

Team Blue Bird

Poor Guys Project





도전! 폴가이즈 클론 프로젝트



## 목차

	01	소개	<ul style="list-style-type: none"><li>- 업무 분담</li><li>- 개발 일정</li></ul>	
	02	목표 설정	<ul style="list-style-type: none"><li>- 개인 목표</li><li>- 팀 목표</li></ul>	
	03	폴 가이드	<ul style="list-style-type: none"><li>- 시장 분석</li><li>- 어째서 폴 가이드인가?</li></ul>	
	04	세부 계획	<ul style="list-style-type: none"><li>- 그라운드를</li><li>- 성능 검증</li></ul>	

## 업무 분담

	<p>김정도</p> <p>팀 리딩 유니티 엔진을 활용한 로그인 인증 시스템 구현</p>		<p>김민관</p> <p>유니티 엔진을 이용한 플레이어 캐릭터의 기본 동작과 맵 구현</p>
	<p>임우영</p> <p>로그 분석을 위한 유니티 엔진과 데이터베이스 연동</p>		<p>홍지현</p> <p>IOCP 게임 서버 개발 및 세션 관리</p>

## 개발 일정

	12월	1월				2월			
	4	1	2	3	4	1	2	3	4
폴 가이드 해체 분석	~29								
PMP 작성	~31								
아키텍처 설계		~7							
개발 환경 구축		~2							
기초 연동			~9						
Unity 및 IOCP 연구 학습			~16						
매치 메이킹 및 맵 구현				~27					
서비스 확장 및 품질 개선					28~				

## 개인 목표



김정도

### 1. 웹 백엔드 개발자로서의 경험적 기반 마련

#### Web Backend Basic, Network Communication

- 게임 개발 과정에 함께 참여하며 웹 서버 사이드의 게임 플랫폼 개발을 담당, 유저들이 원활한 게임 이용을 할 수 있도록 지원
- 게임 플레이 환경 구축에 필요한 간단한 서버 사이드의 작업들을 처리해 보면서 기초적인 백엔드 경험을 쌓는다
- 웹 서버에서 유저를 관리하고 해당 정보를 게임 서버와 실시간으로 필요한 정보를 공유하는 과정에서 네트워크 통신과 관련한 기술과 경험

## 개인 목표



김정도

### 2. 협업 경험을 통한 기업 실무에 가까운 개발 체험

#### GitHub, Git Flow

- 이슈 관리와 Git Flow 방법론으로 branch를 활용하고 폴더 구조나 프로젝트를 체계적으로 구조화한다
- 전체 개발 과정에서 꼭 필요한 역할을 담당하며 팀 프로젝트의 구성원으로써 책임감 경험
- TortoiseGit, ZenHub 등의 프로젝트 관리 툴을 적극 활용해 보다 효율적인 협업 방법을 습득함

## 개인 목표

김정도



### ( Work Flow )

- 게임 플랫폼 웹 브라우저 개발
  - > 웹 디자인과 기초적인 클라이언트 환경 구성
  - > 회원가입과 로그인 등의 인증 서버 구현 (JWT 인증/인가, 로그 관리)
  - > 유저 매치 메이킹 등을 처리하는 로비 서버 구현
  - > 게임 서버와 연동 및 통신 (소켓 프로그래밍)
- 게임 채팅 서버 개발

## 개인 목표



김민관

Clean한 폴더 구조로 만드는 습관 들이기

유니티를 이용한 게임 클라이언트 개발 능력 향상

매일 한 시간 이상, C# 기본 문법 공부

서버, 플랫폼과 클라이언트의 연동 방법 연구

클라이언트와 백엔드의 협업에 대한 흐름 파악

Git 협업 능력 향상



## 개인 목표



김민관

### ( Work Flow )

- Unity 3D 개발
  - > 기초 개발 방법 숙지
  - > 레이캐스트 사용 방법 숙지
  - > 충돌 및 머티리얼 연구
- 맵 제작
  - > 스테이지 맵 제작
  - > 함정 배치

## 개인 목표

김민관



### ( Work Flow )

- 클라이언트 세부 기능 구현
  - > 벽 등반
  - > 점프 슬라이딩
  - > 플레이어 붙잡기
  - > 맵 함정
- 클라이언트와 IOCP 연동
  - > Unity에서 IOCP 서버와의 연동
  - > Protocol buf를 이용한 소켓 통신

## 개인 목표



임우영

1. Understandability가 높은 코드 작성
  - 당일 작성한 코드를 분석해 보며 해당 내용을 README 파일에 작성
  - 코드가 완성되면 Design Pattern에 기반한 Refactoring 작업 시행
2. 다양한 API, 오픈 소스를 활용할 수 있는 능력 기르기
  - 일주일에 한 개 이상 프로젝트에 적용 가능한 API 또는 오픈 소스 탐색
  - 해당 API 또는 오픈 소스가 적용될 수 있는 방안과 사용법을 개인 Git repository의 README에 작성

## 개인 목표



임우영

### ( Work Flow )

#### 1. Unity 3D 기초 습득

- Unity에서 사용되는 C# 언어 학습
  - 플레이어의 입력에 따른 동작 구현
  - 외부 라이브러리 및 컴포넌트 사용법 이해
- Asset과 C# 파일 연동
  - Asset에 Collider를 추가해 충돌 이벤트 구현
  - Asset Animation 학습

-> 학습 진행 상황은 개인 Git Project에 반영하고,  
날짜 별 학습 내용을 Commit에 기록

## 개인 목표



임우영

### ( Work Flow )

#### 2. Unity와 웹 / IOCP 서버 연결 및 통신

- Unity와 웹 서버 연결
  - RESTful API를 사용한 HTTP 통신
  - 연결 확인을 위한 Query문 전송
- Unity와 IOCP 서버 연결
  - Socket 통신
  - 서로 다른 언어 사이에서 데이터 교환과
  - 빠른 속도를 위한 직렬화 전략 (Protocol Buffer 이용)

-> 패킷 통신 결과를 Unity와 각 서버에 출력하며 테스트

## 개인 목표

임우영



( Work Flow )

### 3. 웹 / IOCP 서버와 Database 연결

- 각 서버와 Redis 연결
    - 게임과 Database 사이 빠른 데이터 I/O을 위해
    - In-memory 기반의 Redis 사용
  - Redis와 MySQL 연결
    - Redis에 데이터가 쌓일 경우,
    - MySQL에 해당 데이터들을 저장해 Redis의 부하를 줄임
- > MariaDB와 같은 대체 Database와의 연결 및 성능 또한 테스트

## 개인 목표

임우영



( Work Flow )

### 4. 게임 Log 분석

- Database에 저장된 Log 내용들을 토대로 의미있는 정보를 도출
  - 접속자 수가 가장 많은 시간대가 언제인지
  - 가장 인기있는 맵 또는 모드가 무엇인지

추가 고려사항

- Redis Server를 Docker / Cloud 환경에서 구동 시 성능 차이 분석
- Database에 Index와 Docker를 적용했을 때와 적용하지 않았을 때의 데이터 I/O 시간 분석
- Unity Prefab과 맵 제작 (프론트엔드 개발 보조)

## 개인 목표



홍지현

1. 각 클라이언트의 딜레이에 종속되지 않고,  
같은 게임에 접속한 클라이언트들이  
동일한 상태로 게임을 플레이할 수 있는 게임 서버 구현
2. Unity와 C++ 서버가 통신할 수 있도록 서버 코어 구현



## 개인 목표



홍지현

### ( Work Flow )

1. IOCP를 제대로 활용하여 게임 서버 구현
  - 멀티스레드를 활용하여 게임 서버 구현
    - 데드락이 발생하는지 체크 (코드로 구현해보기 -> 강의에 존재)
    - 게임 기능 세부화
  - IOCP에서의 세션 관리
    - 각 플레이룸 별로 세션 관리
    - 구조 별로 전체 세션 관리
  - 세부 기능 직접 구현해보기
    - 스레드 락, 메모리 풀 등
  - README에 이해하기 쉽게 구조 설명

## 개인 목표

홍지현

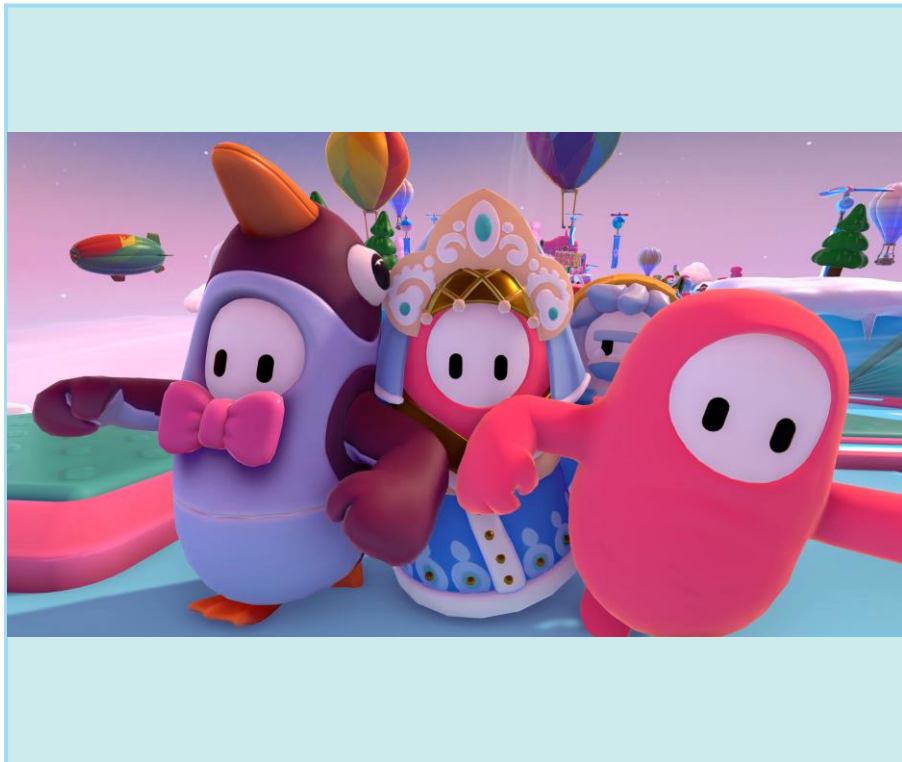


( Work Flow )

### 2. Unity와의 연동

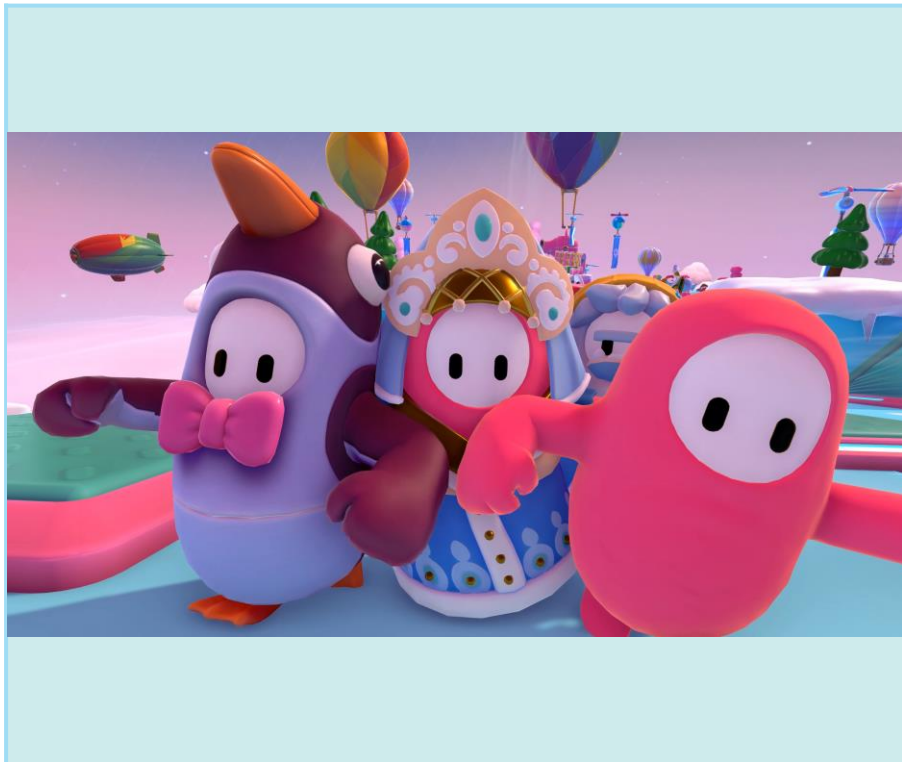
- Protobuf를 통해서 직접 패킷 구조 설계
  - 자동으로 패킷 생성하는 방식까지 구현 (패킷 자동화)
- 각 세션 별로 동시에 동일하게 화면이 보이게 하기 -> 레이턴시
  - 레이턴시를 최소화 (레이턴시 측정을 통해 줄여나가기)
  - 각 인터넷 속도 차이에도 동일하게 화면이 출력되도록 구현

## 팀 목표



- 이해할 수 있는 프로그램 구현
- 각 아키텍처를 왜 사용해야하고,  
Flow에서 어떠한 역할을 하는지 명확히 하기

## 팀 목표



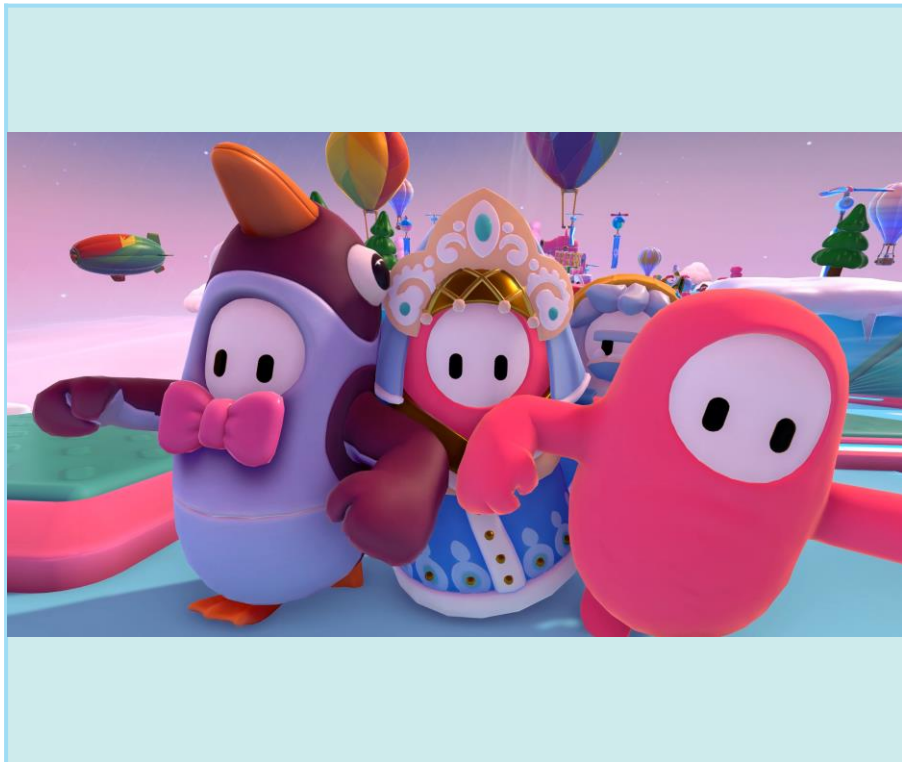
- 이해할 수 있는 프로그램 구현
  - ✓ 왜 해당 아키텍처를 사용하는지 이유를 분석
  - ✓ 각 아키텍처와 Readme 만으로도 그 기능과 역할을 손쉽게 이해할 수 있도록 상세히 기록함.
  - ✓ Design Pattern을 사용한 Refactoring 작업으로 코드의 유지 보수성을 높임

## 팀 목표



- 팀 프로젝트 활동을 통해 개발 습관과 책임감 향상
- 서로 이끌어 줄 수 있는 분위기를 형성하고,  
프로젝트를 성공적으로 완성시킬 수 있도록 한다
  - ✓ (성취 기준 : 프로젝트 목표)

## 팀 목표



- 팀 프로젝트 활동을 통해 개발 습관과 책임감 향상
  - 그라운드 룰을 이행하며 규칙적인 개발 습관을 함양하고, 구성원으로서 팀 내에서 맡은 각자의 역할을 수행하며 책임감을 느낀다
- ✓ (성취 기준 : 코어타임 참가 횟수, 임무 분담 등)

## 프로젝트 목표



- IOCP를 활용한 대규모 인원을 수용할 수 있는 서버 구현
- 게임 엔진(유니티)과 게임 서버(IOCP)를 안정적으로 연결하기
  - 유니티를 이용한 인증 시스템 구현
  - 아키텍처 설계 및 DB 스키마 설계

## 프로젝트 목표



- 게임 플레이 시 발생하는 이벤트 로그를 정제하여 DB에 저장
  - 게임 레벨 디자인
  - 플레이어의 기본 동작, 함정 구현
  - Git 을 사용하면서 협업 환경에 익숙해지기



## 폴 가이즈 ( Fall Guys )

### 무료 선언한 폴가이즈, 이용자 2,000만 돌파... F2P 효과 제대로

출시 48시간 만에...

김재석(우티) 2022-06-27 14:50:01

에픽게임즈 스토어에서 전면 무료화된 멀티플레이 파티 게임 <폴가이즈>가 동시접속자 2,000만 명을 기록했다. 개발사 미디어토닉은 "<폴가이즈>를 무료로 전환한 지 48시간 만에 2천만 명의 플레이어를 기록했다"라고 축하했다.

미디어토닉은 2021년 에픽게임즈에 인수됐고, 양사는 합의를 통해 <폴가이즈>의 무료화를 결정했다. <폴가이즈>는 에픽게임즈 스토어(PC)에서 아니라 Xbox One, Xbox S에서도 제공 중이며 크로스 플레이를 지원한다.

에픽게임즈는 인기 타이틀을 무료로 제공하면서 자사 스토어에 모객 효과를 노리고 정기적으로 <GTA 5>, <보더랜드 3> 등 인기 타이틀을 무료로 배포했다. 이어서 멀티플레이 게임을 무료로 배포하면서 이용자 확보에 열을 올리고 있다. 2021년 <폴가이즈>도 입점했고 게임은 그해 5월 28일부터 6월 3일까지 무료로 제공됐다.

<폴가이즈>는 2020년 8월 출시된 멀티플레이 게임이다. 게임은 출시 초기 선풍적

확산에 따른 세계적인 격리가 이어지는 상황에서 여러 명이 모여 미니 게임을 수행한다는 컨셉으로 유행이 됐다.



## 폴 가이즈 ( Fall Guys )

### 무료 선언한 폴가이즈, 이용자 2,000만 돌파... F2P 효과 제대로

출시 48시간 만에

기대서 / 오픈

### 에픽게임즈 폴가이즈(Fall Guys), 단 2주 만에 5천만 돌파

이재덕 기자 입력 2022.07.08 14:04

에픽게임즈의 플랫폼어 게임 '폴가이즈'가 무료 게임으로 재출시한 후 단 2주 만에 5천만 명의 이용자를 확보했다.

7일(현지시간) 유로게이머 등 주요 외신들은 이 소식을 타전했다.

같은 날 폴 가이즈 개발자는 트위터에 이 소식을 공유하며 "정말 믿기지 않는다... 2주 만에 5천만 플레이어!!!"라는 트윗을 올렸다.

폴가이즈는 무료 출시 48시간 만에 2천 만 명을 돌파했다. 당시

미디어토닉(Mediatonic)팀은 트위터를 통해 "우리는 무료화 첫 48시간 동안 놀라운 2천만 명의 플레이어에 도달했다!"고 밝혔다.

이 소식에 미국의 닌텐도는 "엄청난 이정표를 축하한다"고 했고, 이 글에는 2,000개가 넘는 리트윗과 3.5만의 좋아요가 달렸다.



Fall Guys... FREE FOR ALL! 🍷🏆  
@FallGuysGame · Follow

We're in absolute disbelief...

50 million players in 2 weeks!!!

UNBELIEVABLE

THANKS EVERYONE! 🌍



12:02 AM · Jul 7, 2022

35.4K Reply Share

Read 847 replies

## 폴 가이즈 ( Fall Guys ) 란 ?



- 남녀노소 모두가 즐길 수 있는  
유니티 기반의 3D 플랫폼어 게임
- 최대 60명의 인원이 함께 플레이  
하는 멀티플레이 게임
- 토너먼트 방식의 경쟁 게임
- 자동 매칭 시스템을 통해,  
쉽게 게임 참여 가능

## 어째서 폴 가이즈인가 ?



### 1. 폴가이즈의 성공 사례

- 대다수의 파티 게임들은 모든 맵을 섭렵한 이후부터는 반복 플레이로 인해 재미가 반감되는 한계가 있다.
- 하지만 폴가이즈의 경우 꾸준한 신규 시즌과 함께 새로운 미니 게임들을 업데이트하며 즐길 거리를 끊임없이 제공해왔다.
- 또한 폴가이즈는 시즌을 거듭하며, 2022년 7월 기준 동시 접속자 수가 2천만 명을 돌파하는 등 우수한 성공을 거두었다.
- 따라서 우리는 폴가이즈의 클론 프로젝트를 진행하며, 성공한 게임의 개발 과정을 따라가보려 한다.

## 어째서 폴 가이즈인가 ?



### 2. 프론트엔드 인원의 부족

- 프론트엔드 1명과 백엔드 3명으로 팀이 구성되어, Client의 비중을 줄이고 Server의 비중을 높이는 방안을 채택.
- 따라서 Client가 단순한 플랫폼 게임인 폴가이즈를 대상으로 클론 프로젝트를 진행하기로 함.

## 어째서 폴 가이즈인가 ?



### 3. 개발 과정에서 다양한 경험과 기술 습득

- 대규모 멀티 플레이 서버를 구현하고 서버의 트래픽을 관리할 수 있는 기술 습득
- 게임과 인증 서버, 데이터베이스 등 서버 간 연동을 구축하고, 통신 프로토콜을 구현하는 경험
- 다양한 맵과 함정, 기믹 그리고 이벤트를 구현할 수 있는 기회
- 게임에서 발생한 이벤트 및 로그에서 의미 있는 정보를 추출하는 경험



## 그라운드 룰



- Core Time

매일 13:00 ~17:00, ZEP 스페이스

- 집중 개발 타임:

Winter://Dev.Camp에 모여서  
프로젝트에서 본인이 맡은 역할에 집중하는 시간을 갖는다

- 온라인 회의:

집중 개발 타임을 마무리하며  
매일 30분 정도의 비대면 회의 시간을 갖는다

- 오프라인 회의:

매주 금요일 13:00에 오프라인으로 모여서  
비대면 회의에서 부족했던 부분들을 해소하고  
서로 교류하며 더욱 효율적인 팀 회의 시간을 갖는다

## 그라운드 룰



- Code Convention

표기법: <snake\_case>

ex)

snake_case	(Local variable)
is_dead	(Boolean variable)
Get_Value()	(Function)
MAX_VALUE	(Constant variable)
g_Num	(Global variable)

tap vs space: <space>



## 그라운드 룰

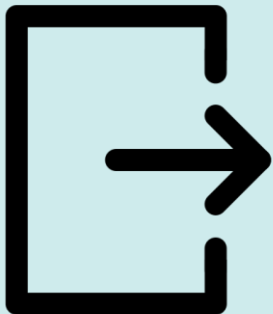


- 기타

회의록, 개발 일지는 번갈아가면서 작성

GitHub 연동 및 소통 등에 Slack을 적극 활용

## 성능 검증 및 오류 체크



### • 인증 서버

#### # 성능 검증

- 다량의 인원이 동시 로그인 시 속도 측정(TPS 측정)

#### # 오류 체크

- 인증 실패 시 실패 처리
- ' , - , \_ 등 SQL 인젝션(삽입, injection)에 대한 보안 필요
- 로비 입장 시 게임 서버에 세션 넘어가는지 확인 필요

## 성능 검증 및 오류 체크



### • 게임 서버

#### # 성능 검증

- 트래픽이 몰려도 정상적으로 작동 및 처리
- 패킷 손실이 많이 생기는 클라이언트의 경우 어떻게 대처할 것인가
- 대량 패킷을 받을 시 처리 속도 체크
- 패킷의 오류 체크 및 보안 검증
- 멀티스레드 사용시 데드락 걸리는지 확인

## 성능 검증 및 오류 체크



- 클라이언트

- 서버 latency time(지연 시간) 확인
- 스케줄링에 따라 순차적인 처리

- 로그 분석

- 어느 어느 서비스에 로그를 남길 것인가

ex) 로그인 로그, 게임 관련 로그(시작, 종료, 우승 등)

- 각각 로그를 어디에 활용가능한가

ex) 어떤 모드를 많이 즐기는지, 어느 시간대에 사람이 많이 몰리는지 등



TEAM BLUEBIRD

Fin.