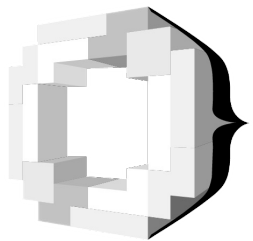


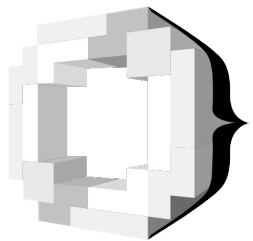
# 2023 유니티 중급반 1주차

오파츠 12기 여정인, 유태환



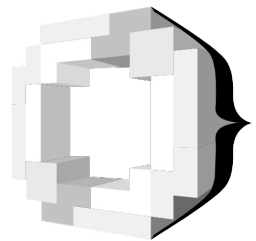
## 목차

- 게임은 어떻게 만들어야 할까요?
- 캐릭터나 UI 버튼 같은 건 직접 그려야 될까요?
- 다른 사람과 한 프로젝트에서 같이 일하려면 어떻게 해야 할까요?



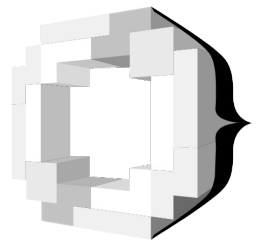
# 목차

- **게임 기획하기**
  - 게임 기획 순서
- **에셋 구하기**
  - 에셋이란?
  - 에셋 조달 방법들
- **협업하기**
  - 버전 컨트롤 시스템이란?
  - git & github
  - Plastic SCM



# 게임 기획을 대충 해봅시다

- 장르와 주제 선정
- 게임의 룰
- 유저 플로우 (씬 구성)
- UI 배치
- 최종 검토



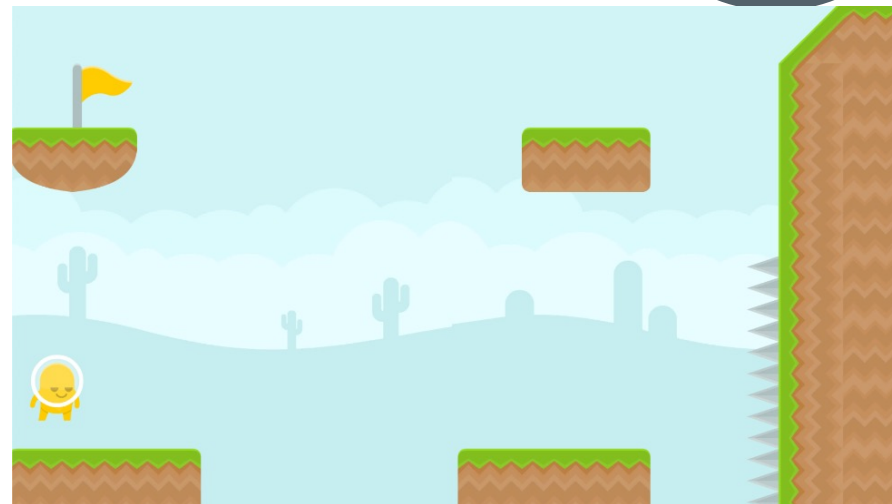
# 게임 기획을 대충 해봅시다

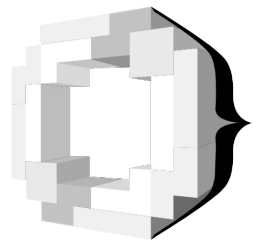
## ■ 장르와 주제 선정

- 게임의 타겟 플랫폼(모바일, PC 등)과 장르
- 컨셉, 주제 등을 생각해봅시다.

이 수업에서는...

- 플랫폼머 게임
- 플레이어가 깃발에 최대한 빨리 도착하는걸 목표로 하자!



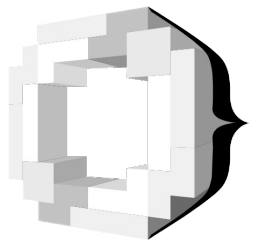


# 게임 기획을 대충 해봅시다

## ■ 플랫폼어 게임이란...



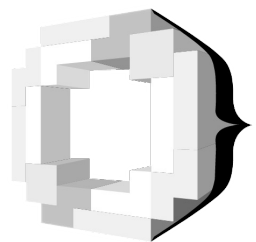
- 발판(플랫폼)에서
- 이동, 점프하면서 목표를 달성하는 게임 (조작 자체에서 재미를 느낀다!)



# 게임 기획을 대충 해봅시다

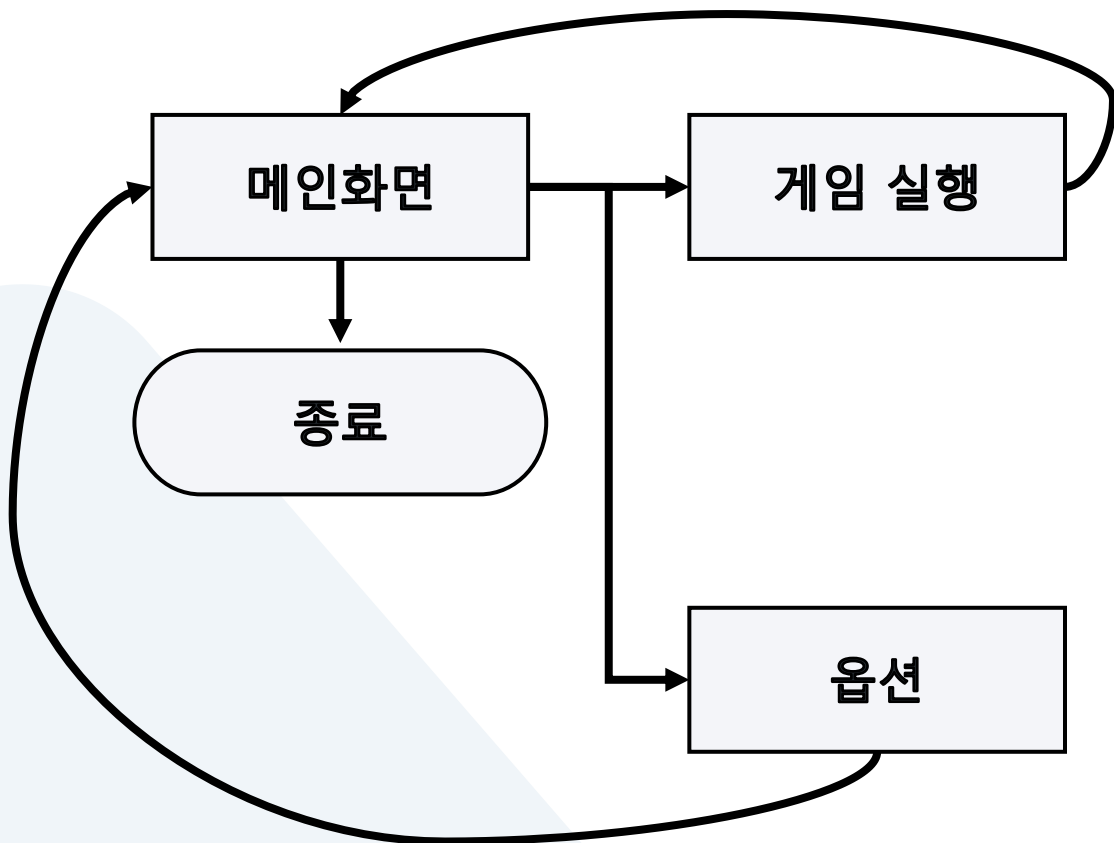
## ■ 게임의 룰을 생각해봅시다

- 플레이어는 플랫폼 위에서 이동한다.
- 좌, 우, 점프 이동이 가능하다.
- 점프는 2단 점프까지 가능하다.
- 점수는 최대 점수에서 시작해서 시간이 지날수록 감소한다.
- 점수가 0보다 낮아지거나 장애물에 부딪히면 게임이 종료된다.
- 점수가 0이 되기 전에 깃발에 도착하면 플레이어의 승리

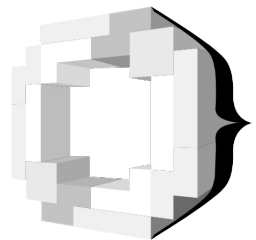


# 게임 기획을 대충 해봅시다

## ■ 유저 플로우 + UI







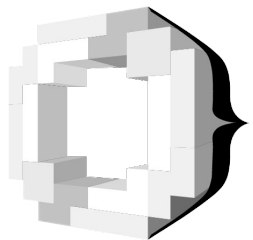
# 게임 기획을 대충 해봅시다

## ■ UI 구성



버튼

텍스트



# 게임 기획을 대충 해봅시다

## ■ UI 화면

사운드 조절 (슬라이더)

해상도 선택 (드롭다운)

전체화면 (체크박스)

취소

확인

OPTIONS

SOUND LEVEL

10

SCREEN RESOLUTION

1920 X 1080

1920 X 1080

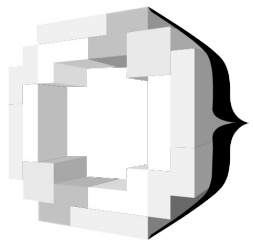
SCREEN MODE

☒ FULL SCREEN

FULL SCREEN

CANCEL

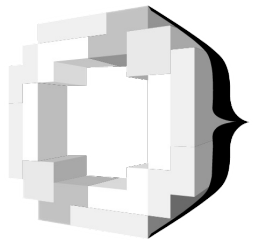
ACCEPT



# 게임 기획을 대충 해봅시다

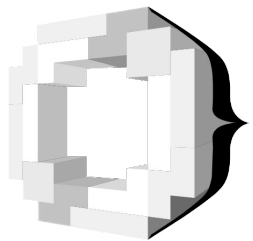
## ■ 체크해보기

- 어디서 재미를 느낄 수 있는가?
- 주제를 잘 표현 했는가?
- 점수 부여 등 밸런싱



# 에셋이란?

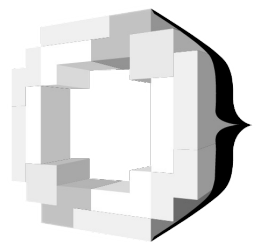
- **게임 개발에 활용할 수 있는 다양한 리소스!**
  - Ex) 3D 모델링, 2D 스프라이트, 오디오, 애니메이션, 스크립트, 플러그인
- **바퀴를 재발명하지 않아도 된다!**



# 에셋을 구해봅시다

## ■ 직접 만들기

- 아트팀과 협업하자!
- 수제 = 커스터마이징
- 이미 구현되어 있다면 라이브러리를 사용하는 것이 보통 더 안정적임
- 나 혼자만 이해하고 있는 상태라면 팀원들이 활용할 때 설명을 해주어야 함



# 어디서 에셋을 구할까요

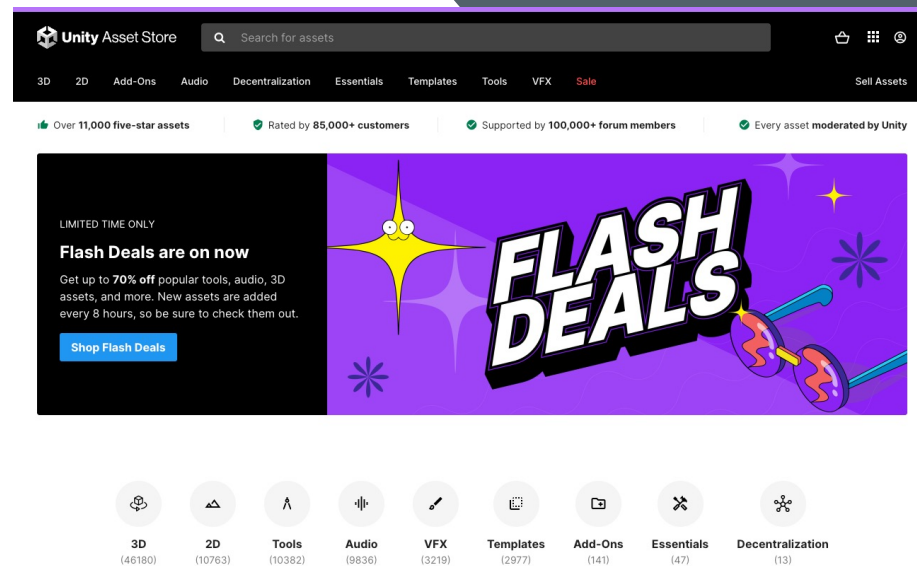
## ■ 유니티 에셋 스토어

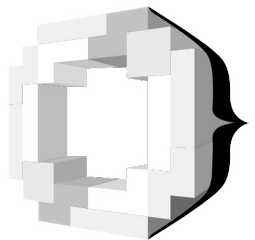
- 방대한 양의 에셋들
- 에디터 내에서 불러오기 쉬움
- 하지만 퀄리티가 보장되어있지 않다

(너무 많은 중복된 에셋들과 유지보수가 멈춰버린 에셋들...)

- 언리얼 마켓 플레이스와 비교하면 초라해지는 양과 퀄리티 (특히 3D!)

<https://assetstore.unity.com/>

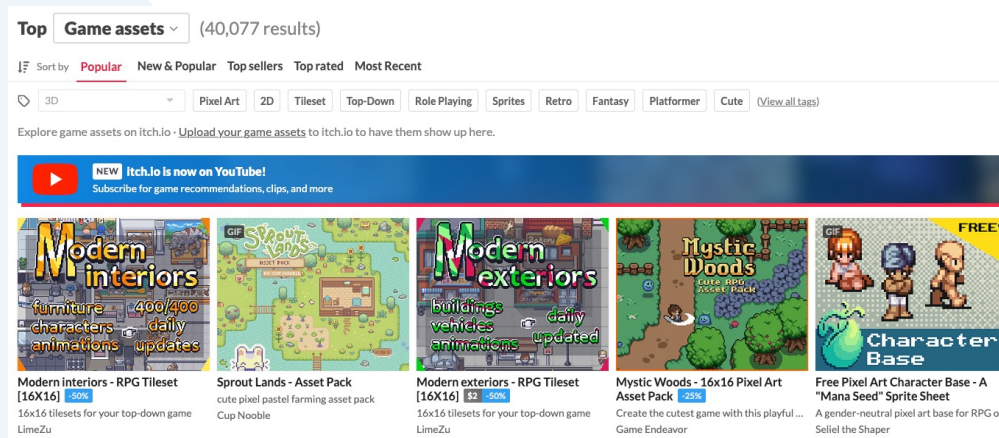




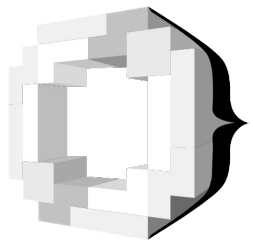
# 어디서 에셋을 구할까요

## ■ itch.io

- 양질의 2D 에셋들
- 저렴한 가격에 오디오, UI, 2D 스프라이트 등등 다양한 에셋을 구할 수 있다
- 3D 에셋은 많지 않고 주로 voxel이나 low poly가 많다



<https://itch.io/game-assets>

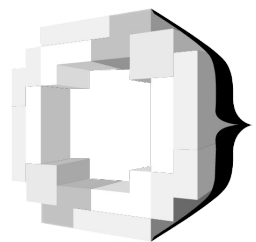


# 어디서 에셋을 구할까요

## ■ 기타 등등

- 구글에 검색해보면 다양한 3D 모델링을 모아놓은 곳이 많다.
- 언리얼 마켓플레이스
  - 굉장히 양질의 3D 무료 에셋이 많고, 주기적으로 무료로 에셋을 나눠주고 있음
  - 언리얼에서 내보내서 유니티에 불러와서 사용해도 되기는 한다
- 라이선스를 잘 확인하고 사용할 것!



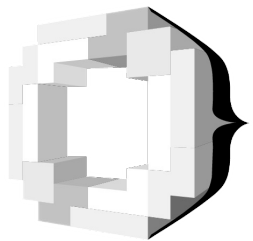


# 저희 수업에서는...

## ■ kenny

- 로우폴리, 2D, 폰트, 오디오, 이펙트 등 정말 다양한 에셋들
- License가 CC0 -> 자유로운 사용 가능
- <https://www.kenney.nl/assets>

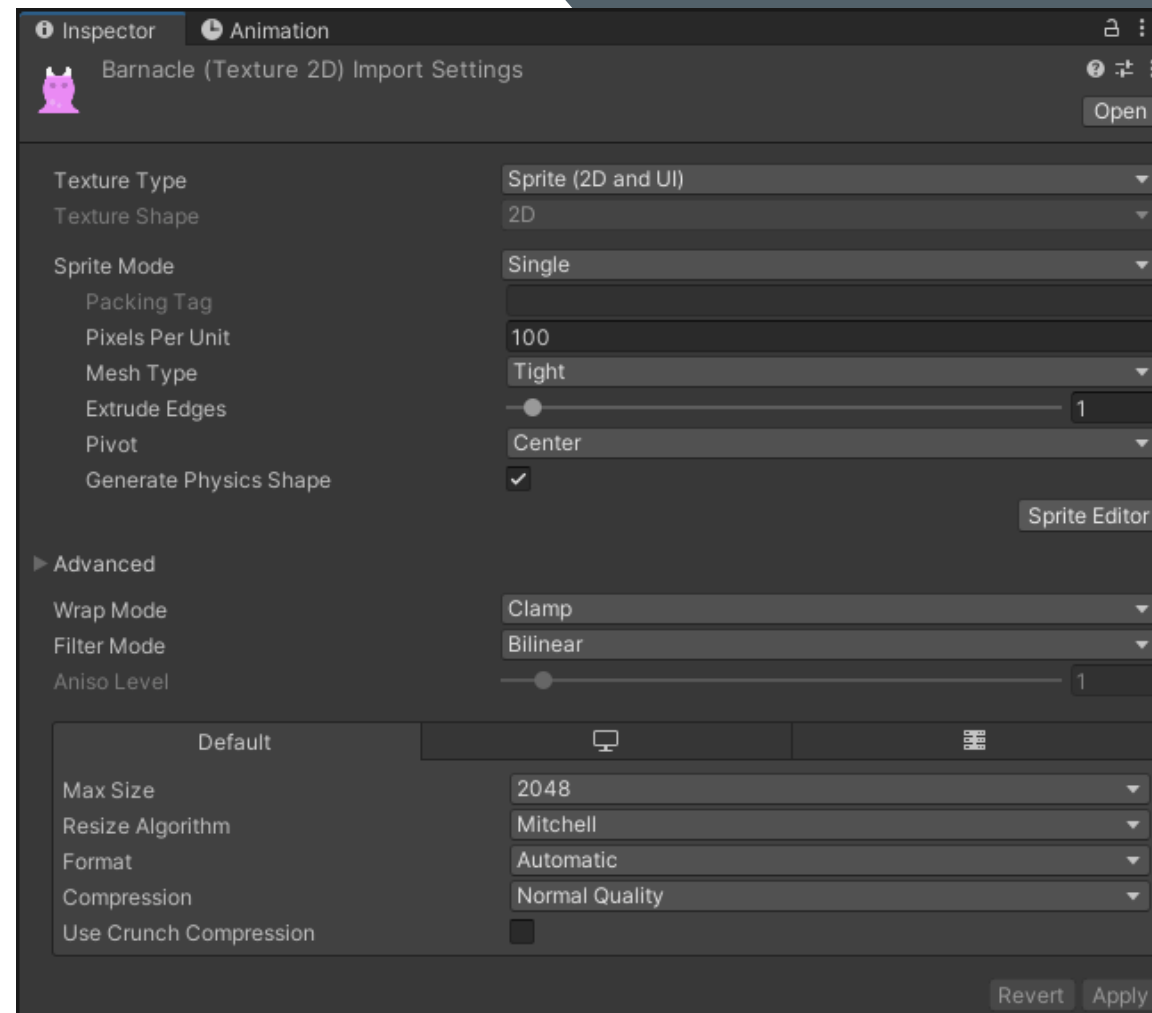


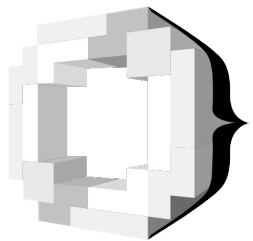


# 2D 에셋 불러오기

파일을 유니티로 불러와서 Inspector에서 세팅을 해줍니다

- **Texture Type : Sprite**
- **Sprite Mode :**
  - 단일 파일에 하나의 스프라이트 -> Single
  - 단일 파일에 여러 스프라이트 -> Multiple
- **Pixels Per Unit:**
  - Unit은 유니티의 좌표 단위
  - 16 \* 16 pixel 그림이라면 16이라고 설정했을 때 유니티 좌표 한 칸에 캐릭터가 딱 맞게 들어간다
- 자세한 것은 다음 수업 때 알아보시다.





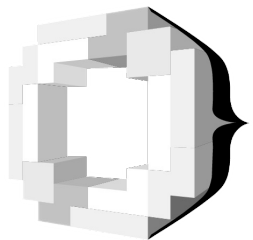
# 버전 컨트롤 시스템 VCS

## ■ 버전 컨트롤 시스템이란

- 파일들의 버전과 변화를 기록하는 시스템
- 원하는 시점의 버전을 다시 불러올 수 있다!

## ■ 사용하는 이유

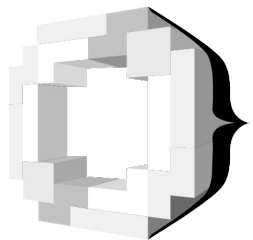
- 파일, 또는 프로젝트 전체를 되돌릴 수 있다! -> 실수에 대한 대처 유연화
- 버전 기록을 자세히 한다면 누가 어디서 작업한 부분인지 확인 가능



# 버전 컨트롤 시스템 VCS

## ■ 버전 컨트롤을 직접 하려면...

- 유니티\_중급반\_OT 수수정보.pptx
- 유니티\_중급반\_OT 수정본.pptx
- 유니티\_중급반\_OT.pptx



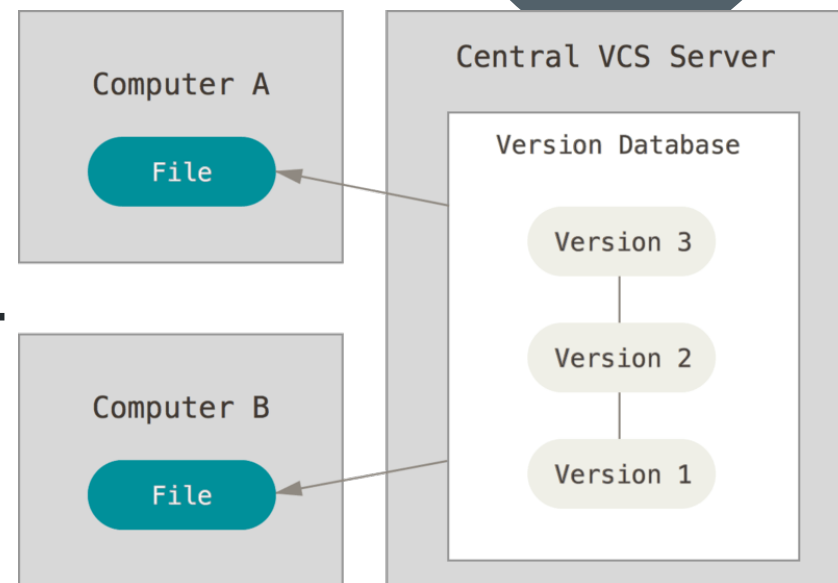
# 버전 컨트롤 시스템 VCS

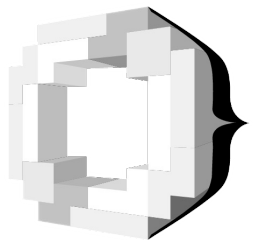
## ■ 중앙 버전 컨트롤 시스템 (CVCS)

- 서버에 저장소를 두고 사용자들은 서버에서 파일을 받아 사용
- 사용자가 변경한 내용이 서버에 바로 반영됨
- 관리자가 쉽게 저장소의 관리를 하기가 쉬움
- 중앙 서버에 문제가 발생하면 모두에게 문제가 발생함

만약 이상한 코드가 바로 반영된다면...?

다른 사람의 작업물과 내 작업물이 같은 파일명...?

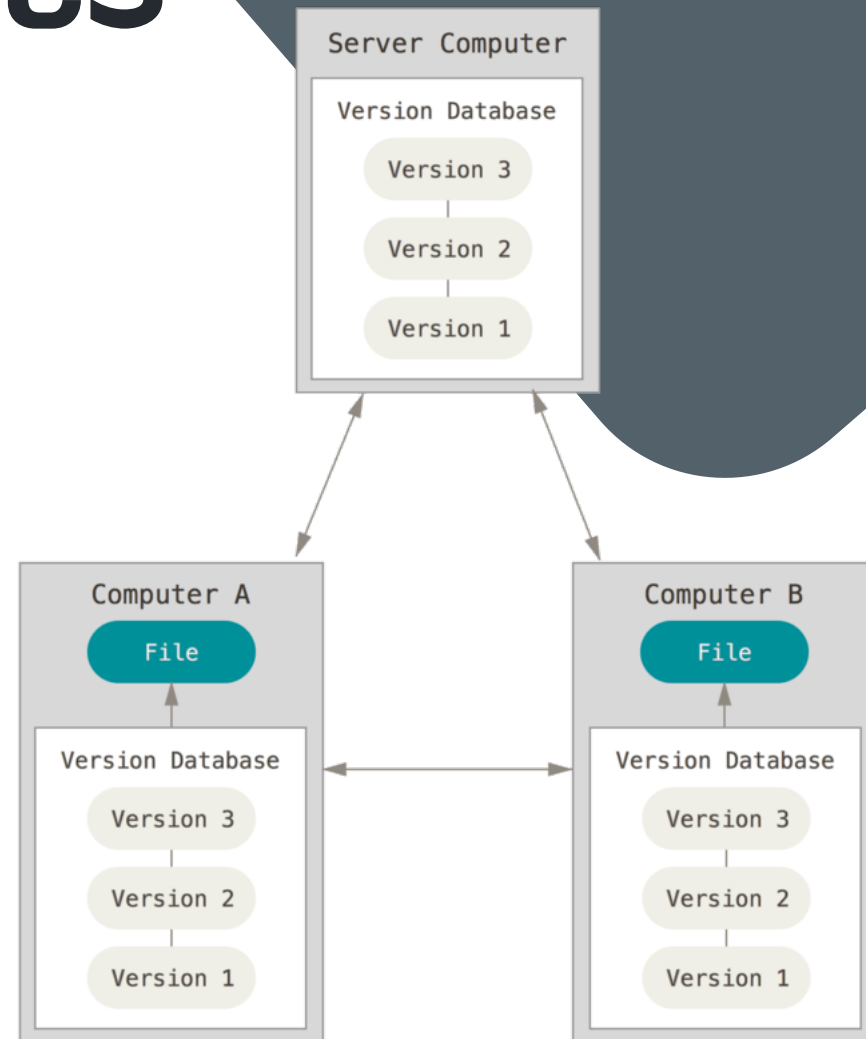


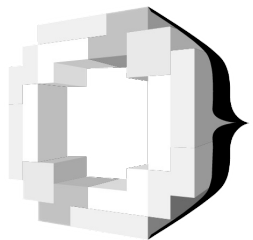


# 버전 컨트롤 시스템 VCS

## ■ 분산 버전 컨트롤 시스템 (DVCS)

- 각각의 사용자가 리모트 저장소를 가지고 있음
- 리모트 저장소에는 파일 뿐 아니라 히스토리까지!
- 서버에 문제가 생겨도 복원 가능
- 다른 사람의 작업물과 충돌하는 문제를 덜 신경 쓰고 작업 가능 (중앙식에 비해)

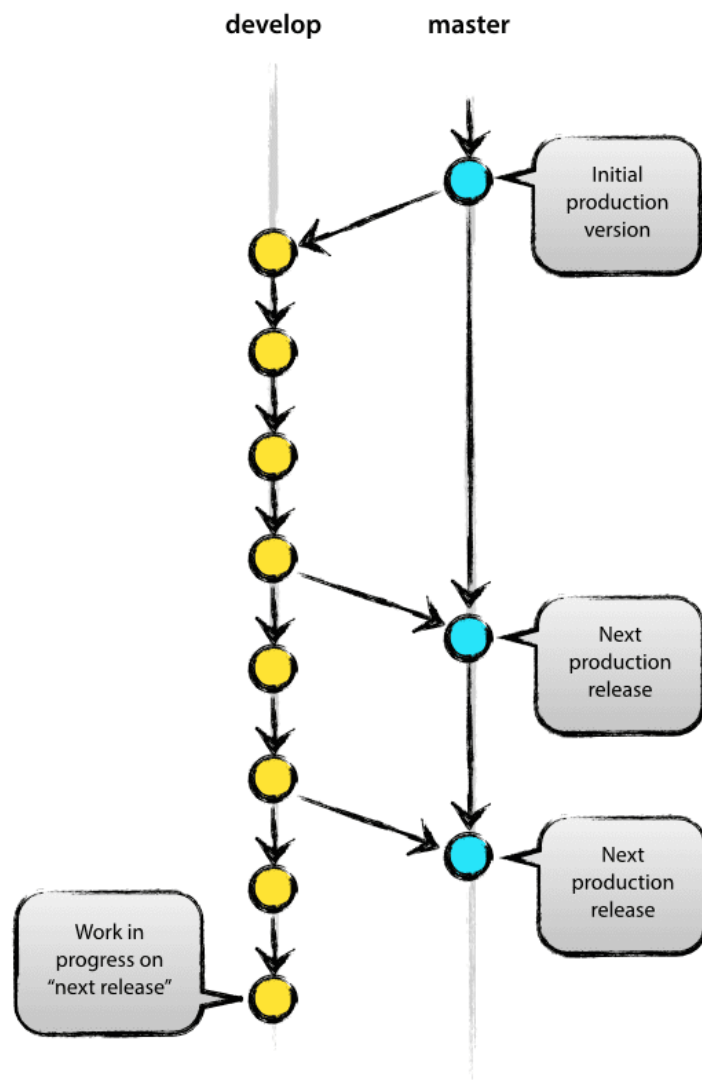


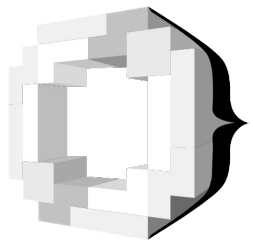


# 버전 컨트롤 시스템 VCS

## ■ Git + Github로 버전 관리

- Git: 대표적인 분산 버전 컨트롤 시스템
- Github: 대표적인 git 저장소 (서버)
- 유니티 프로젝트 파일 크기가 커서  
git-lfs라는 시스템을 사용해야 함
- <https://github.com/>



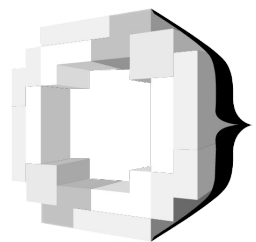


# 버전 컨트롤 시스템 VCS

## ▪ Plastic SCM으로 버전 관리

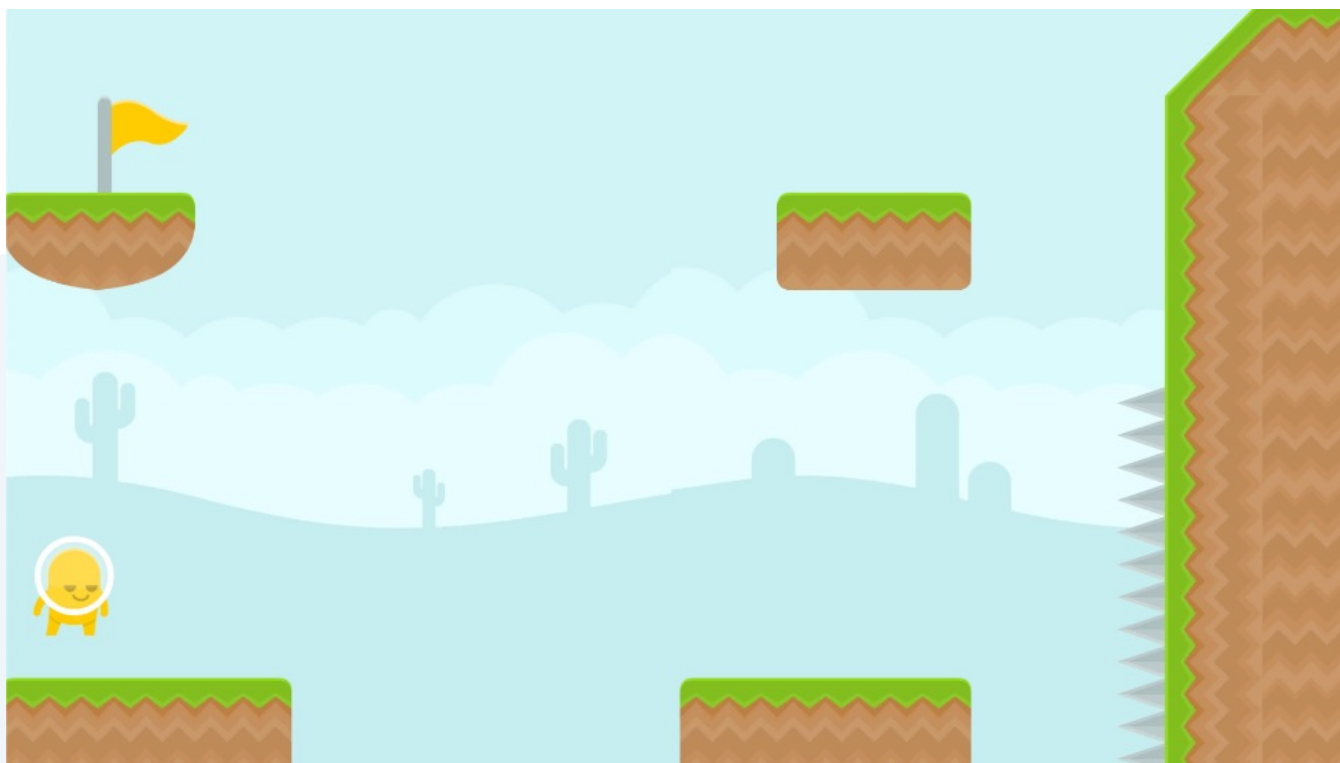
- Plastic SCM 유니티에서 2020년에 인수한 버전 컨트롤 시스템
- 기존 유니티 VCS였던 콜라보레이트보다 훨씬 낫지만 유료 (3명, 5GB까지 무료)
- 러닝 커브가 심한 것과 달리 한 저장소에 프로그래머를 위한 Plastic, 아티스트를 위한 Gluon을 따로 두어 각자 편한 방식으로 작업 가능!
- 중앙 방식과 분산 방식 모두 사용 가능
- 게임 제작을 목적으로 디자인되어 Ifs같이 복잡한 설정 불필요
- 언리얼에서도 사용 가능!
- <https://unity.com/kr/support-services/redeem/plastic-scm#2-create-workspace-and-repo>

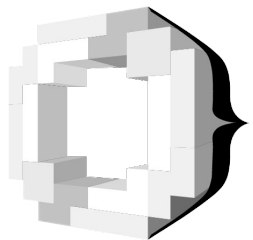




# 다음 수업에서는

- 게임 화면을 만들어보겠습니다





## 다음 수업에서는

- 유니티를 미리 설치해주세요

- 실시간으로 직접 게임을 만들지는 않겠지만 기능들을 바로바로 확인하려면 직접 설치해서 확인하는 것이 좋습니다.

- <https://unity.com/kr/products/unity-student>

- 코드 에디터는 Visual Studio를 추천합니다.

- 맥북 사용자라면 <https://visualstudio.microsoft.com/ko/vs/mac/unity/>

- 유니티 버전은 2021.3.x(LTS) 버전을 사용합니다 (아마도 2021.3.20f1)