '설정', '종료'가 그것들이다. 게임을 시작하고 싶다면 '게임 시작' 버튼을 누른 후 자신이 플레이 할 캐릭터를 정하고, 원하는 맵을 선택한다. 이후 맵에 들어가면 게임 도중 유저들끼리 채팅을 할 수 있게 구성한다. 계속해서 게임을 진행하다 자신의 캐릭터가 죽었을 경우, Game over 화면이 나오면서 'Restart' 버튼을 통해 메인 화면으로 넘어가거나 종료할 수 있게 한다. '설정' 버튼을 누르면 조작 방식을 세팅할 수 있다. 'To main'을 누르면 다시 메인 화면으로 돌아갈지 말지 결정할 수 있다. '종료' 버튼을 누르면 게임이 종료된다.

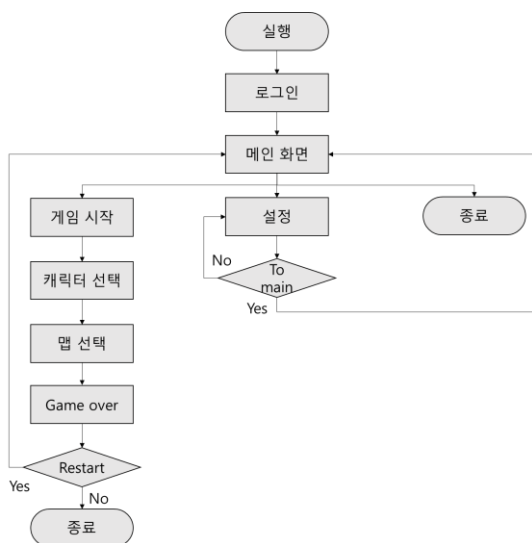


Figure2. 게임 시스템 구조를 정리한 Flow Chart

3.2 예상 시스템 결과

필요한 부분에 대해 새로운 이미지로 대체하여 각 영역 및 사물이 무슨 역할을 하는지 알기 쉽다. 다양한 특징을 갖는 캐릭터와 맵을 제공하여 사용자에게 폭 넓은 사용 경험을 제공한다. 개인 별로 조작 방식을 세팅 가능하게 하여 사용자에게 편의성을 제공한다. 로그인 창, 메인 화면 제작을 제작하여 체계적으로 게임을 구성한다. Figure3과 같은 채팅 시스템 도입을 통해 사용자간 소통이 가능하게 한다.



Figure3. 채팅 시스템 구현 화면 예시

3.3 기대효과

이미지 수정을 통해 게임의 가시성을 높이고 직관적인 이해가 가능하게 한다. 로그인 시스템 도입으로 계정을 육성하는 재미를 높일 수 있으며 레벨 시스템 도입이 가능해진다. 조작 방식의 사용자화를 통해 편의성을 높일 수 있다. 채팅 시스템을 통해 사용자끼리의 소통이 가능해지며 여러가지 요소들의 추가로 사용자가 느끼는 게임성을 향상시킬 수 있다. 이를 통해 궁극적으로 게임을 찾는 사용자 수의 증가를 기대할 수 있다.

오픈소스 라이선스(MIT, GPL-3.0 등)에 대한 이해와 협업 툴(GitHub, Notion, Gantt chart) 활용법 습득이 기대된다. 또한, Javascript와 관련된 라이브러리에 대한 학습과 웹 상에서

진행되는 게임이므로 서버와의 통신 과정을 이해할 수 있다.

4. 프로젝트 일정 및 규칙

4.1 개발 역할 분담

적절한 업무 분담을 위해 WBS를 사용하여 할 일을 정리하였다. 이를 통해 캐릭터 맵 디자인/구현, 알고리즘 개발과 서버 구축, 그리고 UI 디자인/구현으로 역할을 나누었다.

Table2. 역할 분담 표

전재호	캐릭터, 맵 디자인 및 구현
류강현	알고리즘 개발 및 서버 구축
백성욱	UI 디자인 및 구현(로그인 화면, 로비 화면, 채팅 시스템 등)

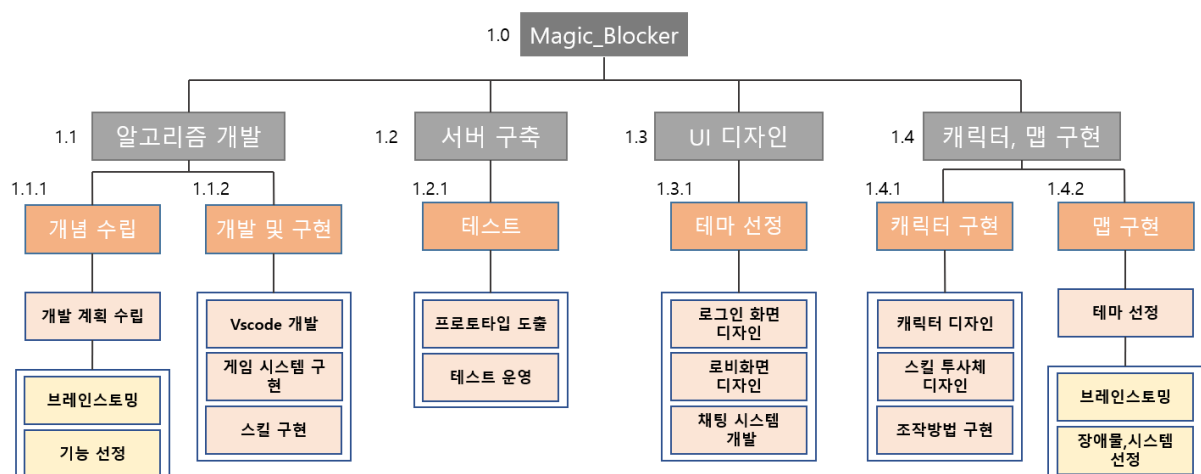


Figure4. 역할 분담을 위한 WBS(Work breakdown structure)

4.2 타임라인

효율적인 업무 실행을 위해 Gantt Pro 툴을 이용한 Gantt chart로 타임라인을 구성했다.

Docs	문서 수정
Refactor	코드 리팩토링(변수명 수정 등)
Style	코드 스타일 변경, 코드 자체의 변경이 없는 경우
Remove	파일 또는 코드, 리소스 제거
Resource	이미지 리소스 등의 코드와 상관없는 리소스 추가

4.3.2 표기 규칙

효율적이고 통일성 있는 코드를 위해 표기 규칙을 정한다.

- 변수: snake_case 표기
- 클래스: camelCase 표기

4.4 협업도구 활용방식

Gantt Pro를 사용하여 일정관리를 진행한다. 한 명이 대표로 수정하며, task 수를 비슷하게 설정하여 공정하게 역할을 분배한다. Notion을 활용하여 문서 작성을 공유한다. 비정기적 피드백 및 회의 날짜 조정 등 기타 연락은 Slack을 통해 진행한다.

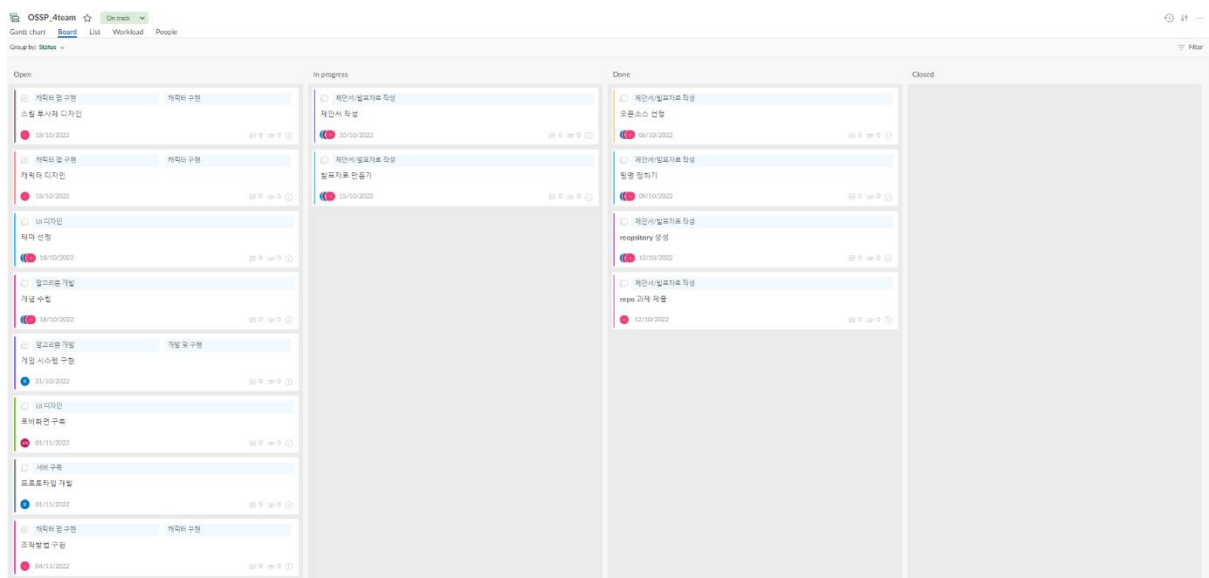


Figure6.1. Gantt Pro를 이용한 Gantt chart task 관리

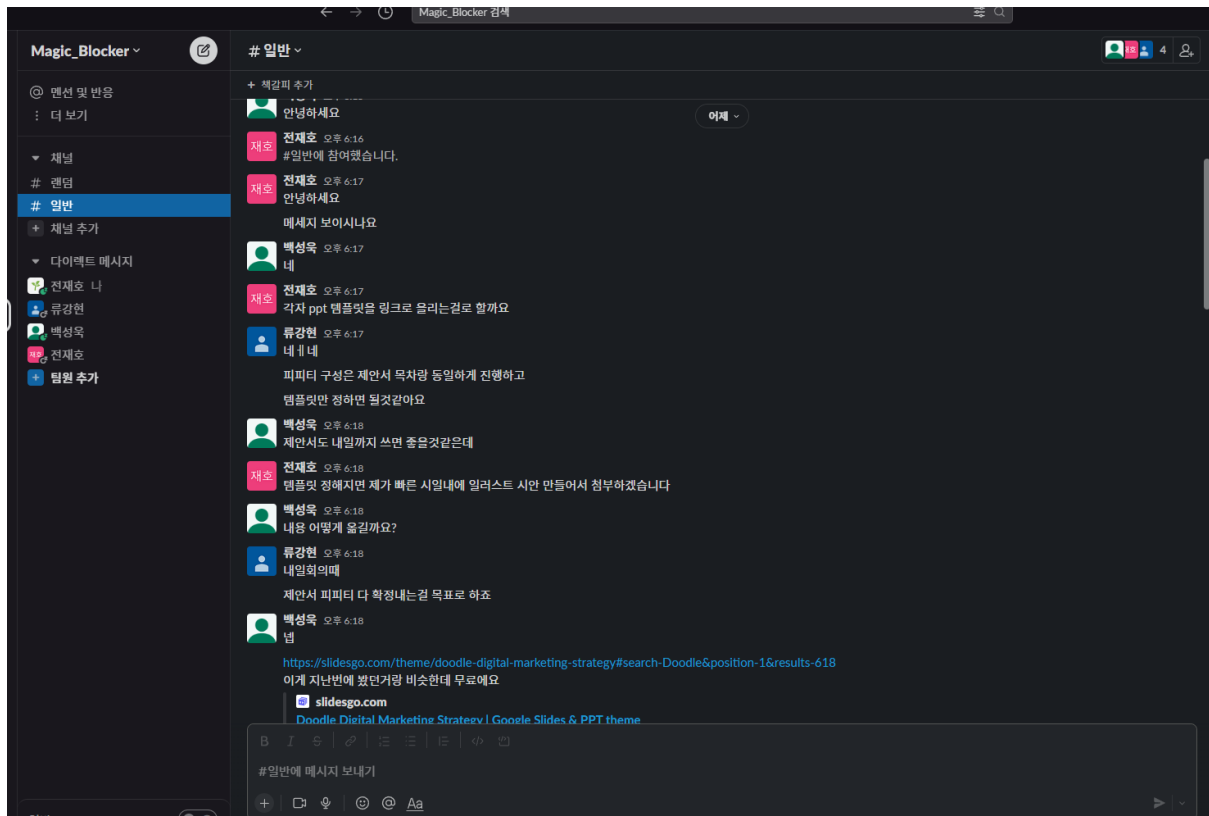


Figure6.2. Notion을 이용한 회의 일정 조율 및 비정기적 피드백

5. 참고문헌

[1] Jojee, ed. "Blocker." Github, September 9, 2021. <https://github.com/jojoe/blocker>.