1. 스터디 방향성

텍스트, 도표, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

3계층 구조(3 Tier Architecture)의 프레젠테이션 계층(클라이언트), 로직 계층(서버), 데이터 계층(DB)를 모두 개발하는 것을 목표로 한다. 클라이언트는 Unity (2022.3.43f1 LTS 버전)를 활용하여 C#으로 방치형 게임을 개발한다. 서버는 Node.js (20.17.0)를 이용하여 개발하며, DB는 MongoDB와 Redis를 사용한다. JWT, 소켓 통신 등 최대한 많은 부분을 학습하는 것이 목표이다.

1. Git 시작하기 (별도 문서 참고)
2. 유니티 시작하기

* 유니티 허브 설치 파일 다운로드
* 유니티 허브 설치 및 실행
* 유니티 계정 생성 및 로그인
* 유니티 아카이브에서 LTS(Long Term Support, 장기 지원 버전)버전 중 최신 버전 다운로드
* 설치 과정 중 Visual Studio Community 2022를 함께 설치
* VS Community 설치 과정 중 워크로드에서 유니티 개발 환경을 설치
* 유니티 설치 완료 후 2D 프로젝트 생성
* 좌측 계층 목록 창에서 우클릭 메뉴 – UI – Legacy – Text / Button 를 클릭하여 오브젝트 생성
* 하단 프로젝트 창에서 우클릭 메뉴 – Create – 폴더 생성 및 C# 스크립트 생성
* 스크립트에서 UnityEngine.UI 네임스페이스를 가져오도록 using 구문을 추가
* Debug.Log, .text 등 필요한 기능을 넣어 OnClickBtn이라는 함수를 완성
* 스크립트 저장 및 연결
* 씬 저장 및 프로젝트 실행

1. 서버 개발 시작하기

* Node.js 공식 홈페이지에서 node.js를 다운로드 받아 설치
* 터미널이나 git bash를 이용, 원하는 작업 폴더로 이동하여 npm init 커맨드 입력
* 계속 엔터를 입력하여 진행
* node -v 명령어와 npm -v 명령어로 node.js와 패키지 매니저의 버전을 확인
* Code . 은 Visual Studio Code를 실행하는 명령어
* vi는 편집기를 호출하는 명령어 (없는 파일을 편집하고 저장하면 해당 파일이 생성됨)
* ll은 현재 위치의 파일 목록을 보여주는 명령어
* rm은 파일을 삭제하는 명령어
* 입력 창에서 파일 이름을 자동완성하고 싶다면 탭을 입력
* Node.js를 이용한 서버를 편리하게 구현할 수 있게 해주는 프레임워크 ‘express’를 설치해야함
* npm install express --save 입력하여 express 설치
* cat package.json 명령어로 package.json 파일 내용을 출력. 의존성에 express가 추가되었는지 확인.
* vi index.js 입력. i를 눌러 INSERT 모드로 진입한 뒤, index.js 파일에 예제 코드를 입력. (구글에 express helloworld example code 검색하여 나온 예제를 복사하여 입력)

(<https://expressjs.com/en/starter/hello-world.html>)

* ESC를 눌러 커맨드 모드로 진입, :wq를 입력하여 파일 저장 후 나오기
* node index.js 를 입력하여 서버를 띄움
* 웹 브라우저를 실행하여 127.0.0.1:3000 또는 localhost:3000 주소로 접속
* Hello World! 텍스트 확인
* Node.js mime-types (웹에서의 파일 포맷(확장자)이 정리되어 있는데 그것이 mime-types)
* 공유기는 public한 아이피를 받아다가 가상의 ip를 뿌려주는 것. 인터넷 사용료 = public ip를 할당 받는 비용
* 기본 게이트웨이는 공유기의 주소. (공유기 네트워크 설정 시 게이트웨이 주소로 접속한다)
* ipconfig 시 ipv4 주소가 나의 ip주소임. 공유기가 뿌려줄 경우 가상 ip가 되겠음.
* whois ip 조회를 통해 ip주소를 할당받은 사업자를 알 수 있음.
* ping naver.com으로 네이버 ip주소를 알 수 있음.
* 결국 ip에는 공인 ip & 사설 ip 가 있는 것.
* ip는 서버나 컴퓨터의 주소다.
* 소프트웨어(서비스)별로 나뉘어 붙을 수 있는 개념이 포트다.
* https 주소로 접근하는 것 = 443포트
* 서버에 접근하기 위해서는 도메인, 주소, 포트가 필요. ip와 주소만 가지고 서버에 붙을건데 이를 쉽게 해주는 것이 express.
* 127.0.0.1 = localhost

(C:\Windows\System32\drivers\etc 의 hosts 파일에 명시되어있음.)

* 도메인, 주소, 포트, path(경로) 등의 정보를 엔드포인트라 부른다(endpoint)
* Json = JavaScript Object Notation의 축약어로 데이터를 저장하거나 전송할 때 많이 사용되는 경량의 텍스트 기반 DATA 교환 형식 (cf. xml 파일)

1. 과제

* redis와 MongoDB 연결해보기
* 유니티를 이용하여 계산기 만들기 (사칙연산 순서 상관없음)