Match-three game engine

Krzysztof Rudnicki

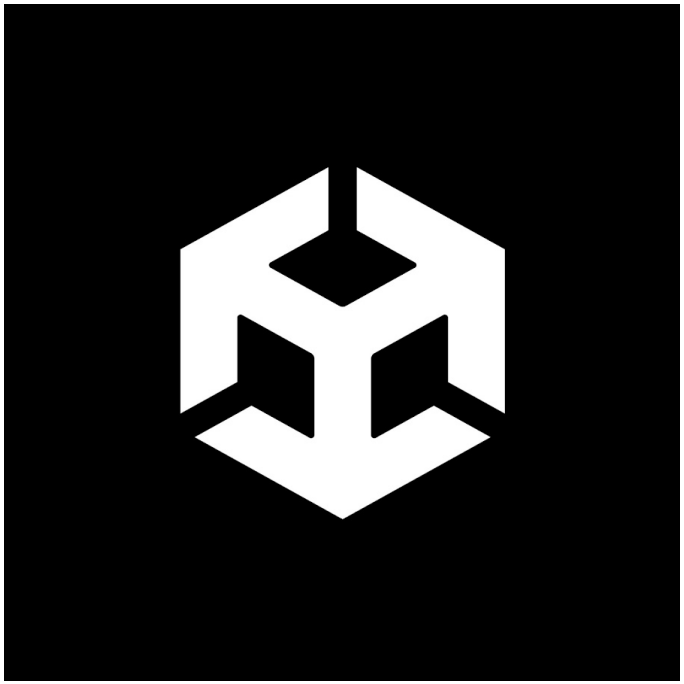
June 2, 2023

Presentation plan

Game-engine

Software framework designed for developing video games

General engines



Specialized engines

Succesful games

TO THE MOON

Why specialized engine

- Smaller scope (easier)
- Existing success stories
- Less competition
- Unix philosophy

match-three example

Match-three concepts

- Board Representation
- Game Logic
- User Interaction
- Animation and Graphics

Why match-three games

- Relatively easy to program
- Popular
- Work on everything
- Graphics dependable

Multiplatform

- Linux
- Windows
- Android/iOS
- Mac

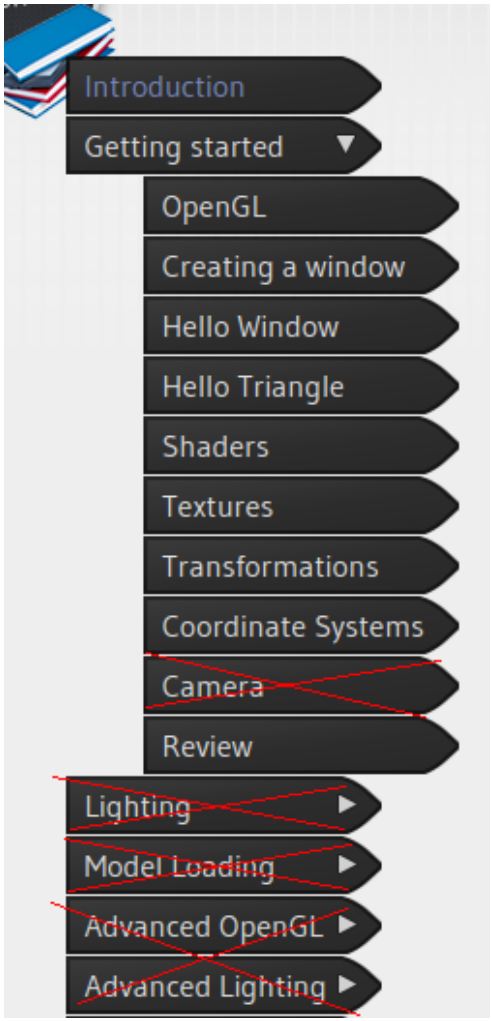
Project scope

- Custom graphics (tiles/backgrounds)
- Custom shaders
- Defining game rules
- Building multiplatform

Technology

- C++
- OpenGL ES (API)
- GLFW (windows/input)
- GLM (math)

Ease of match 3



Smaller scope

Engine design overhaul

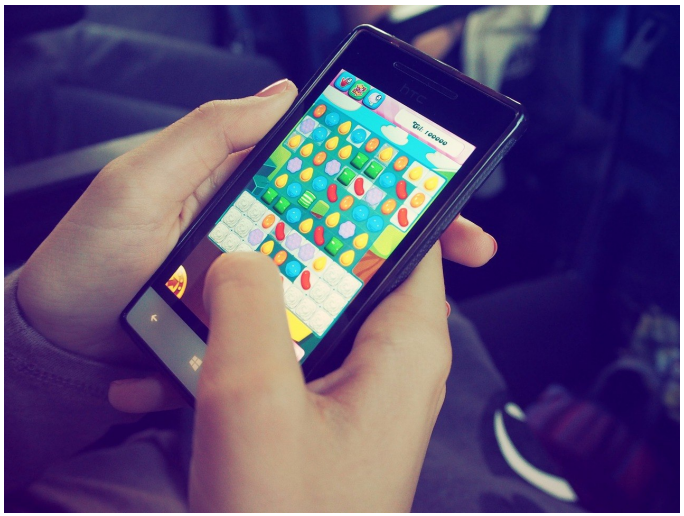
- Window/Input management
- Graphics/Rendering
- Gems attributes
- Game Logic

Window/Input

GLFW:

- Create window
- Handle input
- Rendering context

User inputs



Graphics/Rendering

- Shaders
- Vertex Buffer Objects
- Vertex Array Objects

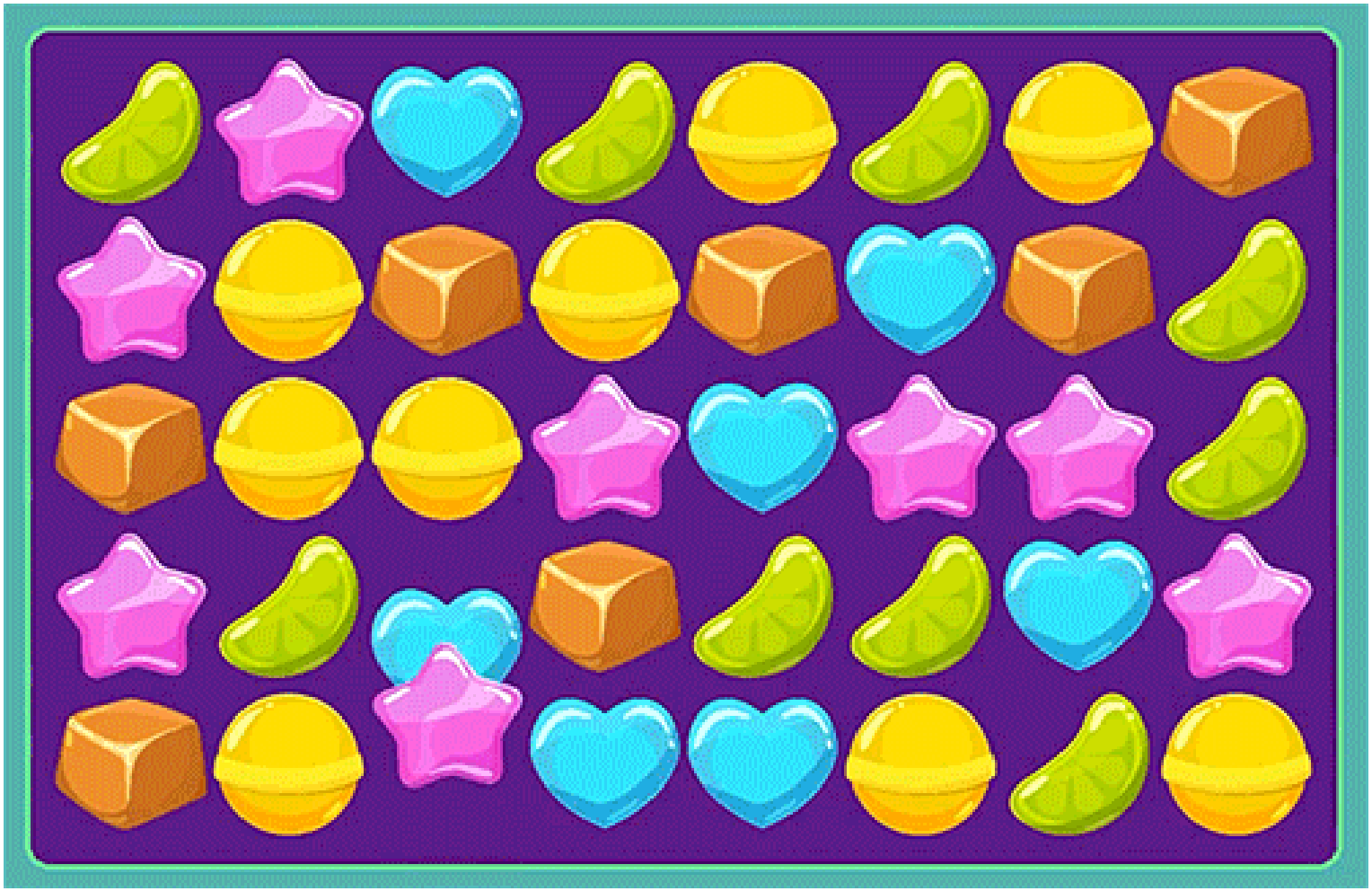
Gems attributes

- Type
- Status
- Position
- Value

Game Logic

- Check for matches
- Remove matched
- Fill board

Animation



Summary

Thesis is about creating a game engine specialized in match-three multiplatform games using OpenGL

References/sources

- https://docs.gl/
- https://learnopengl.com/
- The Cherno
- Game Engine Architecture, Jason Gregory