YamYam

현명은, 구자현, 박동진, 조현민

목차

프로젝트 NoPOKER 설명 및 이유

“인디언 포커 하죠?”

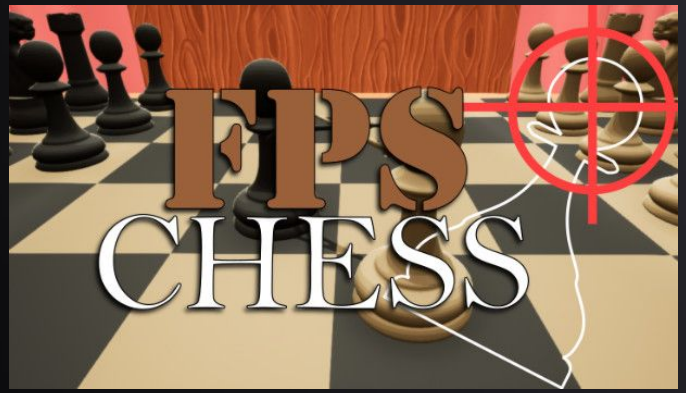
프로젝트 NoPOKER 설명 및 이유

게임계의 평양냉면

프로젝트 NoPOKER 설명 및 이유

몬가.....
MSG가 필요해

프로젝트 NoPOKER 설명 및 이유



체스인척 하는
FPS게임

이걸 인디언 포커에?
당장하자!

프로젝트 NoPOKER 설명 및 이유

게임 컨셉

인디언 포커를 기반으로 한 난투형 액션 게임

게임 규칙

- 인디언 포커를 진행한다
- 숫자가 높은 사람이 좋은 무기를 가질 수 있다.
- 인디언포커가 끝난 후, 플레이어간 전투를 한다.
- 이 라운드에서 모든 상대방을 쓰러뜨린 경우 베팅된 금액을 모두 가져간다.
- 파산하지 않은 최후의 1인만 남는 경우 게임이 종료된다.

프로젝트 NoPOKER 설명 및 이유

‘액션게임까지 붙이다보니
걱정되는건 성능’

성능향상을 위해 **이 정도**는 하자!

최대 50개의 게임 방을 지원	Ping 30ms 및 FPS 60 제한
동접자 200명을 수용 가능한 서버	반응이 빠른 실시간 서버

팀목표

반응이 빠른 실시간 게임 구현하기 (FrameTime, Ping)

<u>유니티 핑 체크</u>

- Client : 3D 게임의 Frame, Ping, 2D 게임의 Frame
- Server : 2D 게임의 Ping, 채팅 서버의 Ping

→ ping 30ms를 넘기지 않도록하기

→ 60FPS 이상의 프레임 속도를 목표로 하기

동시 접속자 200명까지 수용 가능한 서버 구현하기

2주마다 성능 테스트 프로그램을 통해 서버 성능을 체크하기

- Client : Unity 프로파일러를 통해 실시간 서버 성능 확인
- Server : 서버 자체 테스트, SQL 쿼리 성능 테스트, 캐시 성능 테스트 등 수행

쾌적한 게임서버 구축하기

현재 팀의 상태

- 게임서버 구현해본 적 없음
- 클라이언트 -> 서버를 사용한 게임 개발 경험 없음
- 서버 -> 게임 엔진을 사용해본 적 없음
- 화목하다
- 오프라인으로 한번도 안봄..?
- 서로 코드 한번도 본적 없음
- 깃허브 사용할줄 앎!!
- 이슈 얘기 하는게 부담스러운 상태?(일이 많아지니까)

두달 뒤의 팀의 상태

- 이제 서로 코드 보면 바로 파악가능!
- 화목함
- 서로가 어떤 이슈와 개발과정을 거쳤는지 아는..
- 자유로운 분위기
-
- ._____.
- 말하는 것에 있어서 부담갖지 않는 팀
-

팀 목표

- 팀원끼리 한눈에 보면 의도가 파악되는 코드 작성 : 클린코드 작성
- 화목한 팀 분위기 유지하기
- 팀원끼리 온전히 게임 한판을 즐길 수 있을만큼 구현하….
- 회의 및 개발 시간 효율적으로 사용하기
- 깃허브를 통한 형상관리하기
- 만든 게임 출시하기?
- 개인 목표 달성
- 척하면척
- 본인 생각 자유롭게 이야기
- 이슈를 자유롭게 공유할 수 있는 분위기
- 커뮤니케이션의 중요성을 아는 팀이 되기
- 서로의 생각을 존중하고 수용할 수 있는 분위기를 가진 팀
- -> 데일리 스크럼 하는 이유?

저는 배포할 때를 고려해서 읽기쉬운 코드를 작성하는 것을 목표로 했었는데 이 목표는 오늘 강의를 듣고 목표가 될 수 없다는 것을 깨달았습니다. 이 목표를 어떻게 하면 구체적으로 수정할 수 있을지 어렵습니다 ㅠㅠ

→ 클린코드 하나 읽고 그책에서 하라는 기준대로 코드작성하기로 하면 될까요?

→ 그건 수행 방법에 가까울것 같다. 그 목표가 왜 머릿속에 있는 키워드일까?

→ 제가 짠 코드도 제가 잘 못읽겠어서..

→ 그건 좀 문제있는거 아닌가? ㅋㅋ 배포할때를 고려한다는 말은 왜들어갔나? 남에게 보여주기 위해서? 가독성에 대한 문제면 그것 자체로 목표를 삼아야함. 가독성이 좋다는 게 뭘까? 본인 스스로 못읽겠다매, 구체적인 방법이 나오는게 좋을 것 같은데... 클린코드를 짜려는 노력이 있을것이고, 가독성을 높이는 작업을 하고 싶다면 본인이 스스로 어느 일정한 시간이 지났을때 코드를 읽을 수 있어야함. 그러면 일차적으로 가독성이 높아졌다고 볼 수 있을

현재 팀의 상태

- 개발이슈 관련 얘기가 서로 부담스러운 상태
- 회의 할때 오디오 공백이 많이 생기는 상태
- 질문하는게 눈치보이는 상태
- 자신이 구현해야할 작업에 대해 자신이 없어 대답을 잘 못하는 상태?

두달 뒤의 팀의 상태

- 화목함
- 자유로운 분위기
- 말하는 것에 있어서 부담갖지 않는 팀
- 이슈를 자유롭게 공유할 수 있는 분위기
- 커뮤니케이션의 중요성을 아는 팀
- 서로의 생각을 존중하고 수용할 수 있는 분위기를 가진 팀
- 자신이 작업한 것에 대해 확신있게 말할 수 있는 팀

YamYam’s workflow

데일리 스크럼

- 9:30 ~ 10:00
- 건강, 한 일, 할 일, 이슈, 질문

건강 상태 파악을 통해 팀원 태스크 진행속도 가늠

작업 상황 공유를 통한 동기 부여

이슈, 질문을 매일 작성하며 자유로운 분위기 형성

YamYam’s workflow

GROUND RULE

- 코어타임 : 평일 13:00 ~ 15:00
- 화날때 ‘용용체’쓰기
- 의사결정 방식 : 사다리 타기
- 읽씹 안하기, 읽었다는 표현해주기
- 잡담나누기

자유로운 팀 분위기 유지

YamYam's workflow

정기회의

주간 회고 + 주간 회의
금요일 20:00, 강의 있으면 강의 끝나고

회의내용

- 개발 상황 공유
- 개발일지 작성

준비할 것

- 느낀점
- 회의해볼 필요가 있는 사항
- 일주일간 개발 상황
- TodoList 작성

회의 Deadline까지 해야한다는 동기 부여

To Do List 작성을 통해 남은 태스크 파악

비대면으로 인한 작업상황 온라인 공유 필요

목소리로 대화하는 것의 중요성

현재 개개인의 상태 (현명은)

- 유니티 프로젝트 경험 2번(개인프로젝트, 팀프로젝트)
  개인 : 쿠키런 비슷한게임
  팀 : 타워디펜스 게임

  → 완전한 구현을 하지못했다..
  다른사람이 봤을때 한눈에 봐도 이해가 되는 코드가 아닌 나만 아는 코드를 작성하였다....
  다른사람한테 코드를 설명할때 항상 코드구현의 근거가빈약했다..자신있게 설명을 못한다.
  왜냐? 나도 일단 되니깐 적어놓은거라서.. 구체적으로 어떻게 동작하고 왜 다른방법이 아닌 이 방법을 써서 구현해야하는지에 의문을 가지고있지 않고 구현한다.

- 코드를 생각하지않고 짠다
- 전체적인 설계를 먼저 하지 않고 구현부터한다
- 구현에만 급급하다
- 프로그램의 성능을 생각하지않는다
- 코드를 깔끔하게 쓰는 습관을 들여야하는데..
- 유니티를 능숙하게 잘 사용하고싶다
- 서버나 통신을 이용해서 게임을 구현해본적이 없다.

→ 유지보수를 생각하지않고 구현에만 급급한걸 어떻게 해결하지.
→ 코드를 작성할때부터 애초에 코드를 예쁘게 쓰자?

유니티게임과 서버가 어떻게 통신하는지 설명할 수 있는 사람

현재 개개인의 상태(박동진)

- 게시판 팀 프로젝트 경험
- HTTP api 개발 경험
- 제대로 된 MSA 경험 없음
- TCP 통신 해본 적 없음 (HTTP 기반 통신만 해봄)
- 라이브러리(개발도구)를 무지성으로 사용 (구글링으로 사용법만 공부)
    - 비슷한 개념을 계속해서 구글링하는 경향이 있음(시간낭비)
- 게임 클라이언트와 통신 해본 적 없음
- 테스트 코드 작성 경험 거의 없음
- 서비스 배포 경험 없음
- 구현 후 성능 개선 경험 없음
- 형상 관리 시스템(git) 제대로 사용 못함

두달 뒤의 나의 상태

- 라이브러리(개발도구)의 공식 레퍼런스를 보고 사용법을 스스로 익힐 수 있음
    - django channels
    - Redis
    - RabbitMQ
    - Celery
    - Coverage.py
    - Git
    - Ngrinder
    - Apache JMeter
- 습관적으로 단위 테스트 코드 작성
    - 기능 별로 테스트 목적을 문서화

현재 개개인의 상태(조현민)

개발 관련 공부 2년전쯤에 시작

게임 프로젝트 경험 1회 (동아리, 2D 리듬게임)
- Unity 사용. 서버 및 DB 사용X.
- 하드코딩으로 때려넣은 부분이 꽤나 많음

iOS 프로젝트 경험 5회

코드 작성시에 어떻게든 작동만 하면 된다는 마인드였지만 이제 바꾸고 싶음.

서버에 대한 지식이 얕음.
유니티도 프로젝트한지 오래돼서 감 찾는중.

게임 개발 프로젝트를 더 많이 경험하고 싶음.

두달 뒤의 나의 상태

Photon PUN2를 사용해 실시간 멀티플레이 게임을 만들고 만든 내용을 설명할 수 있다.

3D 게임에 셰이더와 이펙트를 적용시킬 수 있다.

스타일 가이드에 맞는 코드를 작성할 수 있고, 재사용 가능한 코드를 작성할 수 있다.

현재 개개인의 상태(구자현)

- 게임 서버를 개발해본 적 없음 (client 경험도 없음)
    - 게임 서버나 게임 클라이언트와 연결할 때 고려해야 할 점도 잘 모르겠음
- socket은 채팅을 구현해보는 정도 -> 게임에서 적용해본 적 없음
- MSA는 Hello World 수준. 실제 서비스 배포 경험 없음
- 쿼리, 애플리케이션 로직에 대한 성능 개선 경험은 있지만 왜 이 성능을 기준으로 잡았냐 하면 할말이 없음 (성능 시나리오 작성 경험이 없음)
- 코드를 작성하다 나중에 보면 의도를 기억하지 못함

PMP 마무리 짓기 위해서 해야할 것들

팀 현 상태 파악, 개인 현 상태 파악, 프로젝트라는 것의 본질 파악,
개인 목표 팀 목표 수정, 회의할때 준비해야할것들?

개인목표

현명은

- 구현된 결과물의 드로우콜의 수를 20%이상 낮추기
- 성능이 평등하게 나오도록 유지하기

조현민

- C# 스타일 가이드에 맞춘 코드 작성
- 재사용 가능한 코드를 위해 디자인 패턴 활용
- 실시간 통신 구현

박동진

- 최대 200명의 실시간 요청을 동시에 처리
- 서버의 병목 현상을 개선하여 성능 향상
- 테스트 가능한 것들은 모두 테스트 코드 작성

구자현

- 견고한 아키텍처와 성능을 고려해 코드 작성하기
- 성능 프로파일링하기
- 왜? 라고 물으면 답할 줄 알기

개인목표 달성 후 모습?

현명은

- 유니티에서 비동기프로그래밍 방식으로 코드를 구현할 수 있는 개발자
- 코드리뷰를 두려워하지 않는 개발자

조현민

- NoPOKER의 게임 네트워크에 대해 설명할 수 있는 사람
- 재사용 가능한 코드를 작성하려는 습관이 있는 개발자

박동진

- 개발도구의 공식 레퍼런스를 보고 사용법을 스스로 익힐 수 있는 사람
- 습관적으로 단위 테스트 코드를 작성하는 사람

구자현

- 성능을 측정하고 개선이 필요한 부분을 짚어내 개선할 줄 아는 개발자
- 코드에 나의 의도가 명확하게 드러나게 작성하는 개발자

How to work (현명은)

설계하기 쉬운 정적인 프로그램 코드만 작성하여 프로그램 자원을 효율적으로 이용하지 못하고
사용자에게 실시간으로 응답할 수 있는 프로그램을 만들지 못하여 항상 아쉬웠음.

목표

유니티에서 비동기프로그래밍 방식으로 코드를 구현하고 설명할 수 있는 개발자가 되기

계획

- HTTP통신을 위한 방식인 UniTask, IMultipartFormSection , WWWForm (레거시 함수) 등의 설명과 장단점을 비교하고 선택한 이유에 대해서 문서화하기
- 클라이언트와 서버가 통신하는 부분을 비동기코드로 작성하기

점검

- HTTP통신코드를 작성 한 후 제대로 작동하는지 확인
- 본인이 짠 통신 코드에 대해서 본인이 설명할 수 있는지 확인

How to work (현명은)

다른사람이 나의 코드를 보는것이 두려움

목표

코드리뷰를 두려워하지 않는 개발자 ? 코드리뷰를 즐기는 개발자? 가 되기

계획

- PR을 하고나서 같은직군인 팀원에게 코드리뷰를 부탁하기
- 같은직군 팀원과 네트워크에 대한 공부를 진행하기로 했음,
  이를 위해 네트워크 관련코드를 작성하고 난 다음에 팀원에게 해당 코드를 설명하기

점검

- ..
-

개인목표를 위한 계획 (박동진)

목표

개발도구의 공식 레퍼런스를 보고 사용법을 스스로 익힐 수 있음

계획

1. 공식 레퍼런스만 보고 구현
2. 어려움이 있으면 구글링, 팀원들에게 질문, 멘토님에게 질문
3. **어려운 부분**이 무엇이었는지 **경험학습모델**에 의거해 문서화

   경험학습모델 예시)

   구체적 경험 : django channels의 consumers 개념을 이해하는데 어려움을 겪어 구글링을 통해 알아냄

   성찰 : 공식 레퍼런스의 영어 문장을 잘 이해 못함, ASGI 개념을 잘 이해 못함

   개념화 : 비동기 개념을 확실히 이해하고 **나만의 언어로 문서화**

   시행 : 비동기 기반 개발도구(RabbitMQ, Celery)들의 공식 레퍼런스를 이해하는데 문서화한 개념들을 사용

점검

새로운 레퍼런스를 볼 때, 나만의 언어로 문서화한 개념들이 도움 되는지 확인

비슷한 개념을 다시 구글링, 질문해야 했다면 문서화한 개념을 수정

사용할 개발도구들:

Django, Django channels, Redis, RabbitMQ, Celery, Postman, Coverage.py, Git, Ngrinder, Apache JMeter

개인목표를 위한 계획 (박동진)

목표

습관적으로 단위 테스트 코드 작성

계획

- model, view, form, method, function 등 기능을 가지는 모든 것에 테스트 코드 작성
  - 각 테스트 코드의 목적을 문서화
- coverage.py를 활용해 프로젝트의 테스트 커버리지를 파악
  - 테스트 커버리지를 낮추는 어떤 커밋도 하지 않음
  - 테스트 커버리지 100%를 목표

점검

매일 coverage.py 로 테스트 커버리지를 점검
테스트 커버리지를 향상하는데 어려움이 있을 경우, 해당 테스트 코드의 목적을 점검

How to work (조현민)

목표	C# 스타일 가이드에 맞춘 코드 작성
계획	C# 스타일 가이드 정독
점검	주 1회 가이드에 맞는 코드를 작성하고 있는지 확인 현재 구현정도와 수행 계획이 일정과 비교했을 때 빠듯한지 점검

How to work (조현민)

목표

재사용 가능한 코드를 위해 디자인 패턴을 활용하기

계획

유니티로 배우는 게임 디자인 패턴 2판을 정독

기획서를 읽으면서 구현해야 할 코드 부분의 디자인 패턴을 설계

점검

주 1회 코드 재사용이 잘 되고 있는지 확인

재사용 이전에 구현 단계에서 어려움을 느낀다면 계획 변경

현재 구현정도와 수행 계획이 일정과 비교했을 때 빠듯한지 점검

How to work (구자현)

목표

견고한 아키텍처와 성능을 고려해 코드 작성하기

계획

- MSA 정확히 짚고 넘어가기
  - 서버를 나누는 기준인 바운디드 컨텍스트 정확하게 알아보고 선정하기
  - 서버들을 연결할 때 도구는 어떤 것을 사용할 것인지? 장단점 비교해보기
  - 각 서버들의 Health-Check과 FailOver 방법은 어떤 방식이 있는지 분석해 적용하기
- 성능을 고려해 코드 작성하기
  - 각종 테스트 도구 선정 (nGrinder, artillery, k6) 및 유스케이스 별 테스트 실행
  - 최적화 방안 탐색 (slow query, caching, algorithm, shading) 기록 및 적용
  - 얼마나 개선이 되었는지와 이를 해결하기 위해 어떤 비용이 들었는지를 기록해 Trade-Off 비교하기

점검

매주 코드 리뷰하기

단, 나 혼자 검증하지 말기 -> 적어도 팀원들에게, 여유가 있으면 멘토님들에게 검증받기

How to work (구자현)

목표

왜? 라고 물으면 답을 할 줄 아는 능력 기르기

계획 및 점검

내가 왜 이러한 생각을 가지고 코드를 짰는지 기록하기

토요일마다 일주일간 작성한 글을 정독하며 회고하기.

팀 회고 때 해당 글들을 보여주면서 설명하고 검증하기.

Milestone

1. 서비스 구현 및 서버 연결 (~ 2월 중순)

인증 서버 구현
회원가입/로그인화면 구현

채팅 서버 구현
로비, 대기방 화면 구현

게임 서버 구현
2D 포커게임 구현

실시간 게임 서버 구현
3D 액션게임 구현

2. 성능 개선 및 기능 추가
   - 성능 프로파일링 및 개선 작업

개발 계획 - Client

현명은

- 로비
- 대기방
- 2D 포커게임

~ 1.22 : **화면설계 및 구현**

1. 로비, 대기실 화면 구현
2. 2D 포커게임 화면 구현
3. UI상호작용 구현

~ 1.30 : **클라이언트 백엔드 연결**

1. 로비 및 대기실 백엔드 연결

~ 2.7 : **게임기능 구현 및 연결작업**

1. 2D게임 구현
2. 2D게임 백엔드 연결
3. 2D게임 3D게임 전환

~ 2월중순 : **성능개선, 프로파일링 및 개선작업**

조현민

- 로그인/회원가입
- 실시간 서버
- 3D 액션게임

~ 1.16 : **화면설계 및 구현**

1. 회원가입/ 로그인 화면 구현
2. UI상호작용 구현

~ 1.27 : **클라이언트 백엔드 연결**

1. 회원가입/ 로그인 백엔드 연결
2. 플레이어 정보 연결 및 로비 전달

~ 2.13 : **게임기능 구현 및 실시간 서버 구축**

1. 3D게임 구현
2. 실시간 서버 구축
3. 3D게임 백엔드 연결

~ 2월중순 : **성능개선, 프로파일링 및 개선작업**

개발 계획 - Server

박동진

- 인증 서버
- 채팅 서버

~ 1.27 : **인증 서버 구현**
- -로그인, 회원가입
- -아이디 찾기, 비밀번호 찾기
- -이메일 인증
- -회원 탈퇴
- -관리자 로그인
- -관리자의 유저 관리 API
- -관리자의 채팅 로그 API

~ 2.10 : **채팅 서버 구현**
- -채팅 목록 API
- -채팅 입력 API

~ 2월중순 : **성능개선, 프로파일링 및 개선작업**

구자현

- 로비 서버
- 2D게임 서버

~ 1.27 : **로비 api 구현**
- - 게임방 생성, 목록, 참여 기능
- - 게임 시작 기능
- - 게임방 채팅 환경 구성

~ 2.7 : **2D 게임 서버 구현**

~ 2월중순 : **성능개선, 프로파일링 및 개선작업**

End Of Document