Portfolio

Andrey Egorov

Works below were created at CGSG(Computer Graphics Support Group) of St.Petersburg PML 30 during 2021-2024 period. Some of them are team projects, some are independent. Most of code is written in C/C++ on Windows.

My GitHub page

TMP(Tough Megapolis Planner)

Urban environment design system - team project (8 people).

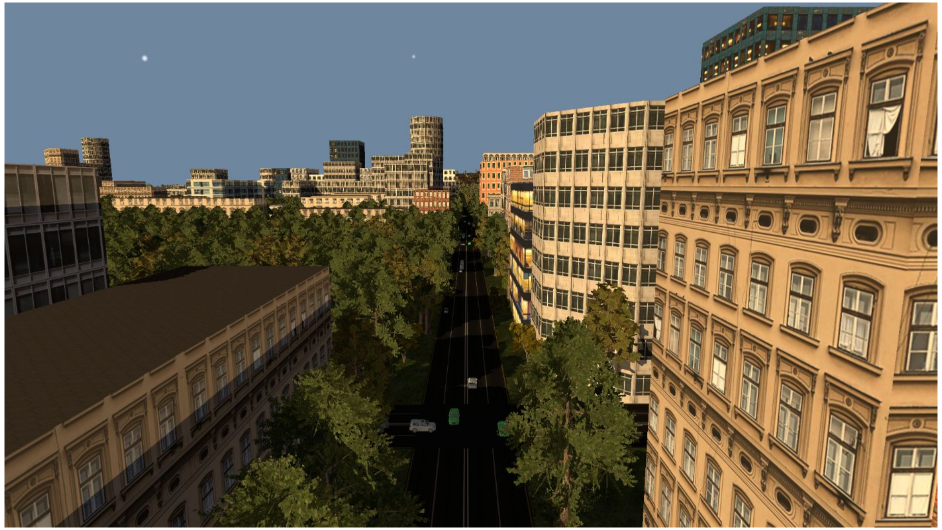
My part: user interface(2nd gen), building system.

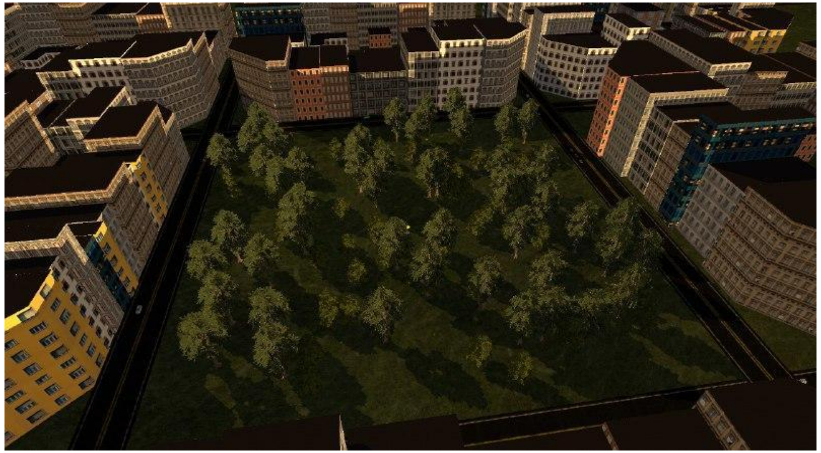
Main features:

- multi-thread render core on Vulkan API.
- road system and traffic simulation.
- landscape system that uses Perlin noise and erosion algorithms. To speed up the landscape drawing we used adaptive tessellation.
- second degree award at 23rd Kolmogorov Readings.

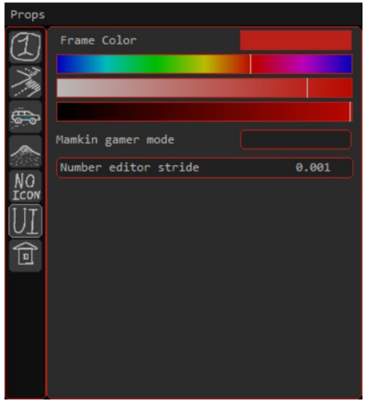
Presentation

(C++, WinAPI, Vulkan API) 2022-23





2nd gen UI:

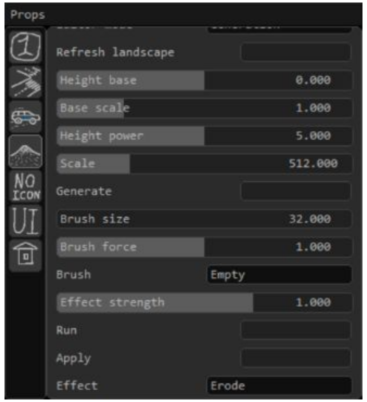


Props

Frame Color

Mamkin gamer mode

Number editor stride 0.001



Props

Refresh landscape

Height base 0.000

Base scale 1.000

Height power 5.000

Scale 512.000

Generate

Brush size 32.000

Brush force 1.000

Brush Empty

Effect strength 1.000

Run

Apply

Effect Erode

TER(Tough Environment Redactor)

System for modeling and visualization of the environment in rea
time - team project (13 people).

My part: user interface(1st gen), UBO/SSBO buffers, compute shaders usage

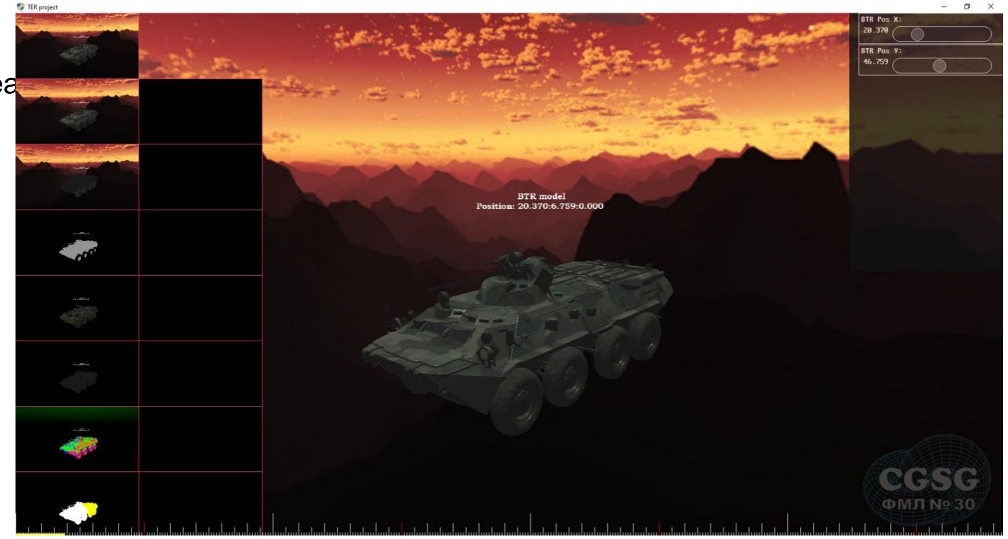
Main features:

- GPU accelerated OpenGL render core.
- unit system.
- matrix calculating at compute shaders.

[Presentation]

[Video presentation(Russian)]

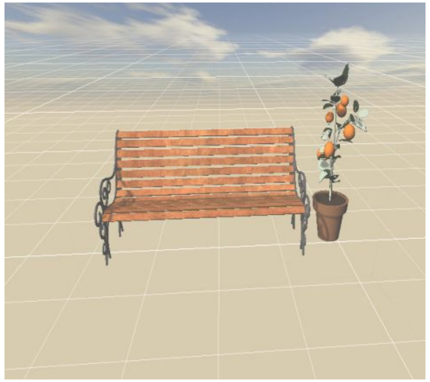
(C, WinAPI, OpenGL) 2021-22

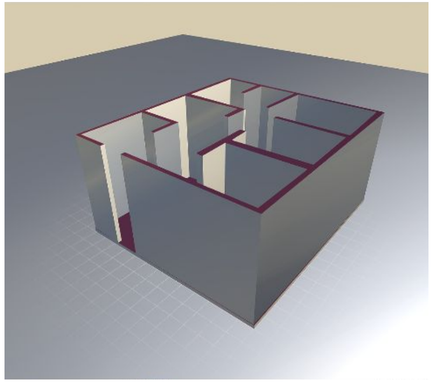


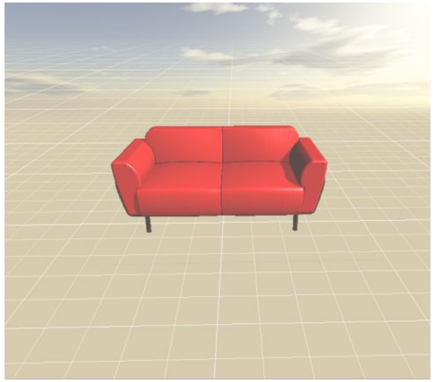
TAP(Tough Ambiance Plotter)
Interior design system - team project (8 people).
My part: user interface(3nd gen).
Main features:

- updated platform independent multi thread architecture with global message queue and separated interfaces and implementations.
- multi-thread render core on Vulkan API.
- physics collision system.
- environment edit tools.
- third degree award at 24rd Kolmogorov Readings.

Presentation
(C++, WinAPI, Vulkan API) 2023-24





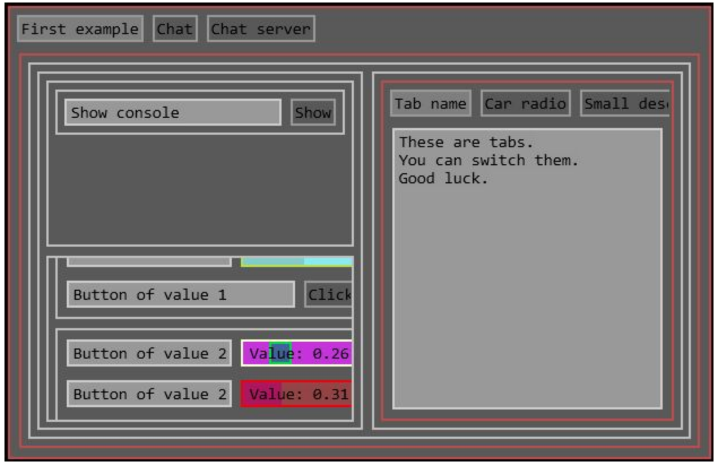


User Interface(3rd gen)

Main features:

- advanced resize system.
- deferred render.
- draw functions of simple objects are accelerated by assembler.
- element redraws only if it has changed, so by default UI doesn’t influence on performance.
- dynamic update

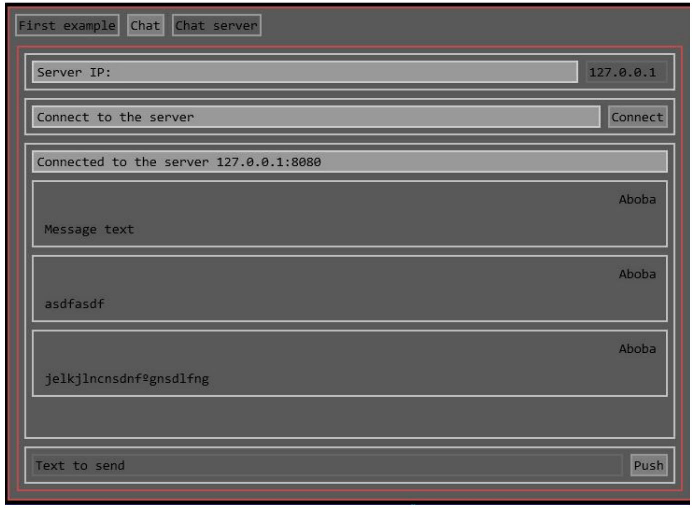
(C, WinAPI, OpenGL) 2023-34



```
These are tabs.
You can switch them.
Good luck.
```

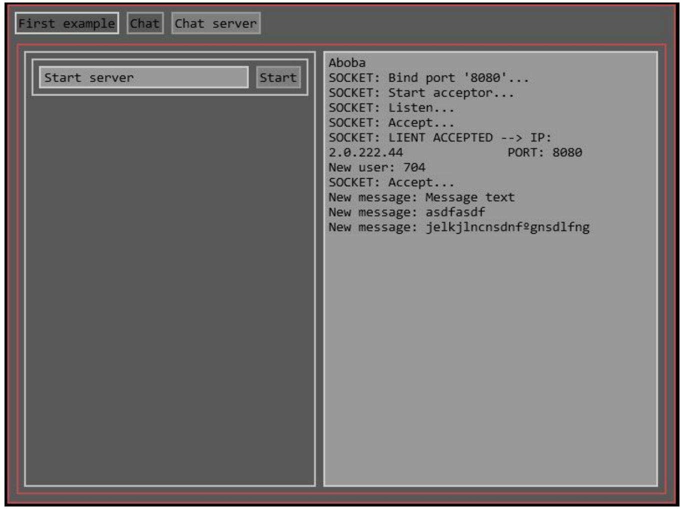
Win socket client-server messenger example:

Client:



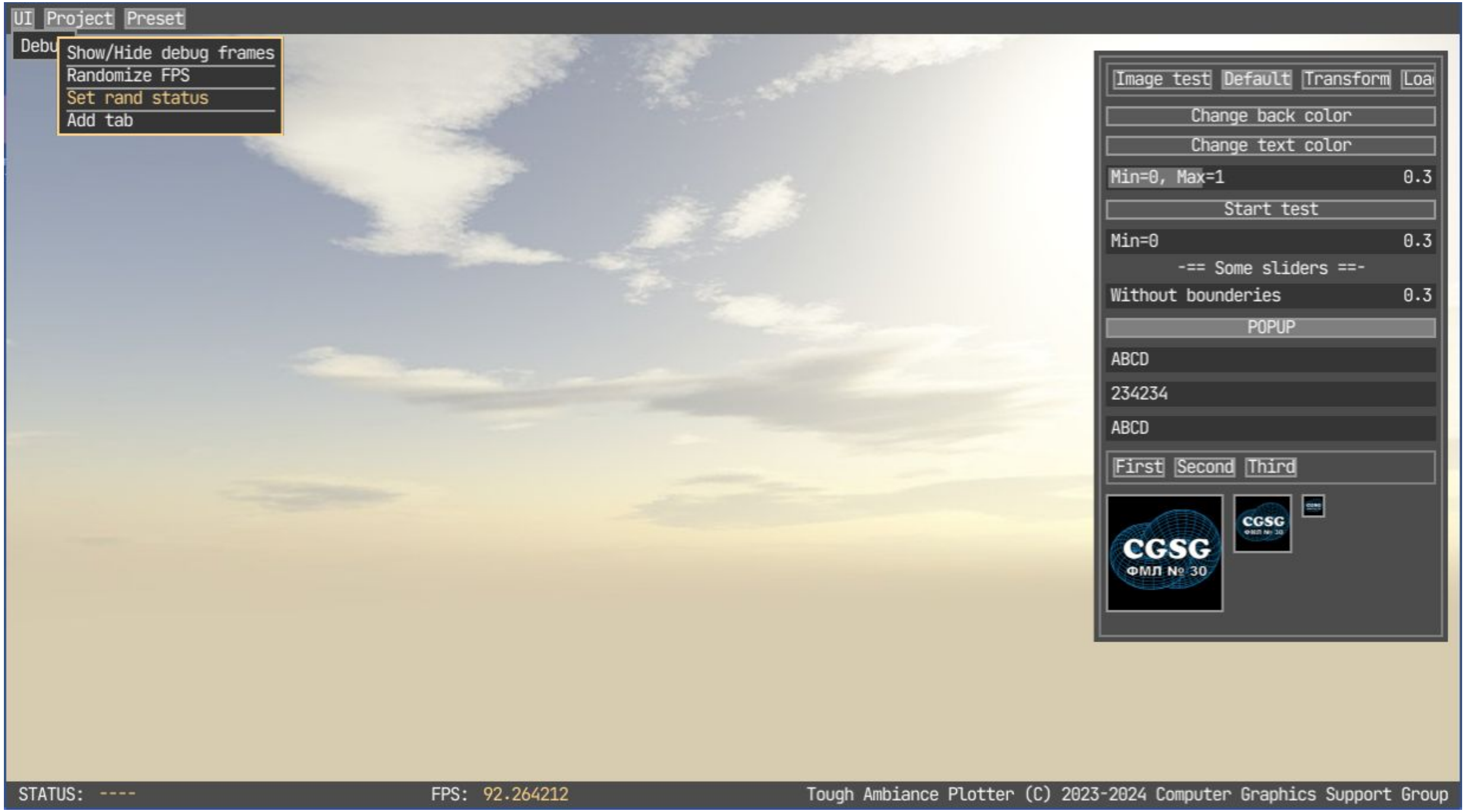
```
Connected to the server 127.0.0.1:8080

Aboba
Message text

Aboba
asdfasdf

Aboba
jelkjlncnsdnfºgnsdlfng
```

Server:



```
Aboba
SOCKET: Bind port '8080'...
SOCKET: Start acceptor...
SOCKET: Listen...
SOCKET: Accept...
SOCKET: LIENT ACCEPTED --> IP:
2.0.222.44            PORT: 8080
New user: 704
SOCKET: Accept...
New message: Message text
New message: asdfasdf
New message: jelkjlncnsdnfºgnsdlfng
```

Example of user interface(3rd gen) in TAP



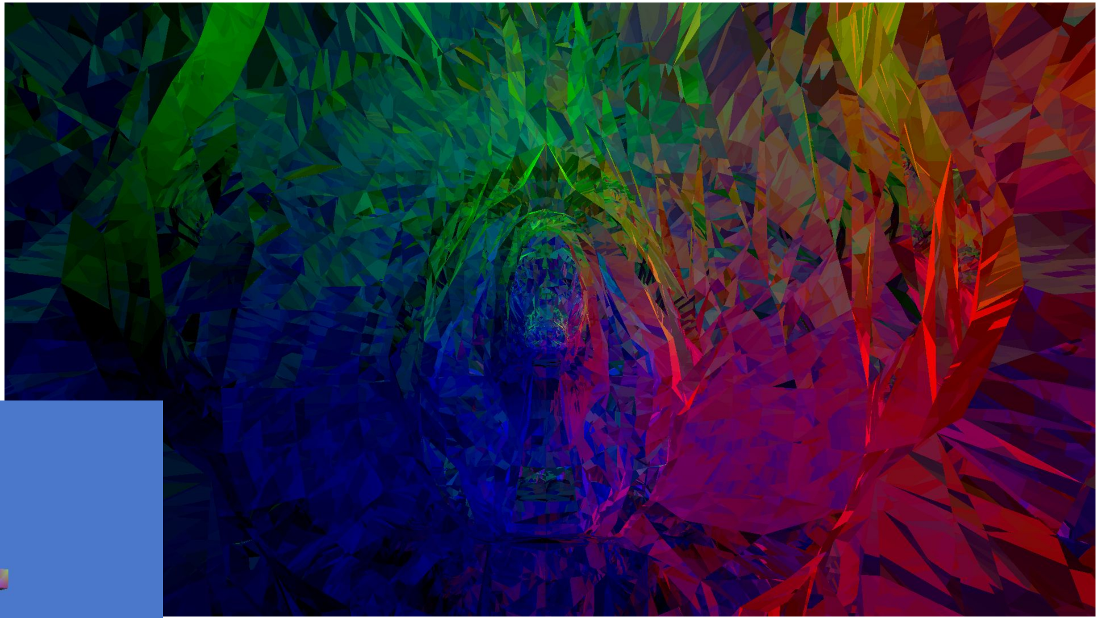
RTX

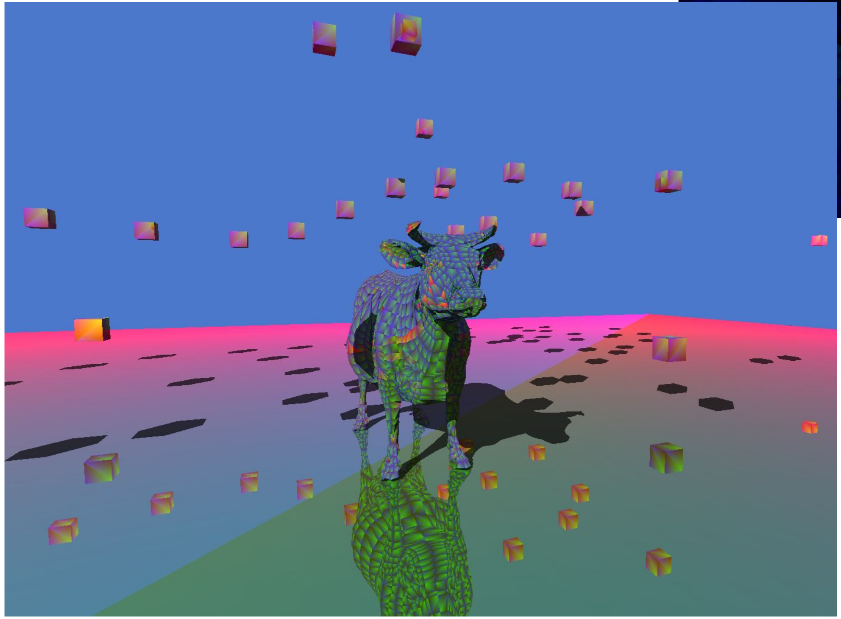
Render core based on Vulkan API that uses GPU accelerated ray tracing algorithm.

Main features:

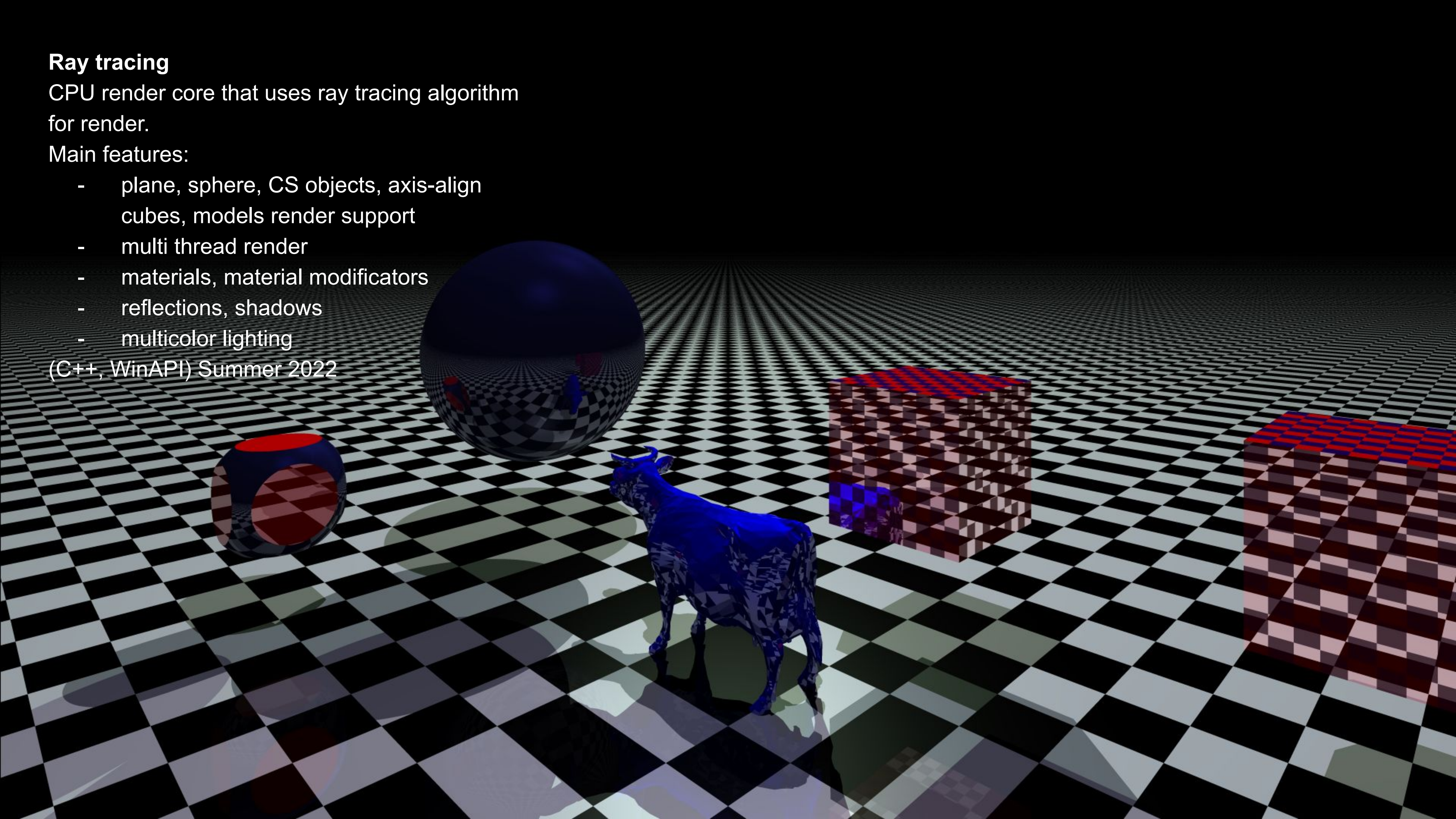
- Realistic render with reflections and shadows
- Real time rendering

(C++, WinAPI, Vulkan API) Summer 2023





Inside cow(example of 3 level recursive reflections; FPS: ~60)



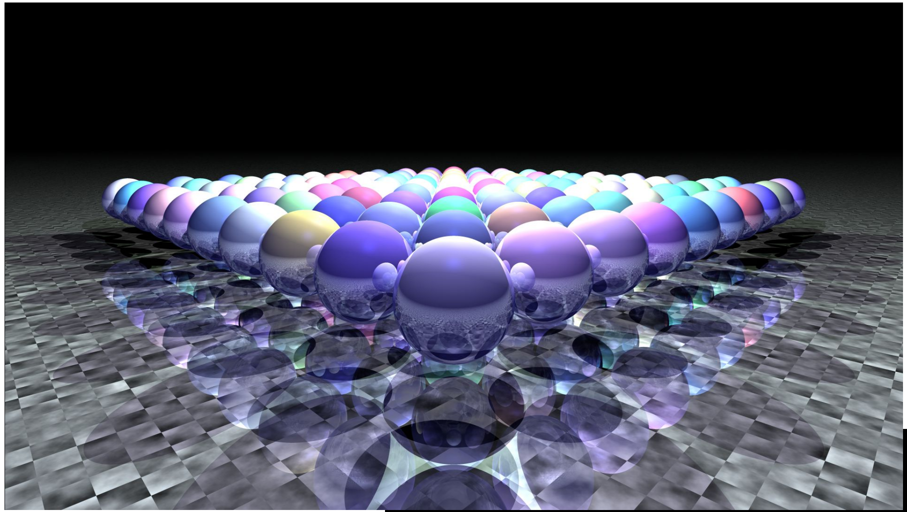
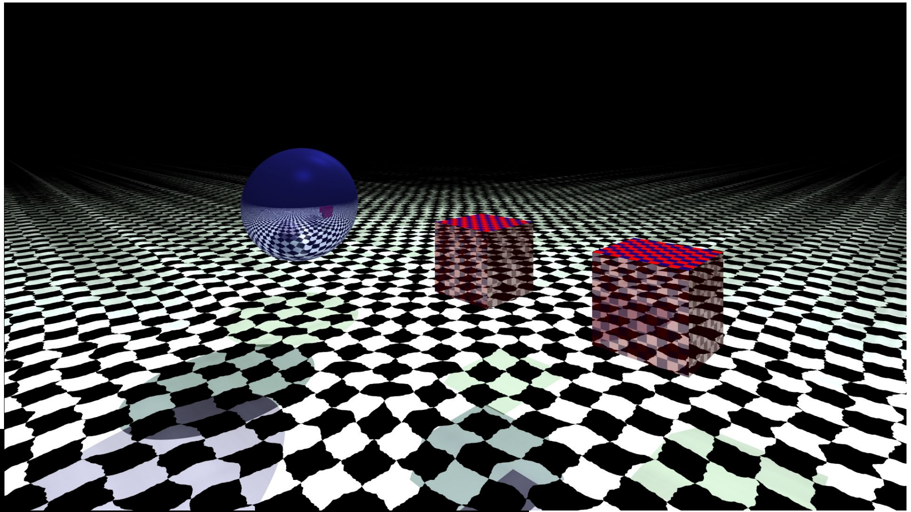
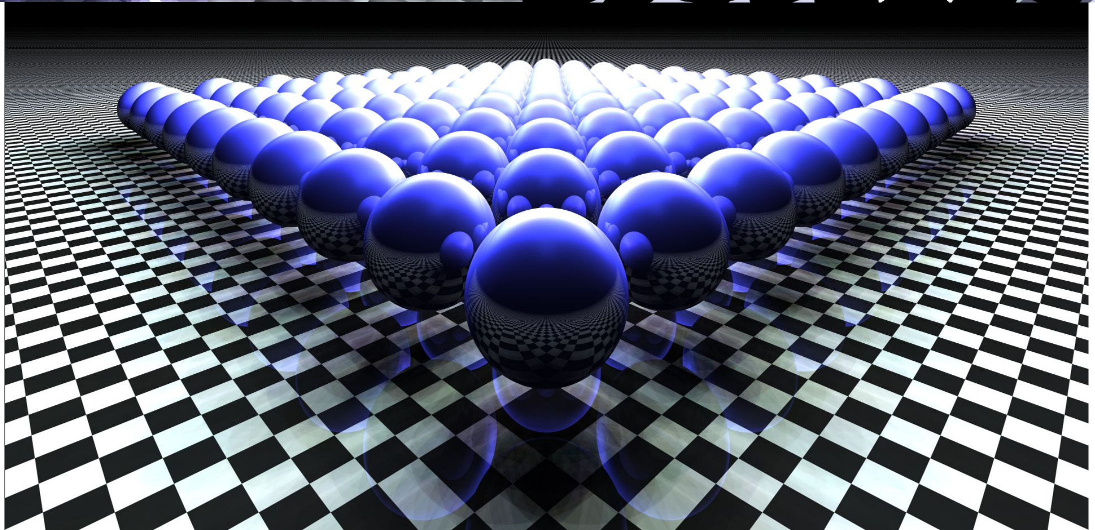
Ray tracing

CPU render core that uses ray tracing algorithm for render.

Main features:

- plane, sphere, CS objects, axis-align cubes, models render support
- multi thread render
- materials, material modificators
- reflections, shadows
- multicolor lighting

(C++, WinAPI) Summer 2022

Fractals

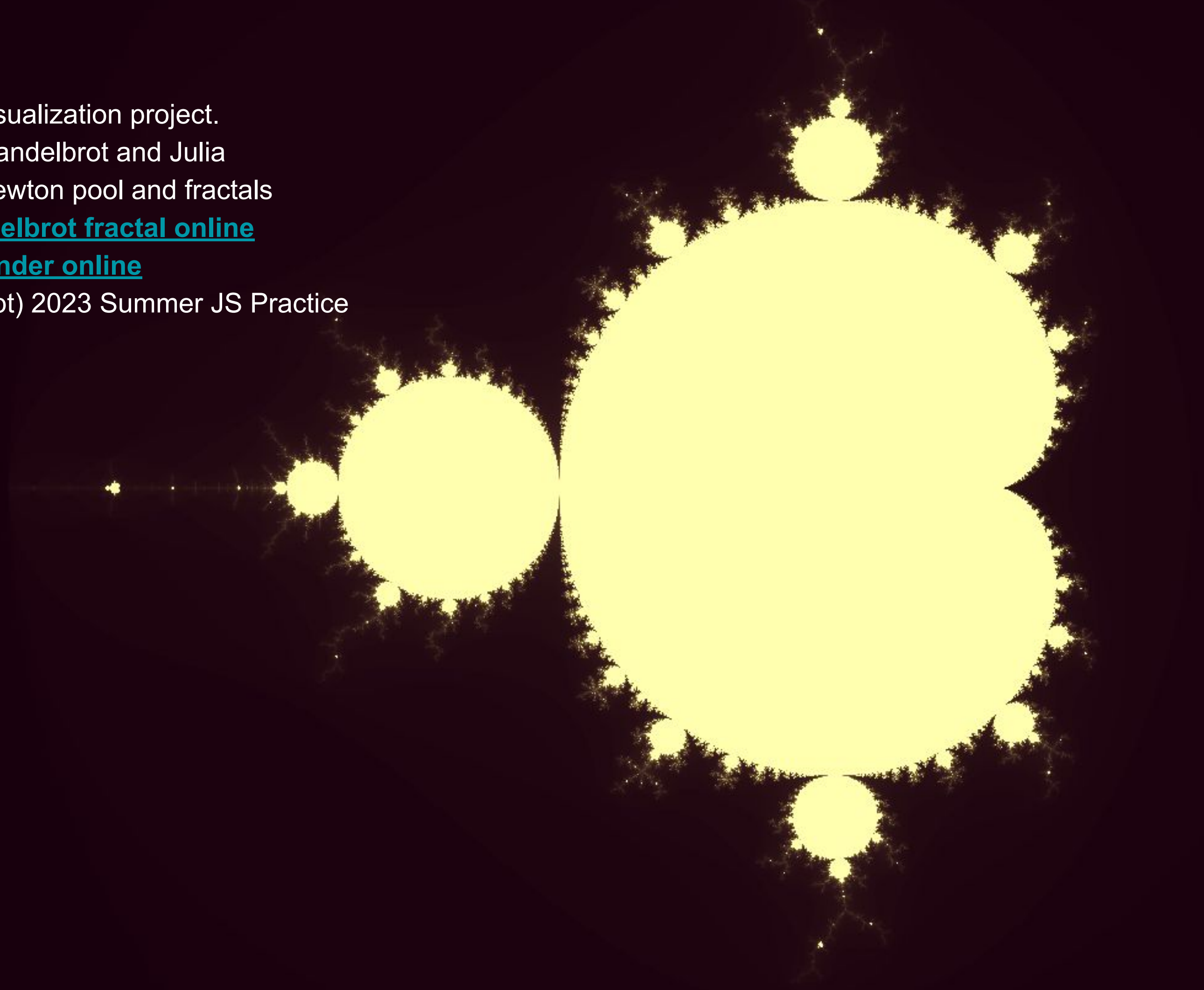
Fractal visualization project.

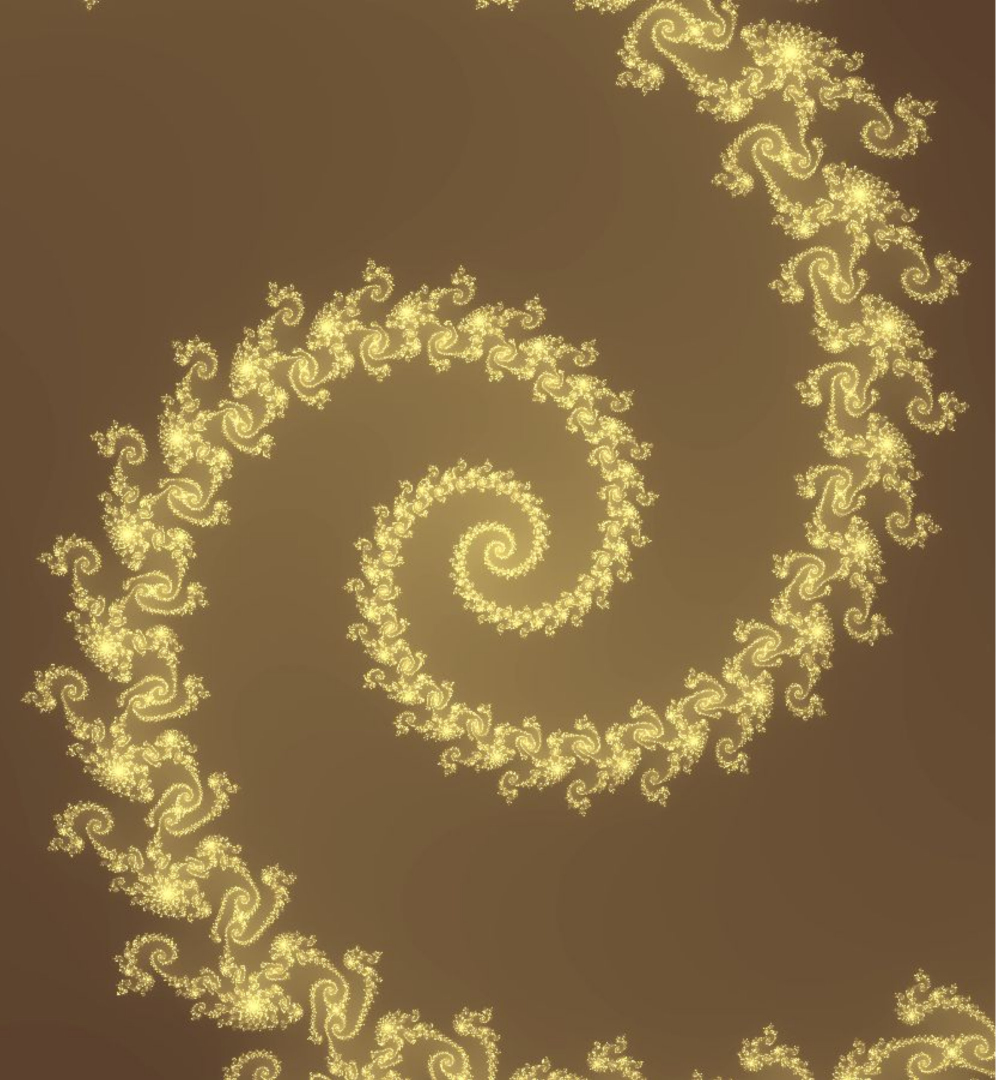
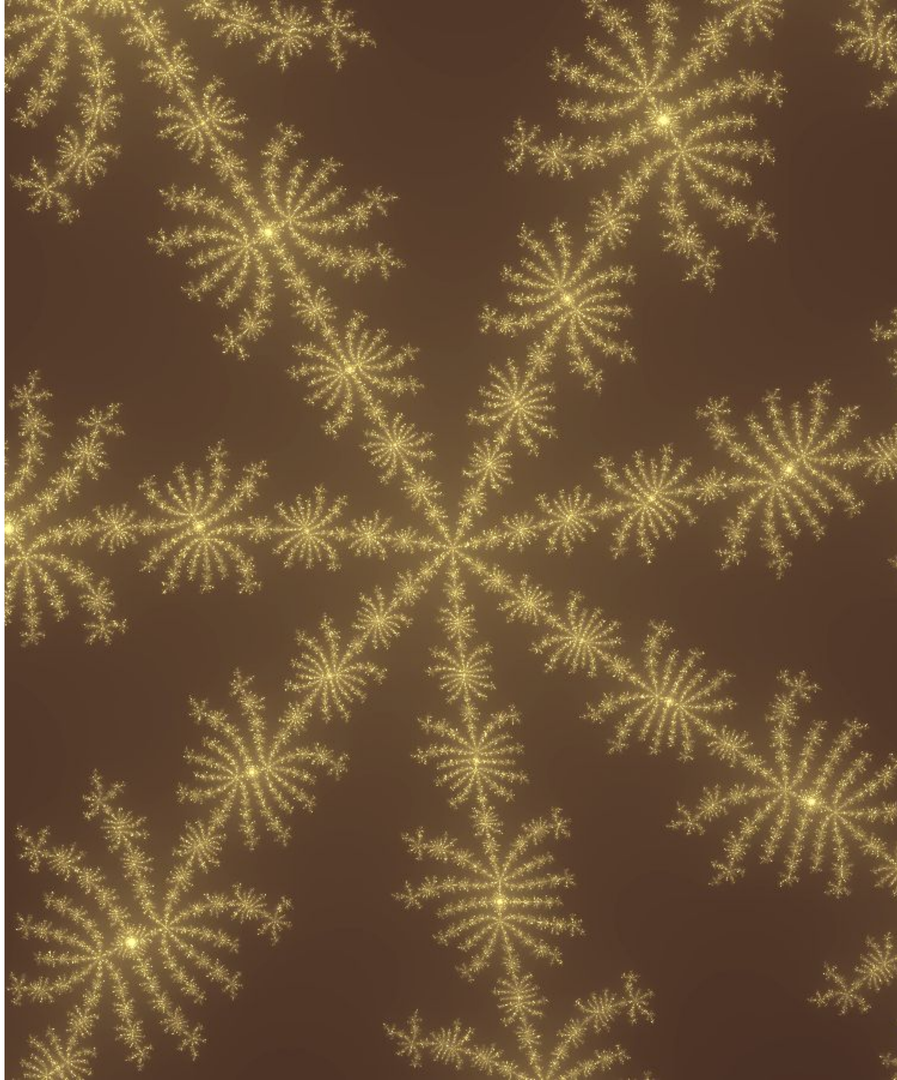
- Mandelbrot and Julia
- Newton pool and fractals

[My mandelbrot fractal online]

[My JS render online]

(Javascript) 2023 Summer JS Practice



Newton sinus fractal

(C++) Summer 2022

Game Life
(C++, OpenGL) 2021