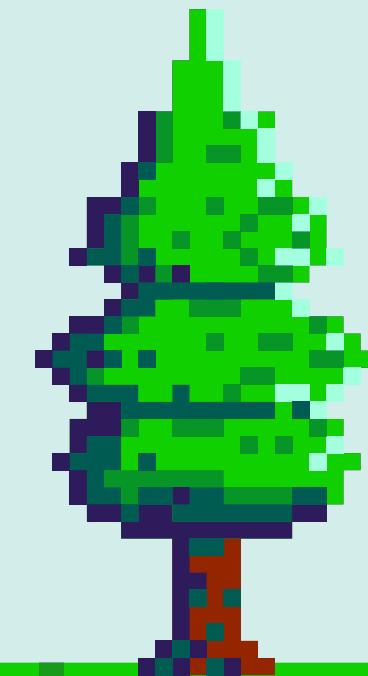
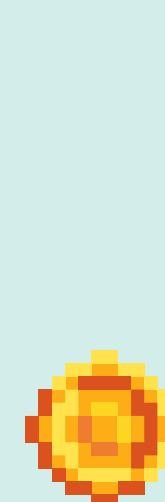


JOIN US AT CHEADLE LIBRARY

23 ASHFIELD RD, CHEADLE SK8 1BB, MANCHESTER, UK

4TH NOVEMBER 2024 ⏰ 18:00 – 20:00

# LET'S BUILD VIDEO GAMES!



LIMITED SPOTS – RSVP TODAY!



# LET'S BUILD VIDEO GAMES!

1

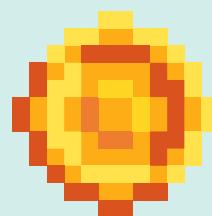
Welcome & Coffee  
18:00 – 18:15

2

Game Dev Techniques  
By your host: [Spark Games](#)  
18:15 – 19:15 

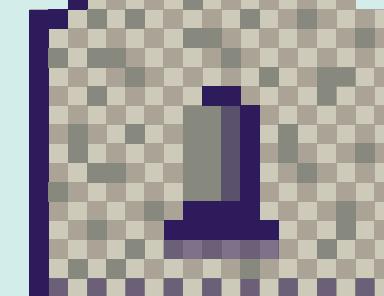
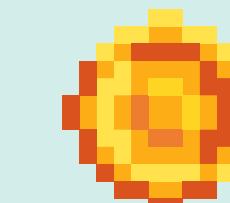
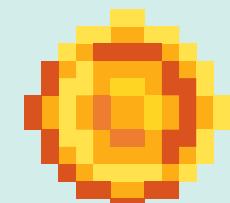
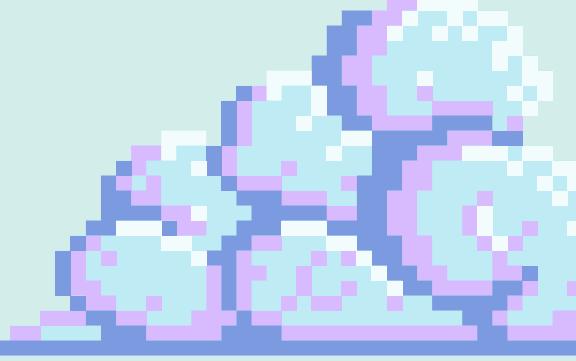
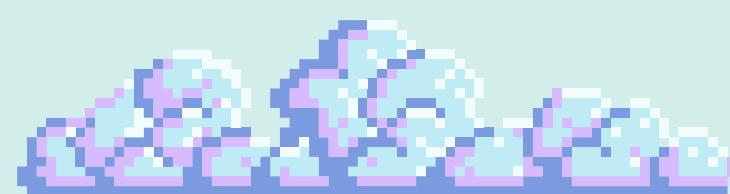
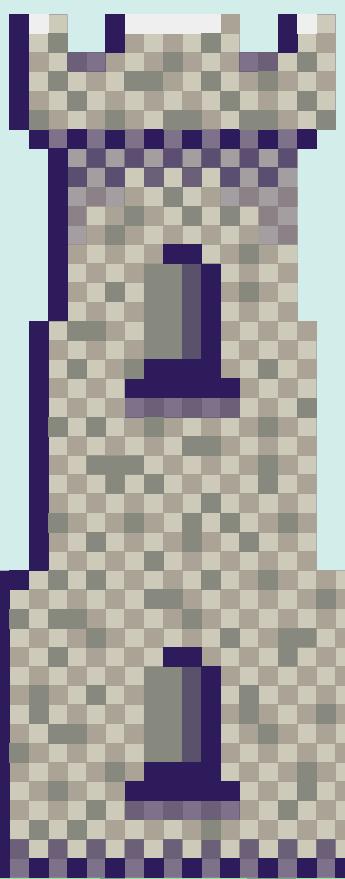
3

Open Discussion  
19:15 – 19:45

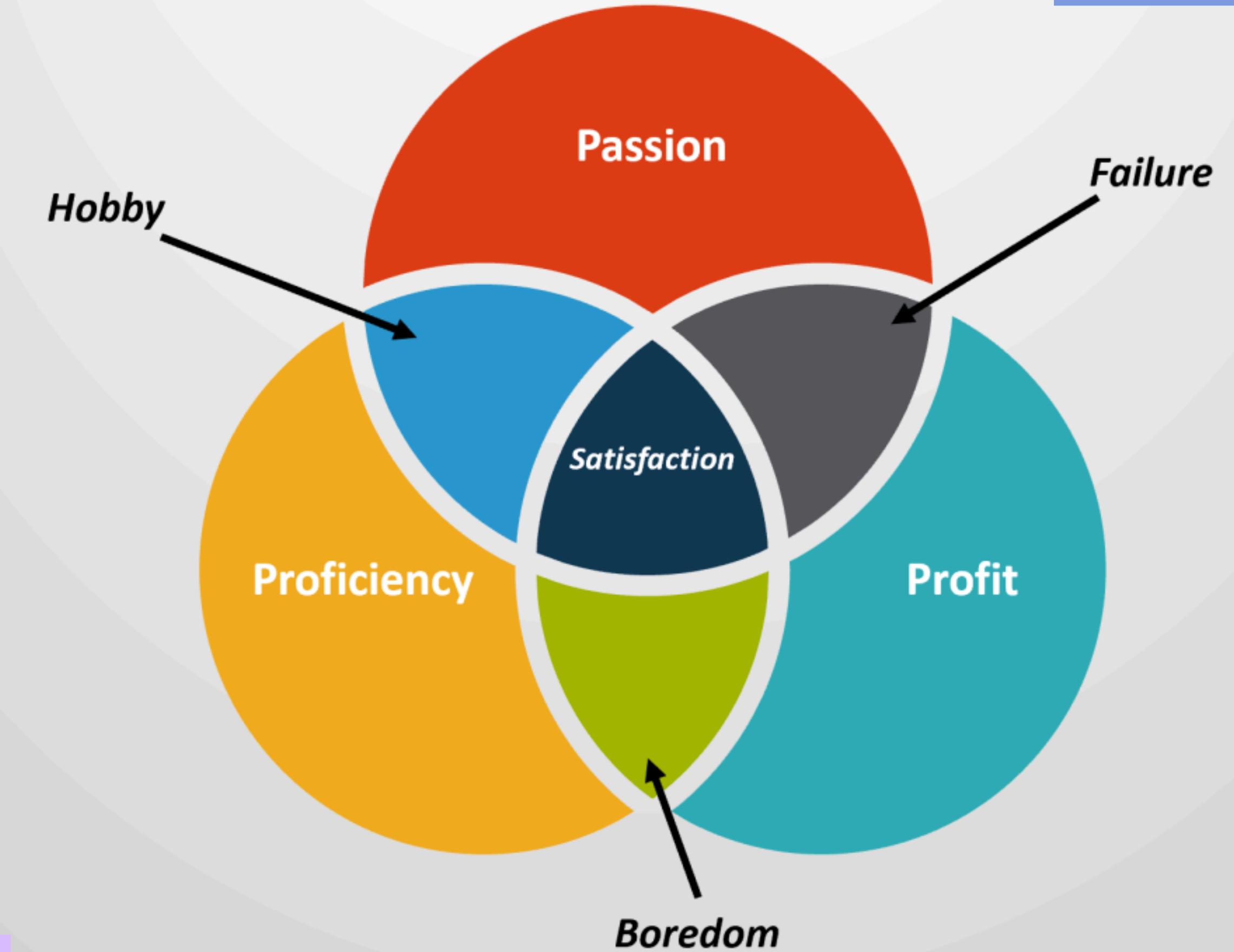


4

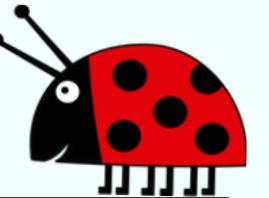
Wrap up & Networking  
19:45 – 20:00



# IKIGAI



# TECHNICAL DEBT

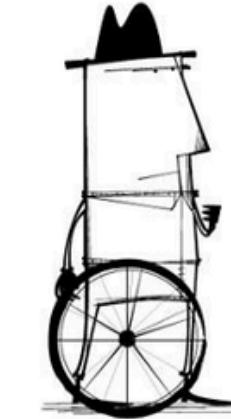


Code is like a garden: without regular maintenance, it quickly becomes a jungle of bugs



ERRR...

CAN'T STOP.  
TOO BUSY!!



# 1. PLANNING & DOCUMENTATION

## Game Design Document (GDD)

- MVP (Minimum Viable Product) i.e. Vertical slice. i.e. Genre, core Game mechanics, Objectives and progression.
- UML Diagram (Sequence & Class Diagram)

## Task Planner

- Identify MDAs (Mechanics, Dynamics & Aesthetics)
- Organize tasks, milestones, and deadlines.



## 2. DEVELOPMENT & COLLABORATION

### Version Control (GitHub Repo)

- Create repo with GDD and UML diagrams in ReadMe.md
- Branch for feature & Link to task
- Open PR (Pull Request) Code review, approve and merge PR to main
- Create semantic release and notify testers or
- [Automate](#) test, Deployment and setup Web-hooks to notify testers.

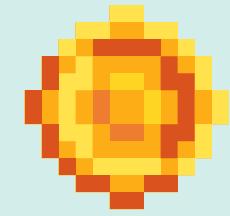


### Collaboration Platform

- communication, feedback and real-time discussions.
- Set up specific channels (e.g. #gameplay, #artwork)

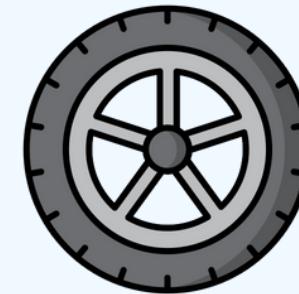


# BUILD, RECYCLE, REPEAT!



## Stop Reinventing the Wheel:

Design for reusability and adaptability.



## Recognize Patterns:

Déjà vus call for a template or even better, [framework](#)!

**<DRY>**  
DON'T REPEAT YOURSELF!

## Build a [Knowledge Base](#):

Document insights for collective learning.



# CODE SMELLS





# BLOATERS

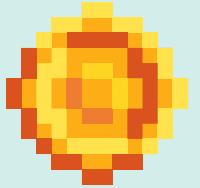
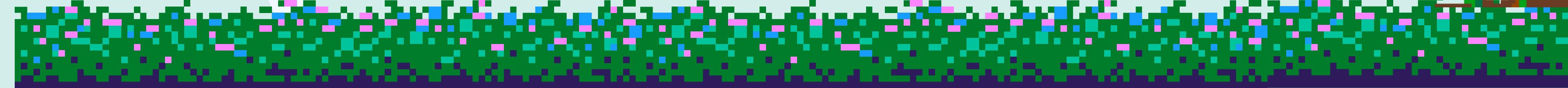
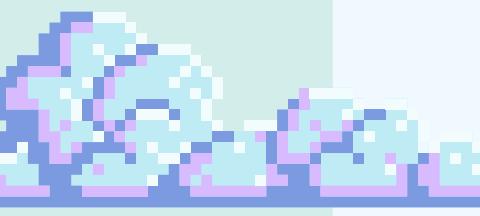
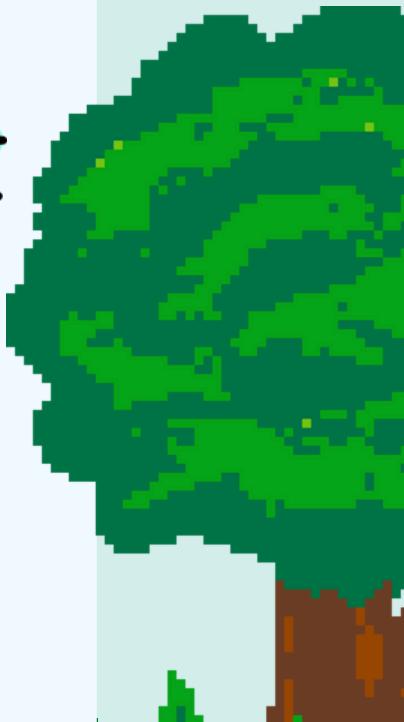
- Large Class: GodManager handling rendering, physics, audio & UI
- Long Method: An Update() with 200+ lines of player logic
- Primitive Obsession: Using float x,y instead of Vector2 for positions



**Solution: Separate of Concerns - Break into smaller, focused components**



CODE SMELLS





# 💩 PREVENTERS

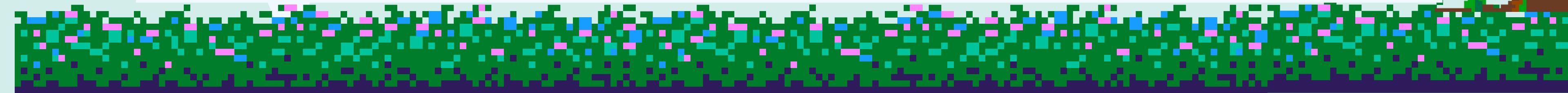
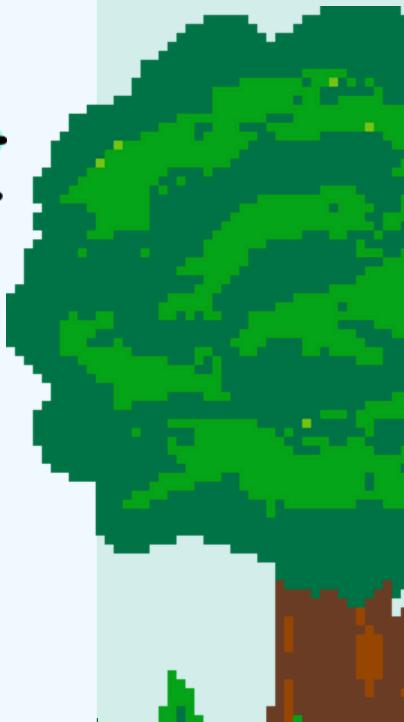
- Divergent Change: Enemy class changes for both AI and animations
- Shotgun Surgery: Speed change affects Player, Animation & Effects
- Parallel Inheritance: New Character requires matching AnimationController

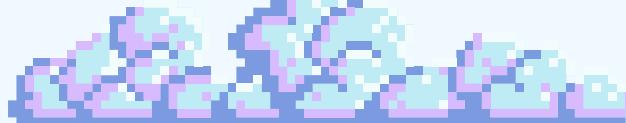


**Solution: Composition over Inheritance - Use component-based architecture**



CODE SMELLS





# 💩 COUPLERS

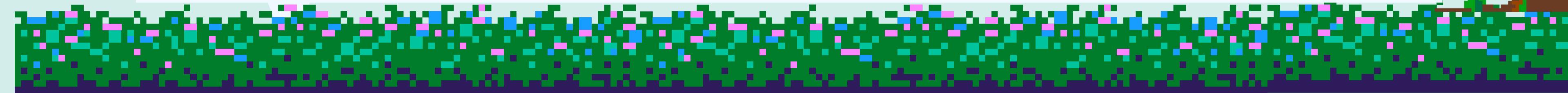
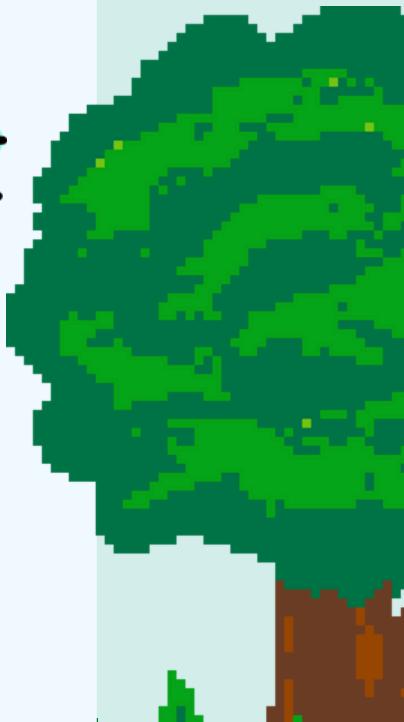
- Feature Envy: HUD constantly accessing Player's private stats
- Inappropriate Intimacy: Weapon directly modifying Player's health
- Message Chains: player.GetInventory().GetWeapon().GetAmmo()



**Solution:** Event-driven communication - Observer pattern - Law of Demeter



CODE SMELLS





# 💩 DISPENSABLES

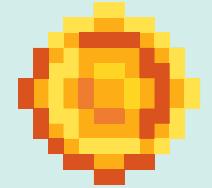
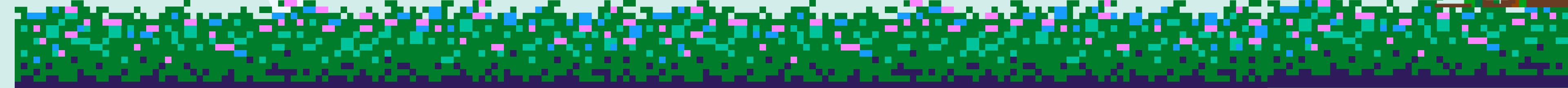
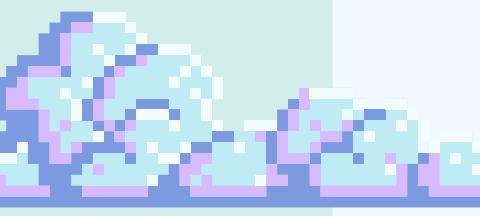
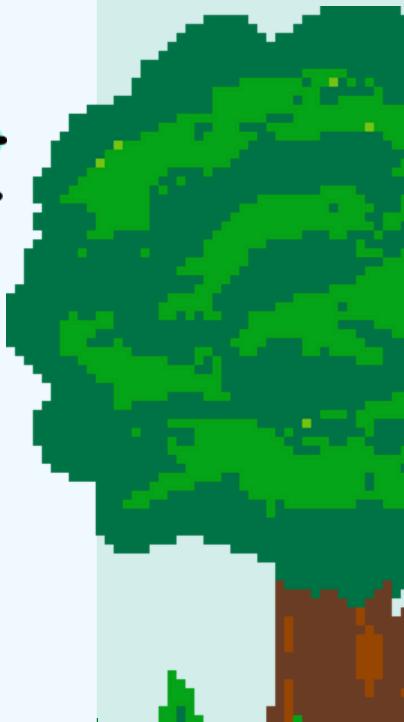
- **Comments:** "://This complex math calculates gravity" instead of named method
- **Duplicate Code:** Same damage logic in all enemy classes
- **Dead Code:** Leftover multiplayer code in single-player game



**Solution:** Extract, Abstract, Delete - Follow DRY principle



CODE SMELLS





# 💩 ABUSERS

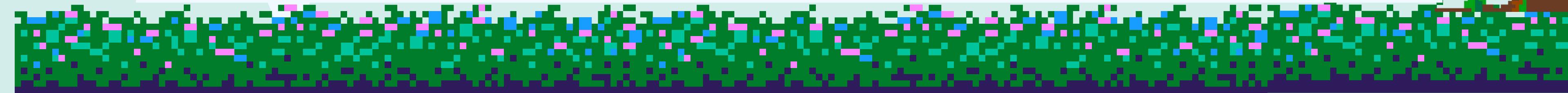
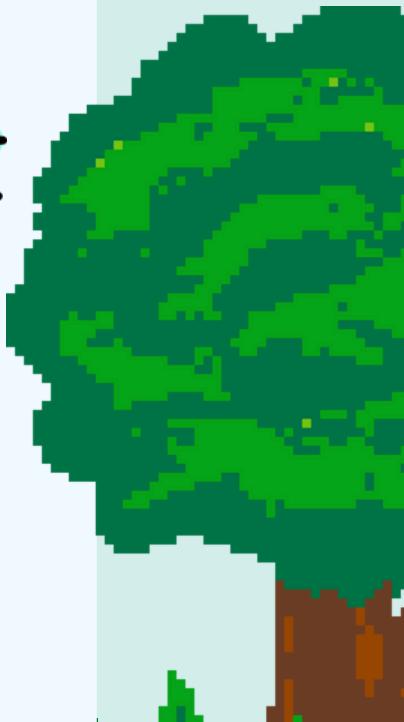
- **Switch Statements:** Huge switch for different powerup types
- **Temporary Field:** UnderwaterOxygen only used in water levels
- **Alternative Classes:** EnemyMover and MonsterController doing the same job



**Solution: Interface-based polymorphism - Replace conditionals with inheritance**

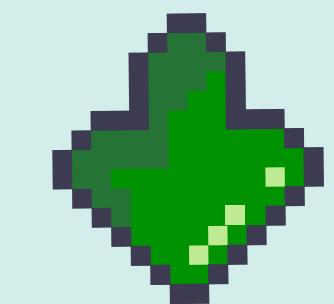
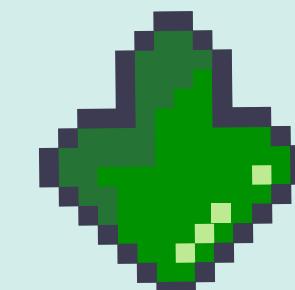
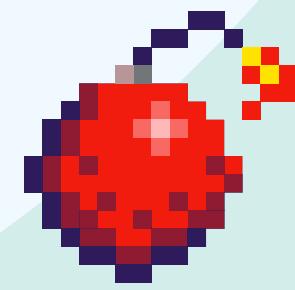


CODE SMELLS



# DOS AND DONT USE!

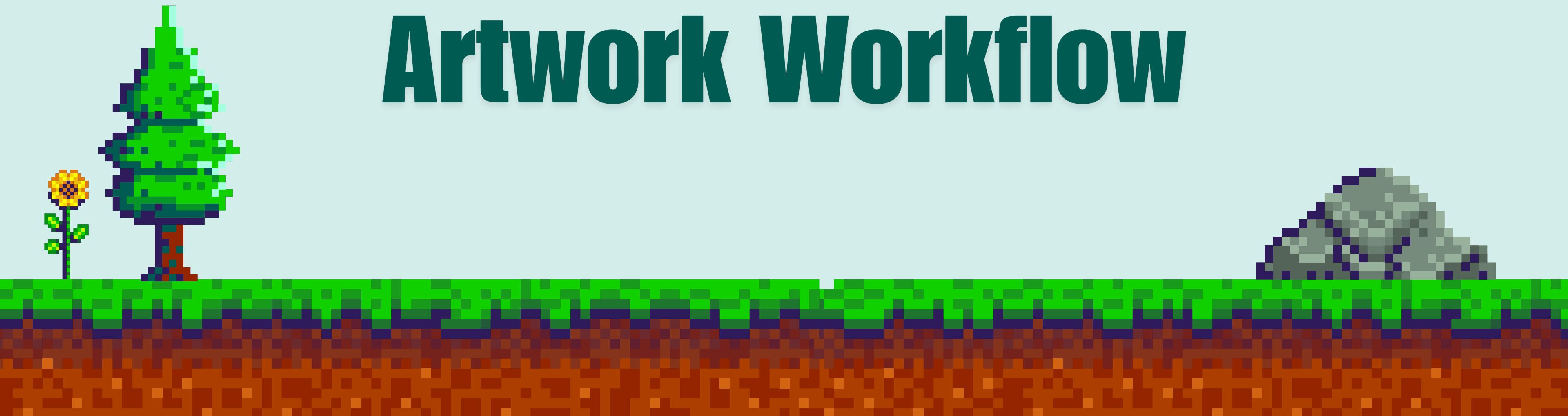
---





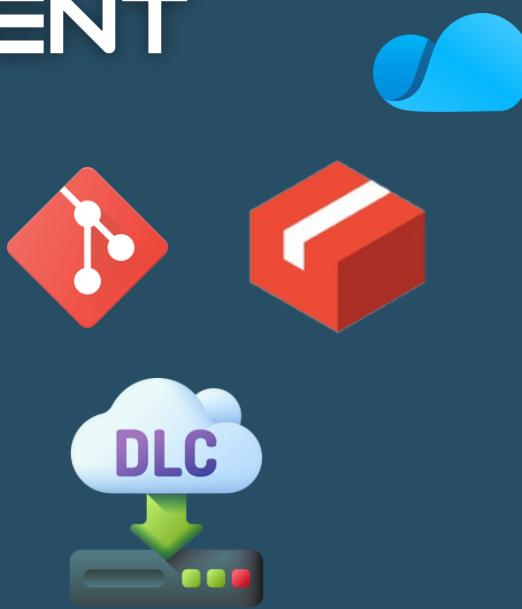
---

# Artwork Workflow



## RESOURCE MANAGEMENT

- Asset Repos
- Versioning with LFS
- DLC using Addressables



## MODELING

- Object Topology
- Origin, hierarchy, scale & rotation.
- Poly count and LOD

## SHADING

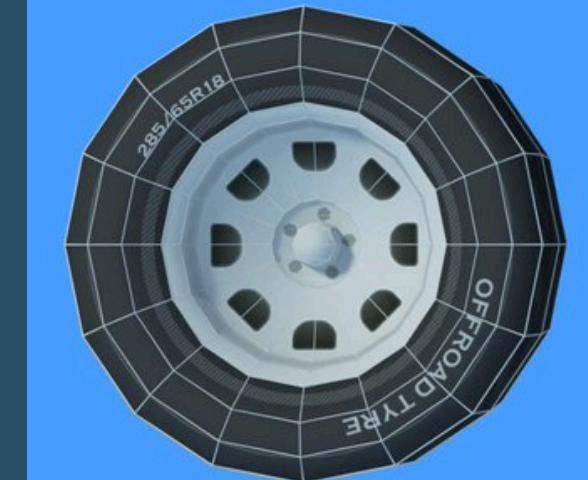
- Normals orientation and weight
- Limited materials with PBR
- Textures: Atlas maps
- Shaders: simplify and expose parameters
- Post processing: Bake Lights & AO

## ANIMATION

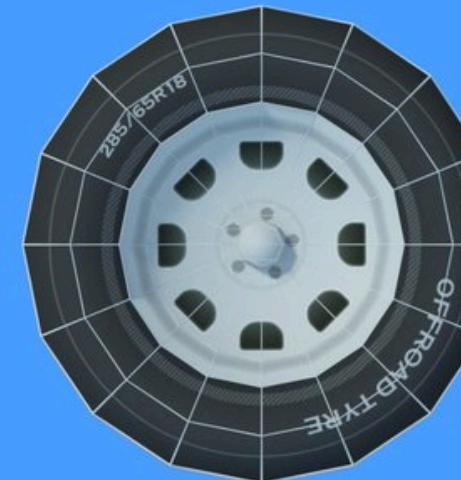
- Morphing/Shape keys names
- Rig and drivers
- Topology and engine triangulation
- Rig config & reusable controllers
- FSM and cross fade
- Motion tweens



**LOD0**

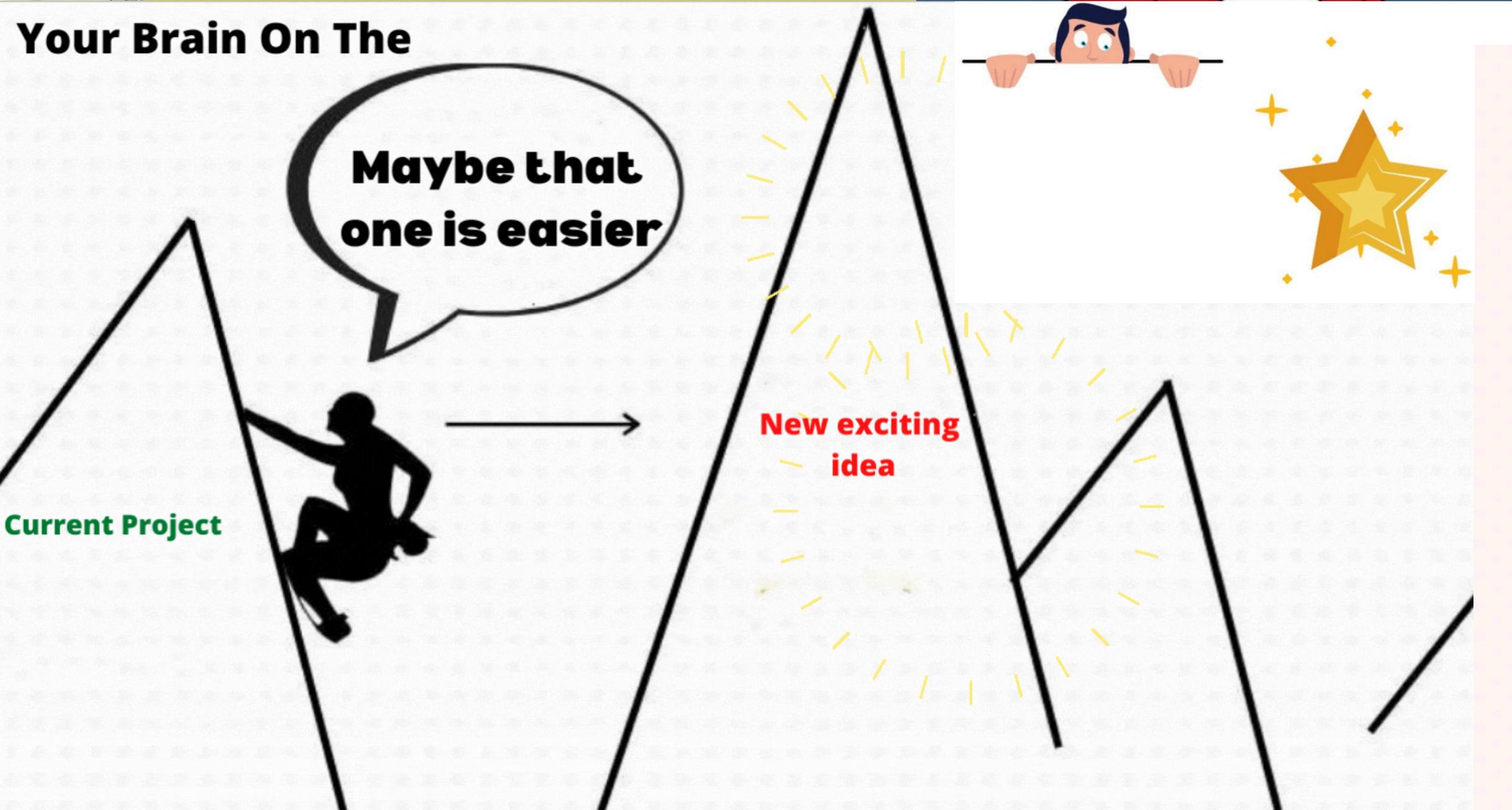


**LOD1**



**LOD2**





# THANK YOU!

GO HOME

