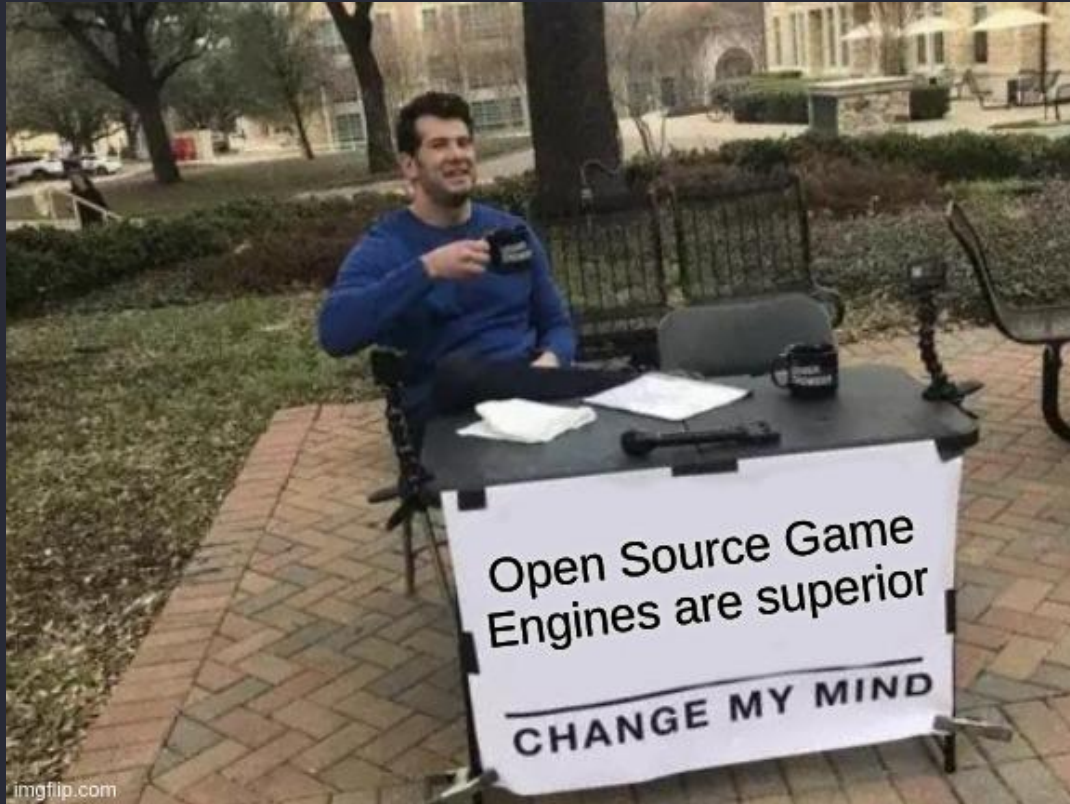# Agenda

- Idea
- Milestones
    - Playable Prototype
    - Software Architectures
    - Addictive Patterns
    - Game Analytics
- Design Review
- Technical Review
- Whats next?

Agenda

Idea

First Draft

A Journey of Music - Idea (Jan)

Idle Game

A Platformer (like Super Mario or Rayman Legends) synced with famous music tracks. The Player can travel through a History of Musical Events: From Mozart over Good' Old' Rock'n Roll to Billie Eilish or something -> kinda like this one https://www.youtube.com/watch?
* Jan Neumann

What is the Game about?
In "A Journey of Music" you are a student dreaming about Music and travelling through a musical time.

Technical Side
The Game be able to nearly all PC's; windows over Mac to Linux and be playable by Controller
https://godotengine.org/

Challenges
Its only 2d with parallax effect for background + the levels have to be designed in such a way, that they are synced to the music and only solvable if you follow the "rython"
Game Scrolling automatically

Level:
1. Mozart (classical music) -> Walking through an old german/austrian town
2. Rock'n Roll (rock setting (60s/70s)) -> maybe a concert with a stage in the background
3. Modern Age (maybe Oldschool Rap, or Eminem (Rap God)/(Rollegah) -> Strip Club or so

TODO:
- set up blog
- try out godot
- simple poc with blocks

Inspiration:
Inspiration for this Game are old Platformer-Games like Mario and Rayman Legends. The Idea with the synced Music is from Rayman as seen here:
https://www.youtube.com/watch?

https://blog.newmanity.eu/

Further Ideas:
- abstractness of dreams
- Eminem in Mozart

Artstyle of the Game (Pixelart)

https://www.artstation.com/artwork/W6wA

Idea

- rythm-based-platformer
    - like rayman legends
- story about a music student
- cross-plattform
- controller-playable
- levels synced to music
    - algorithmic level creation
- character selection

Graphics

- pixel-art style
- parallax-background
- 2d
- cute little characters

Technical Side

Milestones

Playable Prototype

Playable Prototype

Objectives accomplished:

- Setup of the Blog
- Movement
- Sprite Animation
- Parallax Background

Objectives not accomplished:

- physics
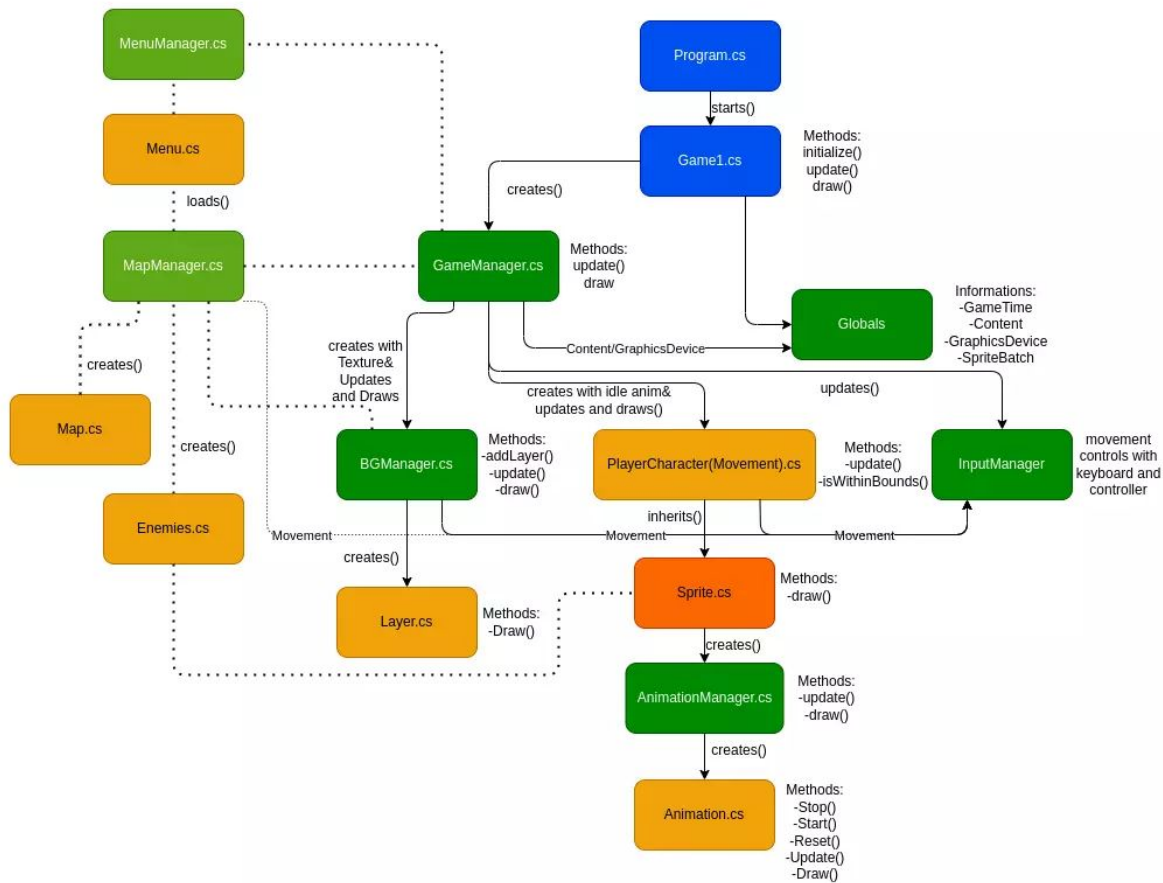- map

Playable Prototype

(old) Software Architecture

(old) Software Architecture

And the best part is, it's open source!

- based on C# and MonoGame > BAD MISTAKE
- object-oriented
- inheritance-based
- Example Hierarchy:
  - Gameloop > GameManager > ...Managers > Object > Subobjects
- everything has to be written ourself
- at least, our blog is working beautifully!

(old) Software Architecture

MenuManager.cs

Menu.cs

loads()

MapManager.cs

creates()

Map.cs

creates()

Enemies.cs

Program.cs

starts()

Game1.cs

Methods:
initialize()
update()
draw()

creates()

GameManager.cs

Methods:
update()
draw

Content/GraphicsDevice

Globals

Informations:
-GameTime
-Content
-GraphicsDevice
-SpriteBatch

creates with
Texture&
Updates
and Draws

creates with idle anim&
updates and draws()

updates()

InputManager

movement
controls with
keyboard and
controller

BGManager.cs

Methods:
-addLayer()
-update()
-draw()

PlayerCharacter(Movement).cs

Methods:
-update()
-isWithinBounds()

inherits()

Movement

Movement

Movement

creates()

Layer.cs

Methods:
-Draw()

Sprite.cs

Methods:
-draw()

creates()

AnimationManager.cs

Methods:
-update()
-draw()

creates()

Animation.cs

Methods:
-Stop()
-Start()
-Reset()
-Update()
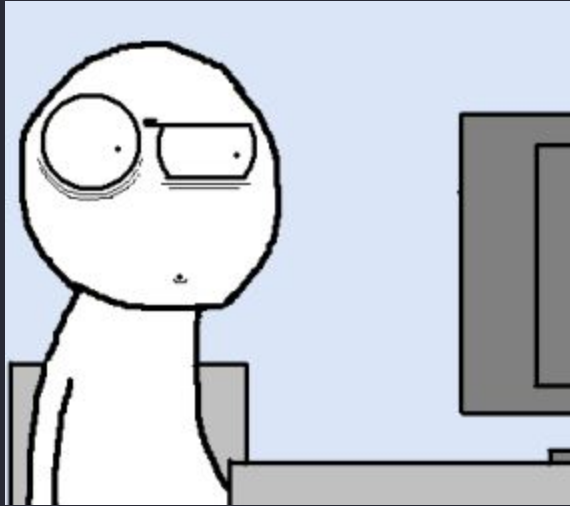-Draw()

Addictive Patterns

Addictive Patterns

- basic addictive patterns of platformer games like super mario
- further points added by commilitones:
  - coins
  - highscore
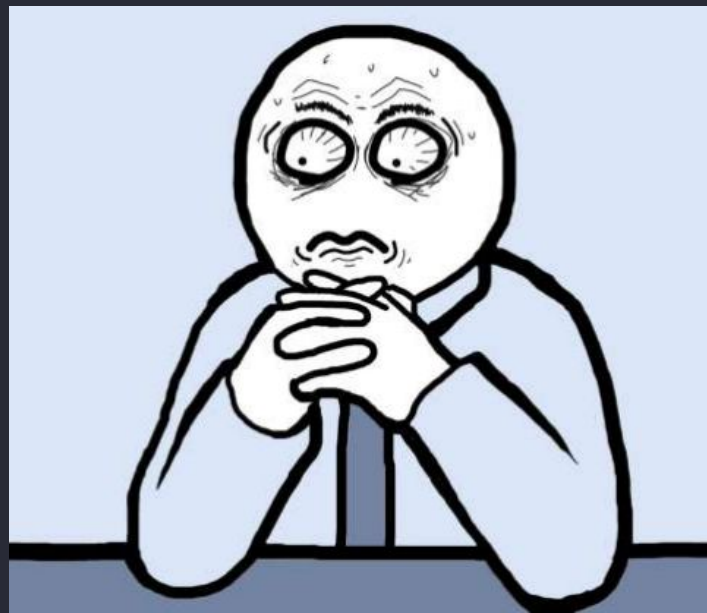 - accuracy

Addictive Patterns

Come Back

Come and try it

Share it with a Friend

explore

collaborate

compete

express/create

Finishing a level: Listening to a complete Song

Advancing to the next level: Enjoy new Skins and Songs

Beat yourself: If you lose, you lost nothing, but if you win, you already better than yourself

More complicated levels: 4/4 on the floor is easy, how about something complicated, like jazz?

Expand the game: With an algorithm to level building, a player should be able to import his own songs

Cooperative Multiplayer: Master the levels together with your dearest (NIS)

Meanwhile, in the code-corner:

Game Analytics

Developer's reaction:

Game Analytics

Three main categories of analytical data:

- game popularity
- game hardness
- player skill

Game Popularity

- Retention rate
- Time spent in-game
- Average session length
- Social sharing
- Song popularity

Game Hardness

- Completion rate
- Number of levels played
- Song completion rate
- Death rate
- Rage quit rate

Player Skill Rating

- beat accuracy
- Song completion rate
- Score/points earned

Design Review

source:
https://image.geo.de/30
977986/t/dv/v1/w1440/r1/
-/mozart-jugendlich.jpg

Mozart

source:
https://www.selfies.com/wp-content/uploads/sites/5/2023/04/gene-simmons-kiss-schwacheanfall.jpg?w=1200

Kiss

source: (screenshot)
https://www.youtube.com/watch?v=_etGw_lsNrI&ab_channel=ALPHAMUSICEMPIRE

Kollegah

Animations

Idle

Run

Jump
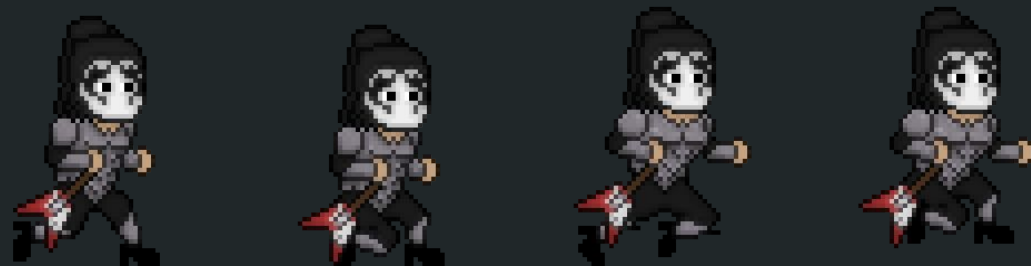
Tools
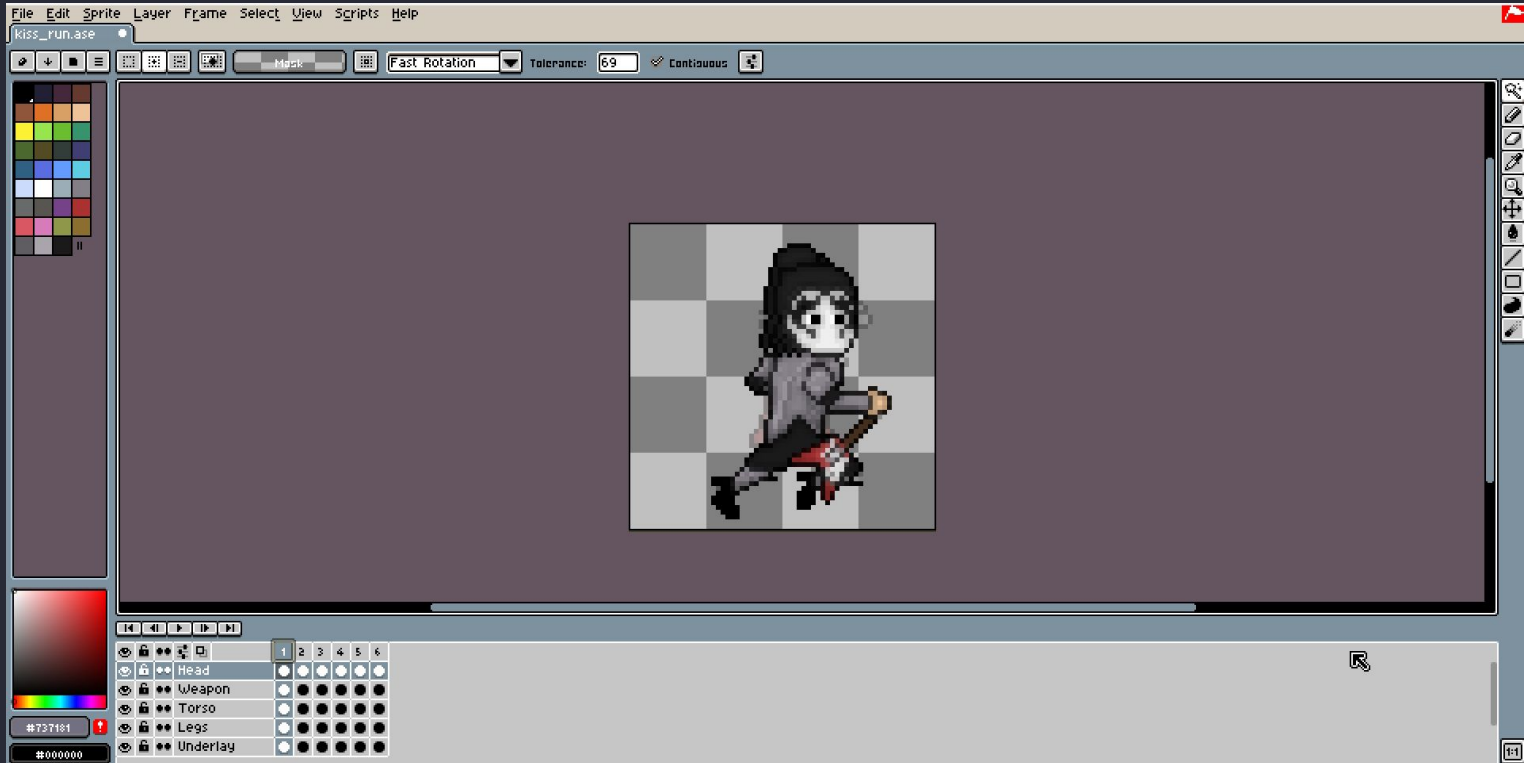
Background

Layer 1

Layer 2
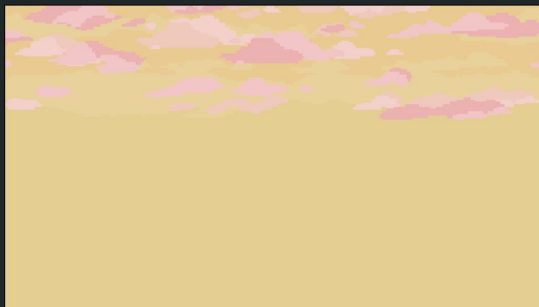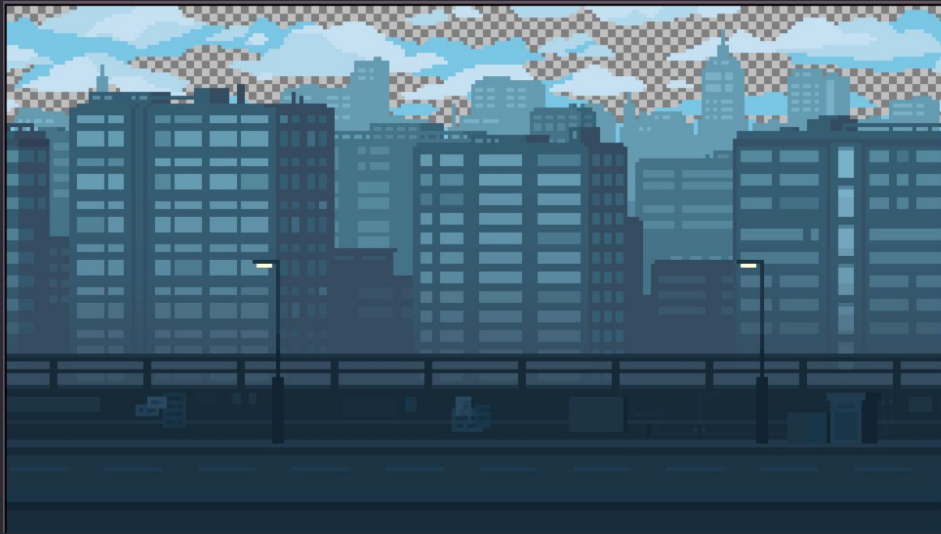
Layer 3

Layer 4

Layer 5
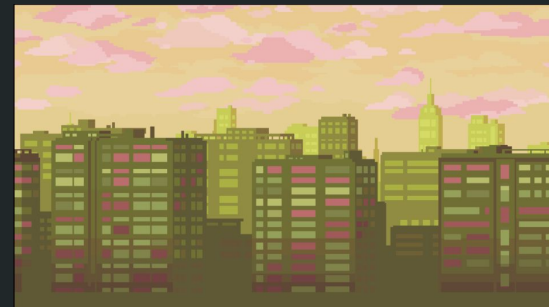
Layer 6
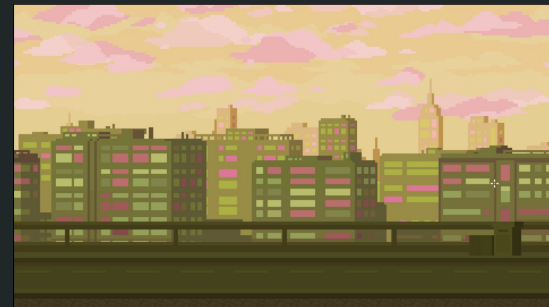
Comparison

Old

New

Technical Review

Technical Review - Foreword

- gain experience in game development
  - apply to personal projects
  - grow your beard of knowledge ;)
- place to try things out
  - we did!
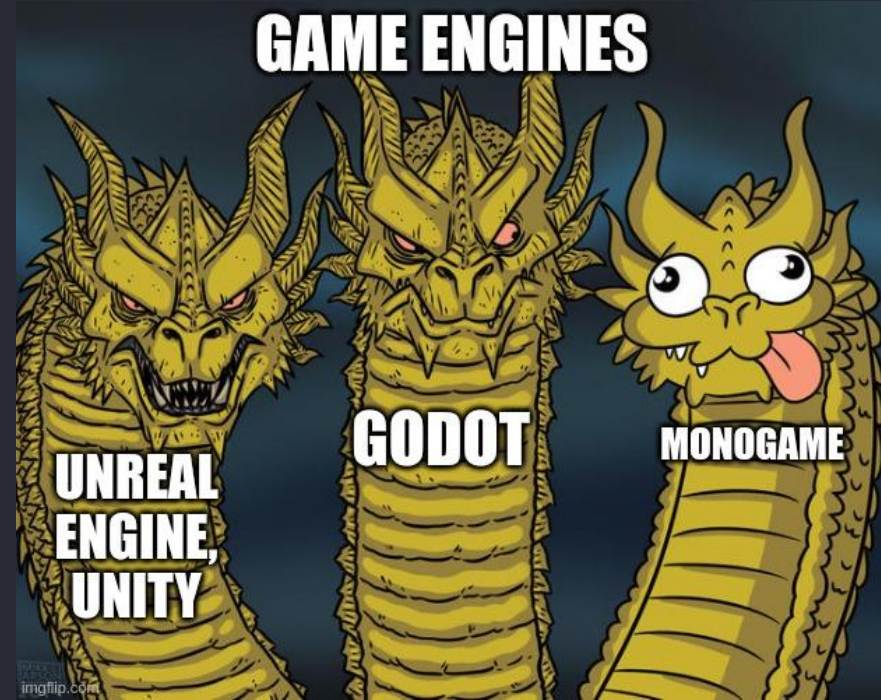- open source software

Technical Review - Why Monogame

- more control over your game
  - highest optimization potential
- cross compatibility
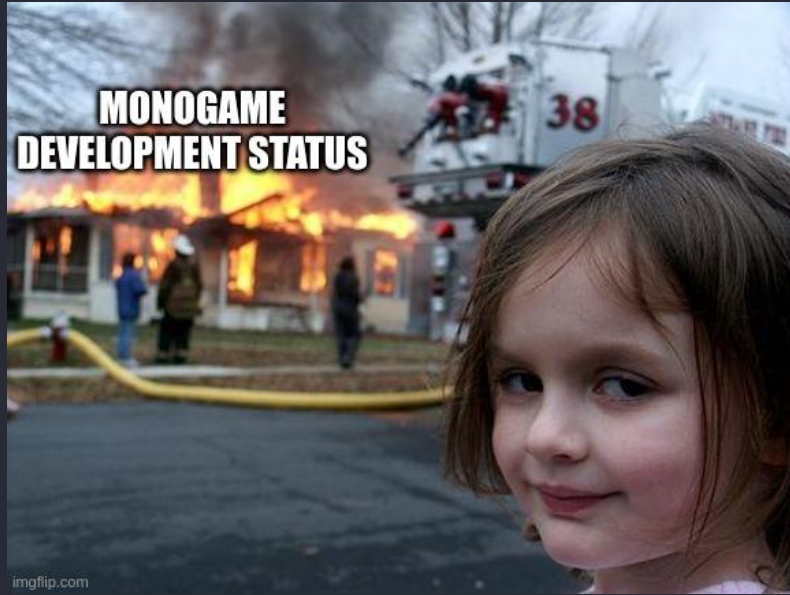- game-vibes we want to imitate

Technical Review - Monogame - Cons

- outdated engine
- outdated libraries
- little to no documentation
- write everything yourself

Transition to Godot

Transition to Godot

Transition to Godot

- felt, that MonoGame was too much for us
- Godot has a much broader & younger community
- good Documentation
- many Libraries
- TUTORIALS
- WYSIWYG

Transition to Godot

Transition to Godot

- Rebuild entire development status within one weekend
- added further features
  - menus
  - moving sky
  - coins
  - scoring
  - death
  - local game analytics (logs)
  - sounds

Live Demo & YT-Video

Whats next?

Our Game is not nearly finished!

What we will do (in our freetime & after the 22.)

- add a second level, with the promised rhythm based jumping
- opt for a game analytics service that actually works
- improve game & code structure
  - cleanup
- setting up ci/cd for automatic releases
- most important: add documentation
  - when we learned one thing, its that documentation is key
- extend our blog

Godot   MonoGame

What can the music students do?

- develop game sounds
  - walking, jumping, dying, coins, success sound
- develop music
  - rock, classic, hip hop
- have ideas for the game
  - and mail them to us, so we can write them into a list on the blog

What can everybody else do?

- It's open source
    - you can do everything you like with it
- You can read our blog
    - it's cool
    - if you have some ideas, write us an e-mail: ajourneyofmusic+blog@newmanity.eu
    - look at our then freshly created meme-site about our project :)

Thank you for your attention!

The End