SHADERFUNK

2021

Introduction to Shader development

h_da WS2020/21

Paul Nasdalack

info@paul-nasdalack.com

@littleBugHunter

About myself

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

Blue Byte
A UBISOFT® STUDIO

UBISOFT®

MARMOSET

INDIEBURG

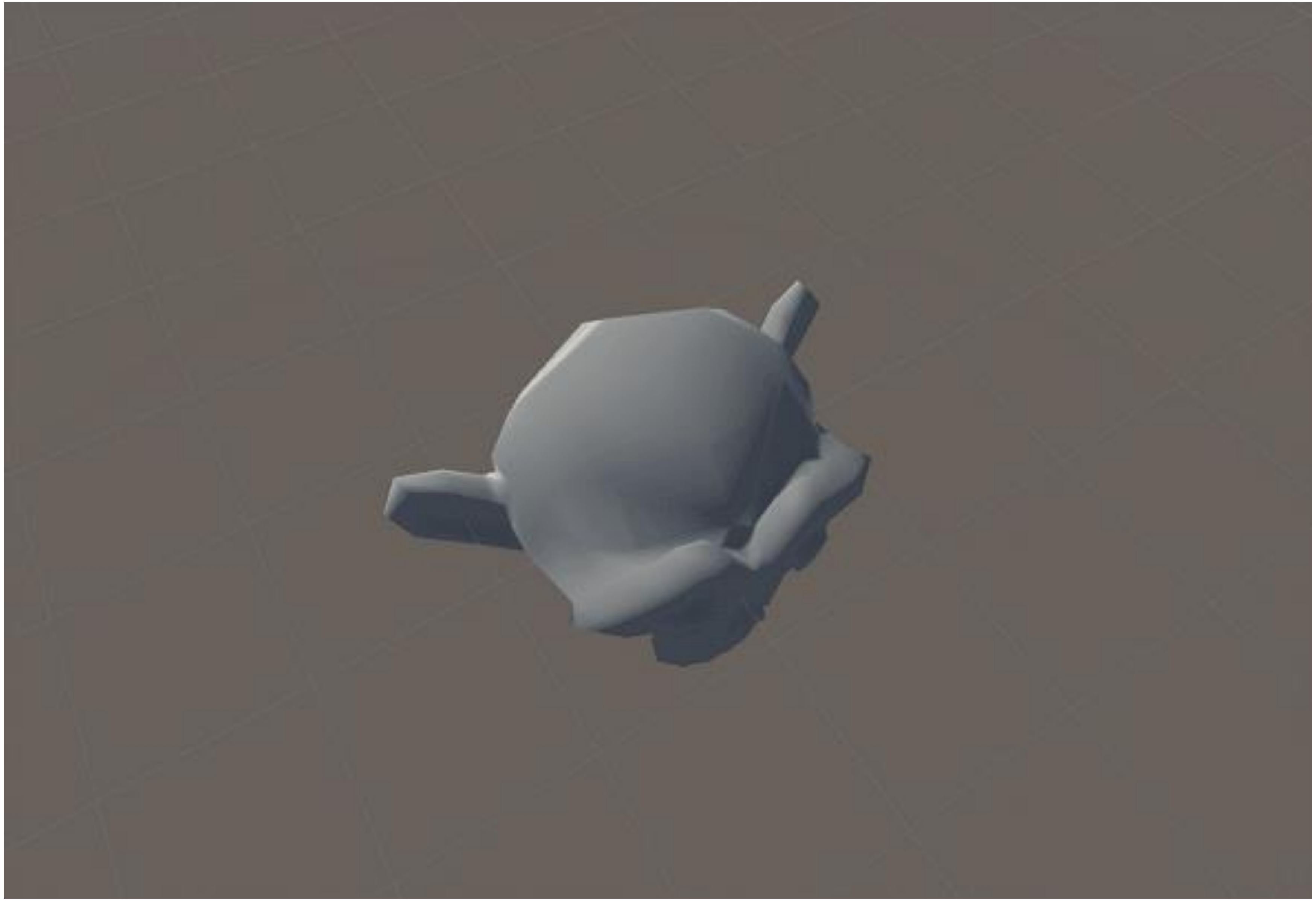
Nintendo
ENTERTAINMENT SYSTEM

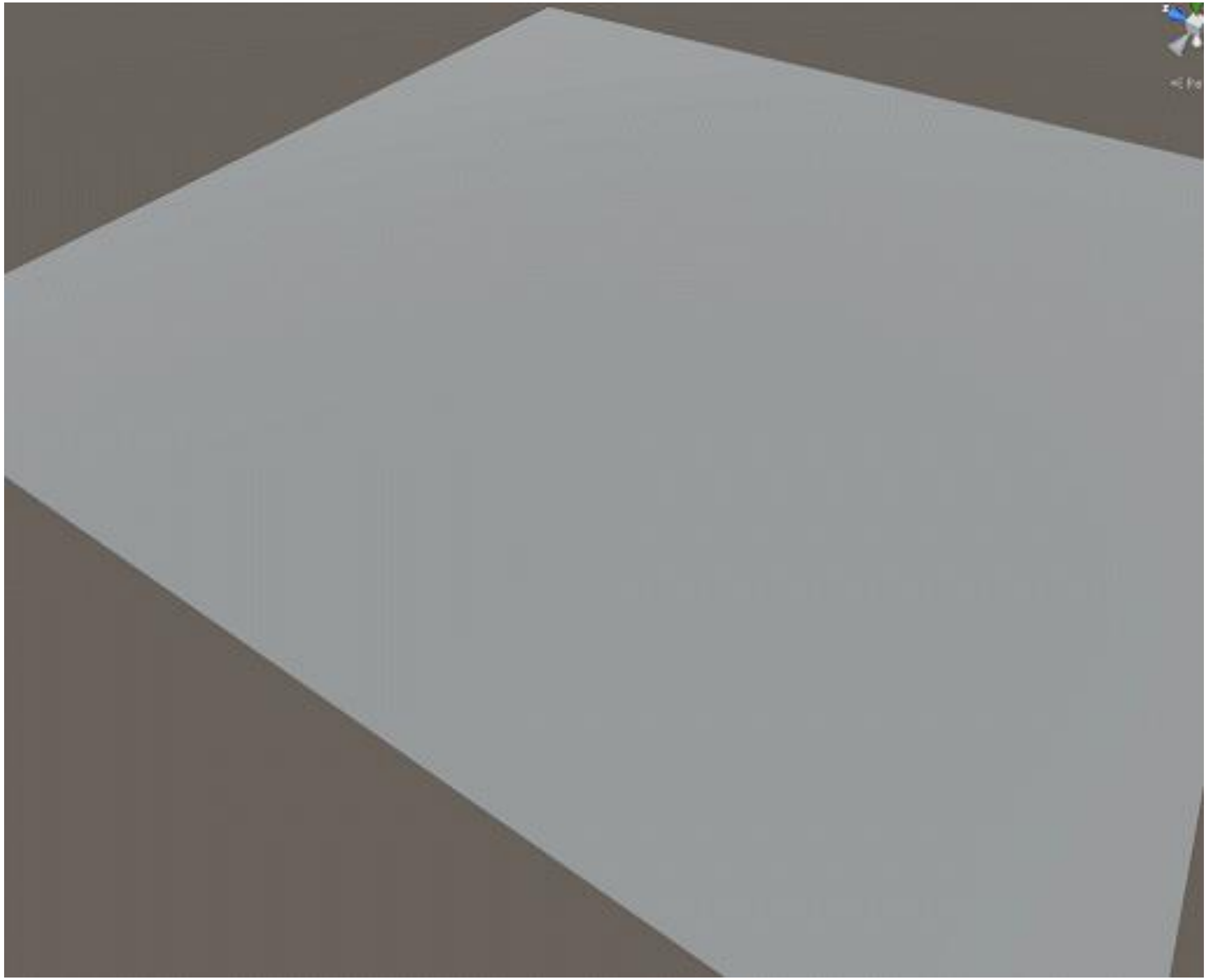
GAME BOY®

GAME BOY™
COLOR

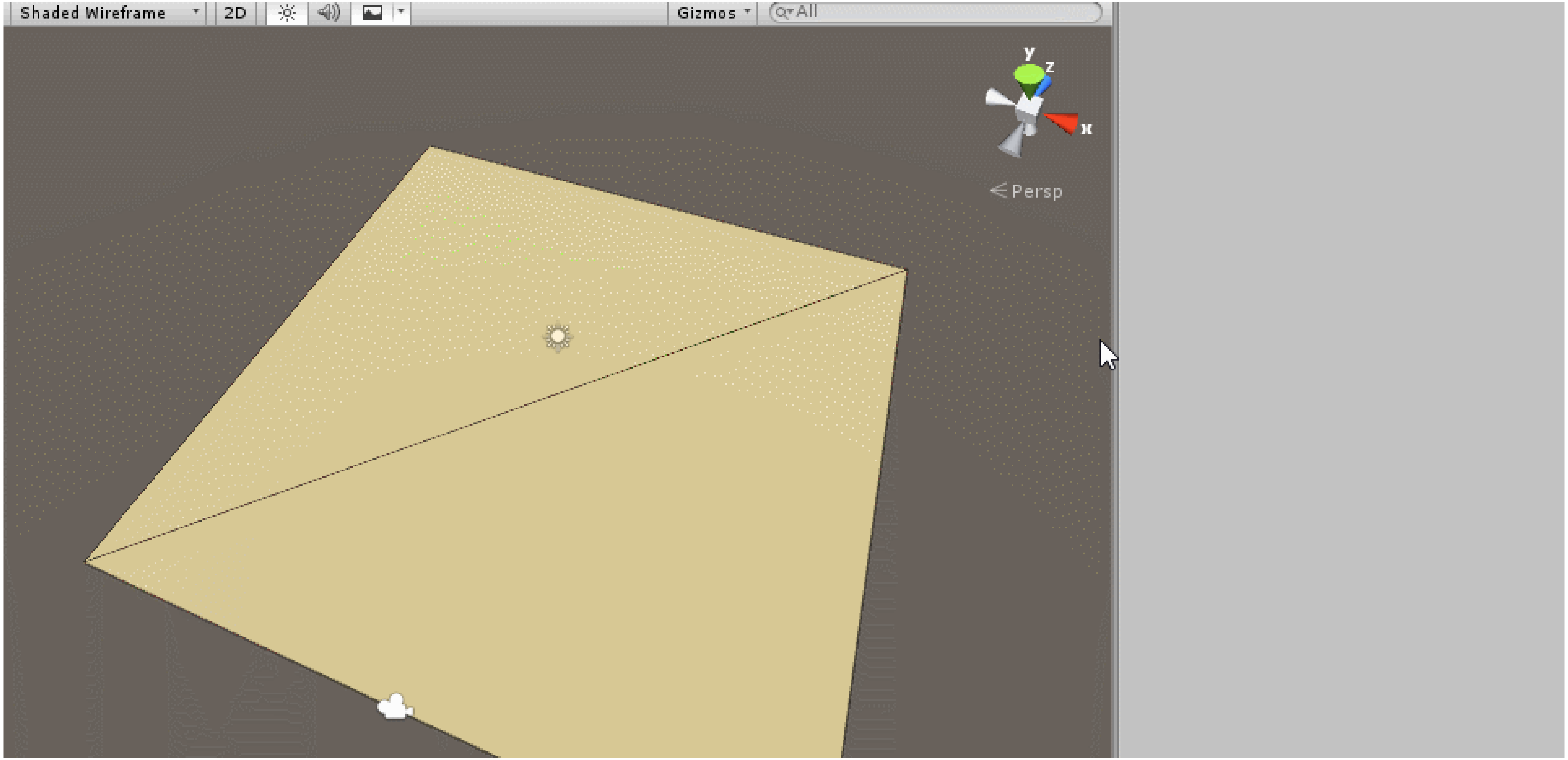
About myself

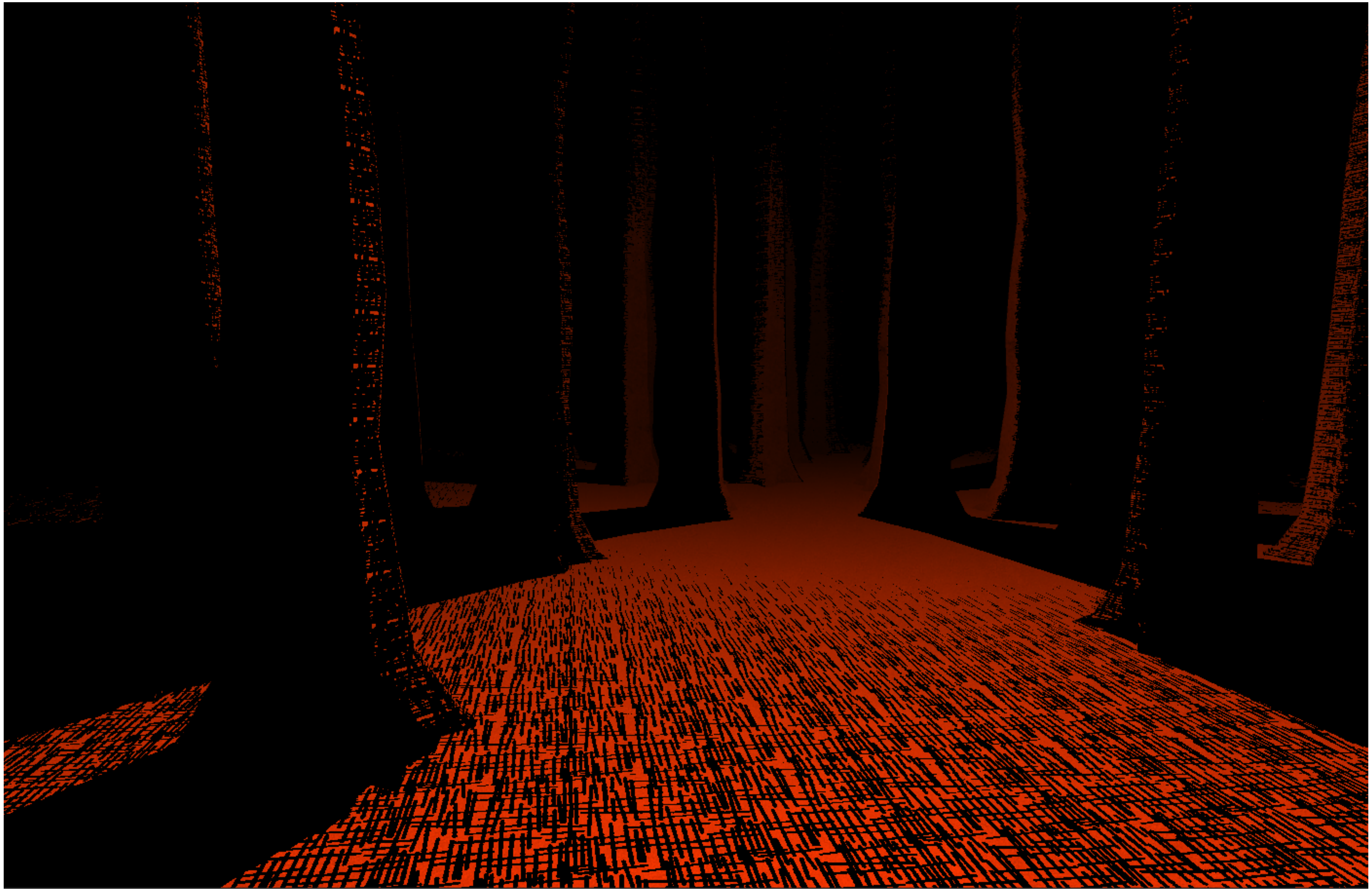
Some things we might be able to cover





z
Pe


Shaded Wireframe
2D
Gizmos
All
y
z
x
Persp



SHADERFUNK
2021

About this course

- Introduction to Graphics Pipeline and Shaders
- General fx tips and tricks
- Prototyping techniques
- We will be writing code!

- I’m counting on your input!

About this course

- Recap
- Presentation
- Prototyping in ShaderGraph
- Technical implementation in Code
- Creative Shader Building

Possible Takeaways for you

- Create your own shaders and effects
- Have a better understanding of the GPU
- Be able to optimize your own assets

Possible Takeaways for your team

- You’ll have a better understanding of other departments
- You’ll be able to handle your own shit
- You won’t cause Performance drops
- You’ll make the game BEAUTIFUL!!!!!!! *O*

What do you need

- Some knowledge about 3D development
- A computer/laptop during the lecture
- School math
  - Vectors
  - Matrices
  - We will do a quick recap on both
- Do an assignment at the end of the course

What about you?

- Why did you choose to attend this course?
- Do you have any experience with shaders and or coding?
- What do you expect from this course?

Structure

- Introduction (we just did this)
- What is a GPU and why do we need it
- How does 3d hardware work
- How does a shader work

What is a GPU and why do we need it?

aka: a brief history of graphics hardware

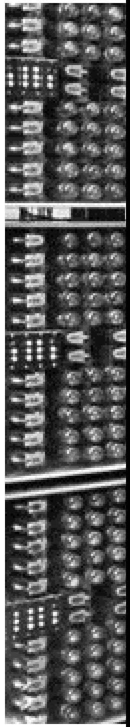
DISCLAIMER!!!!

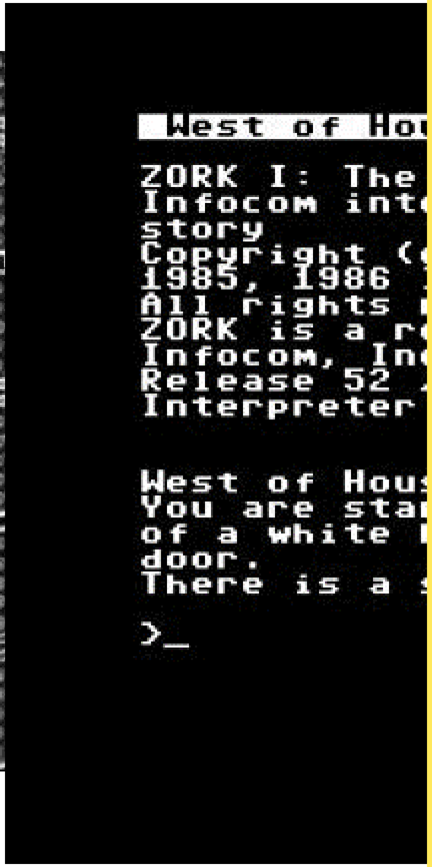
- This is not a history lesson
- A lot of things will be very simplified

What is a GPU and why do we need it?

aka: a brief history of graphics hardware

- In the beginning there was the CPU

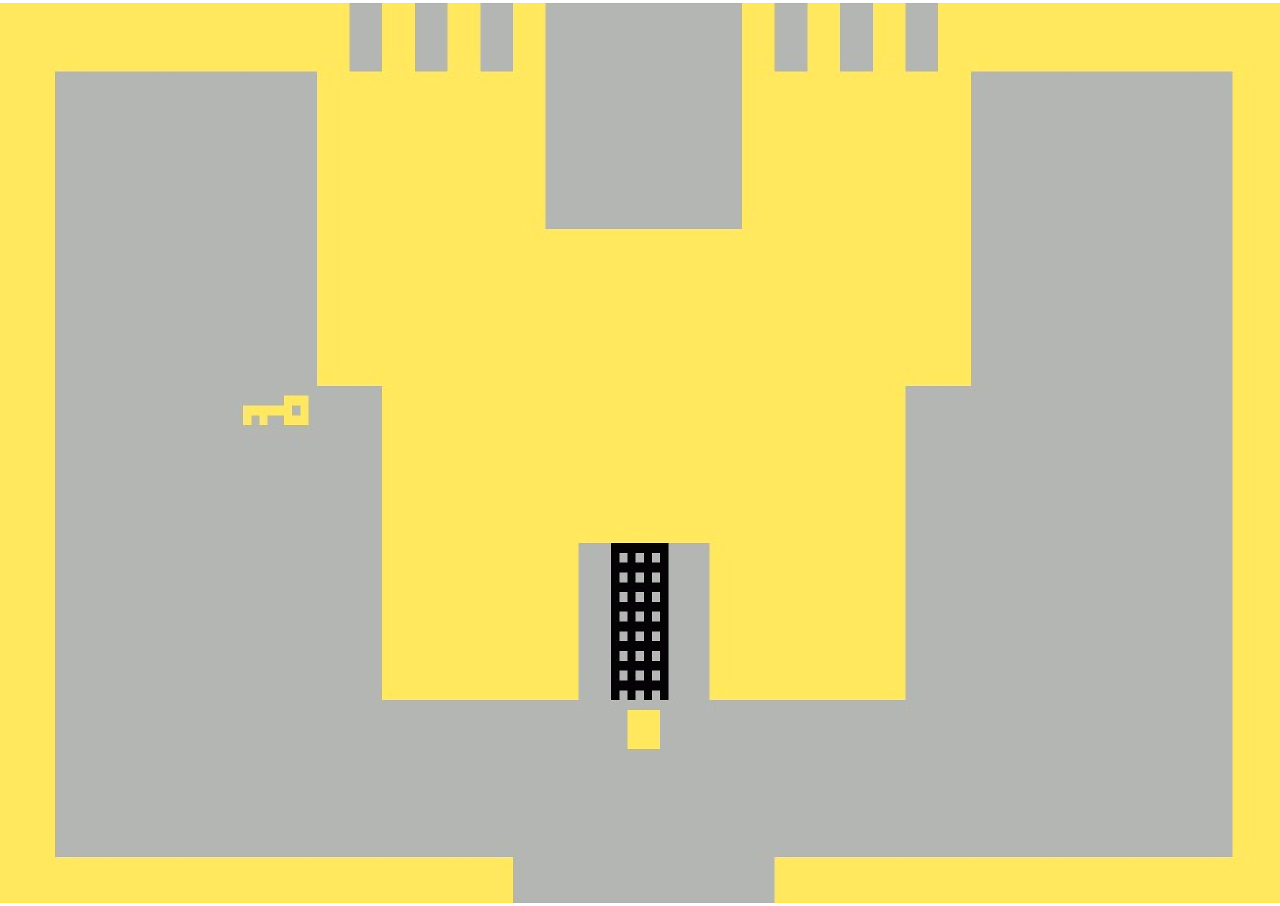




```
 West of Hou
ZORK I: The
Infocom inte
story
Copyright (
1985, 1986 
All rights 
ZORK is a r
Infocom, Inc
Release 52 
Interpreter

West of Hous
You are star
of a white 
door.
There is a 

>_
```



What is a GPU and why do we need it?

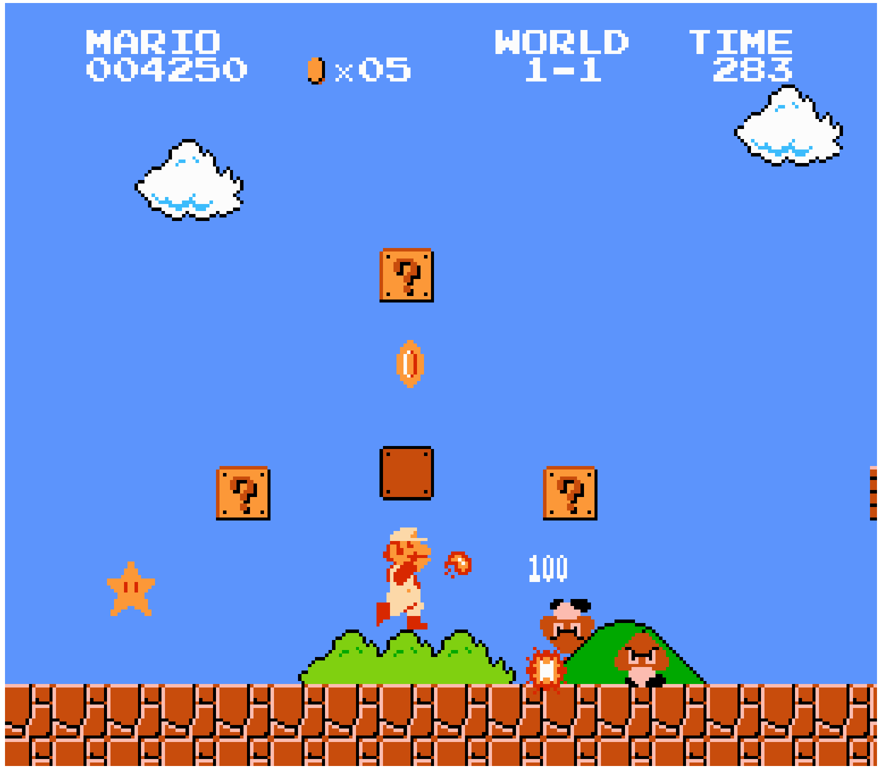
aka: a brief history of graphics hardware

- Only text support
- We want direct pixel access!!!!!
- 320x240px = 76800 pixels 60 times a second
  - (or 50 times, if you live in Europe)
- That’s 4,608,000 pixels per second!
- CPUs back then could only do 1,000,000-2,000,000 operations per second
- FUCK!

What is a GPU and why do we need it?

aka: a brief history of graphics hardware

- Wait a minute the NES had more than text!

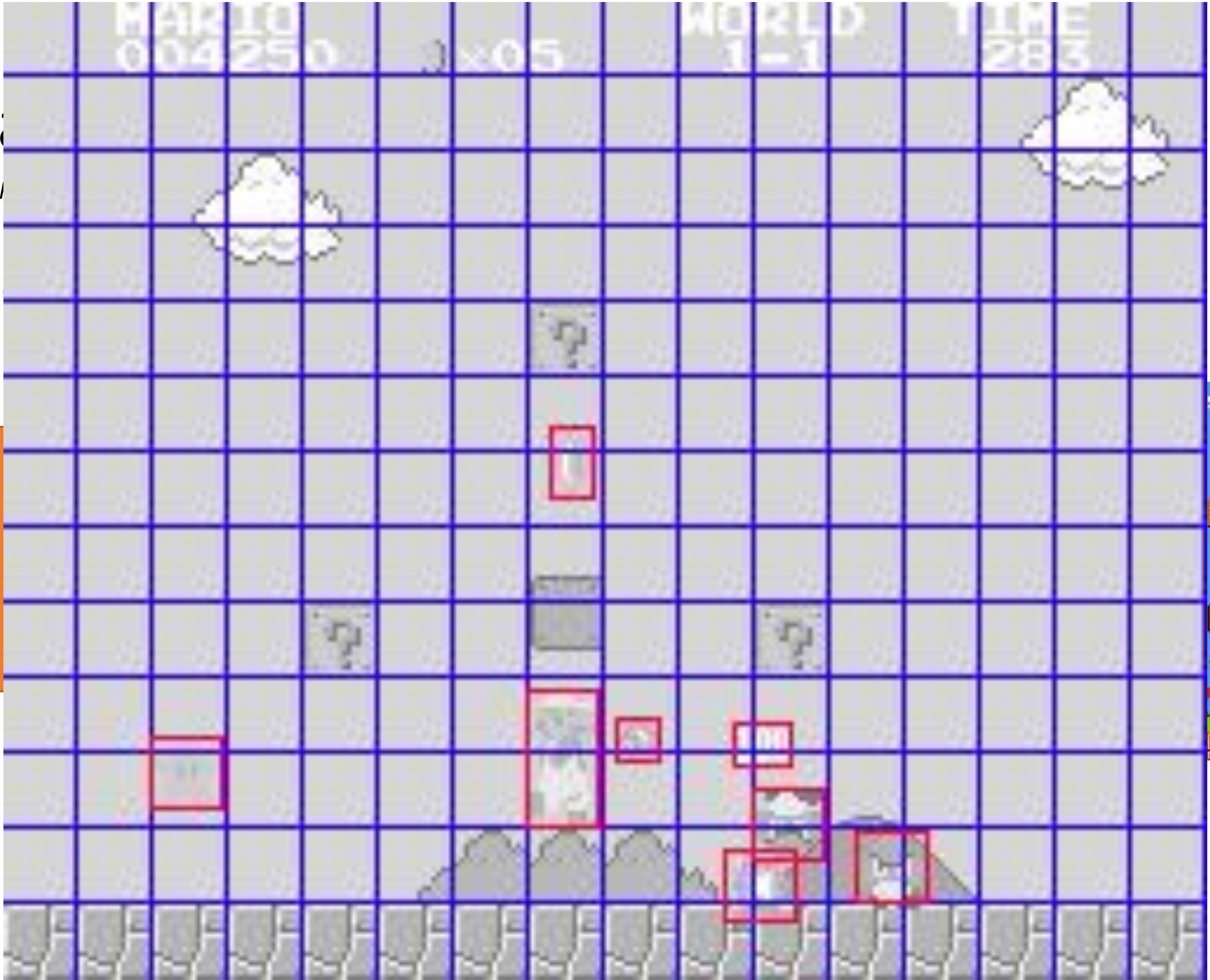


Wha

aka: a bi

- BEH

CPU



What is a GPU and why do we need it?

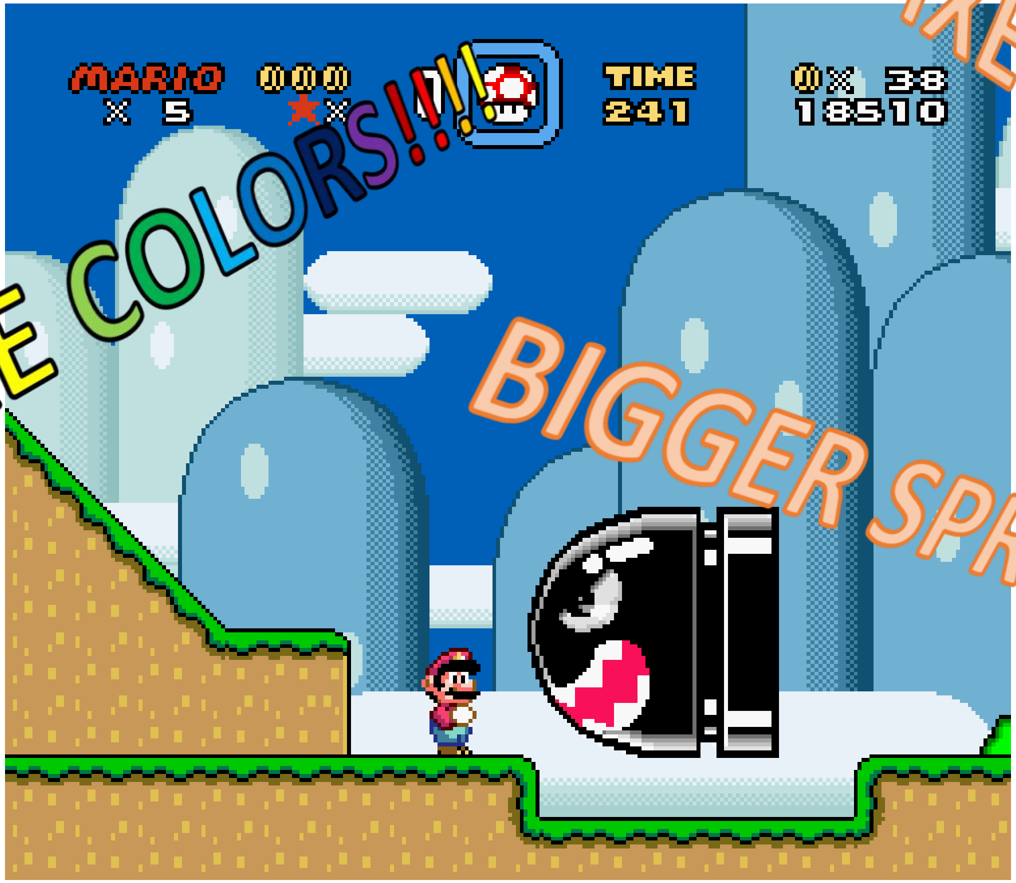
aka: a brief history of graphics hardware

- PPU is specifically designed to render pixels
- It is fixed to always render a background and a bunch of sprites
- Can’t do anything else!
- CPU prepares all the “level data” and sprite positions, the PPU just talks to the screen

What is a GPU and why do we need it?

aka: a brief history of graphics hardware

- BEHOLD THE SNES!



MORE COLORS!!!!

MORE PIXELS!!!!

BIGGER SPRITES!!!!

What is a GPU and why do we need it?

aka: a brief history of graphics hardware

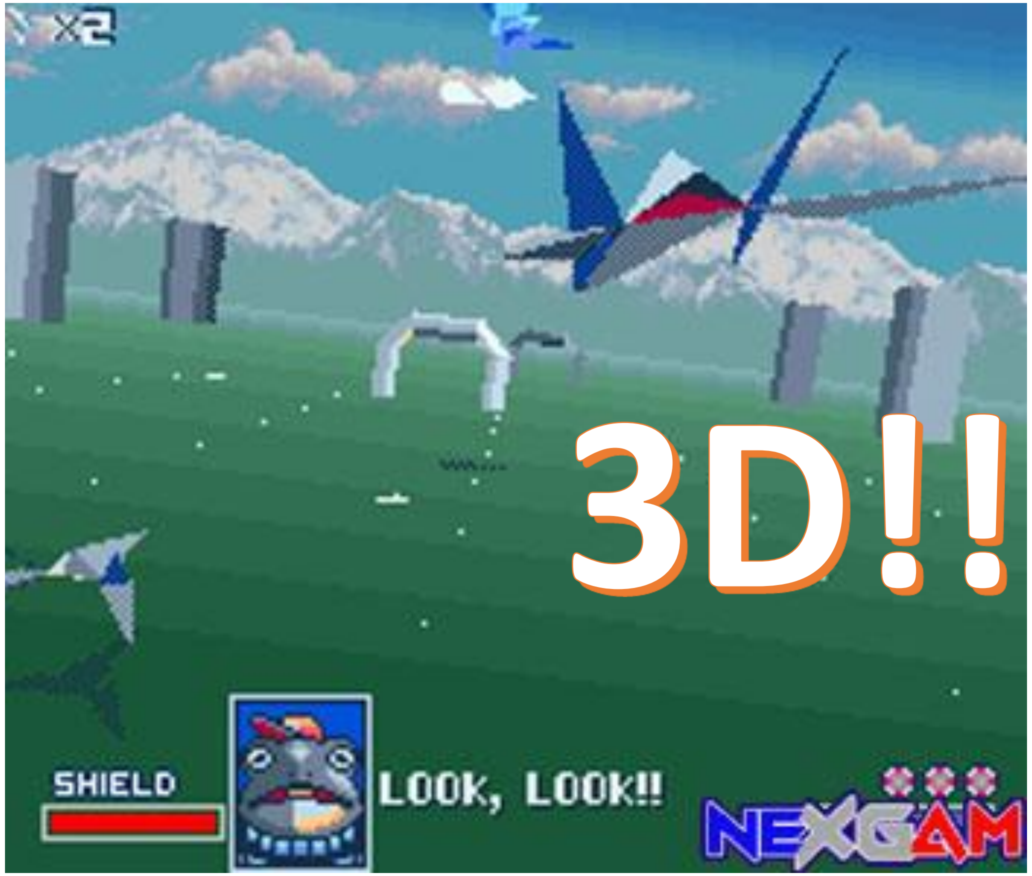
- BEHOLD THE SNES!

BUT MOST IMPORTANTLY…

What is a GPU and why do we need it?

aka: a brief history of graphics hardware

- BEHOLD THE SNES!



3D!!!!!

What is a GPU and why do we need it?

aka: a brief history of graphics hardware

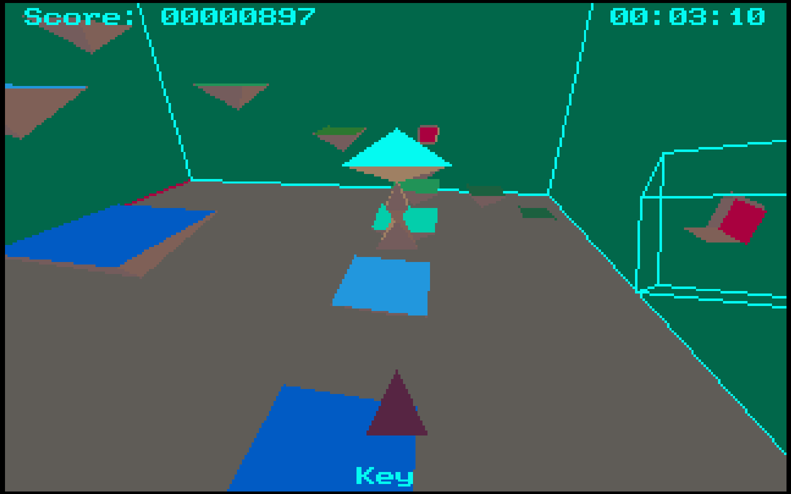
- SuperFX Chip
  - Separate (faster) CPU on the game cartridge
  - Very fast at multiplications
  - Could render arbitrary Bitmaps (not tile data)

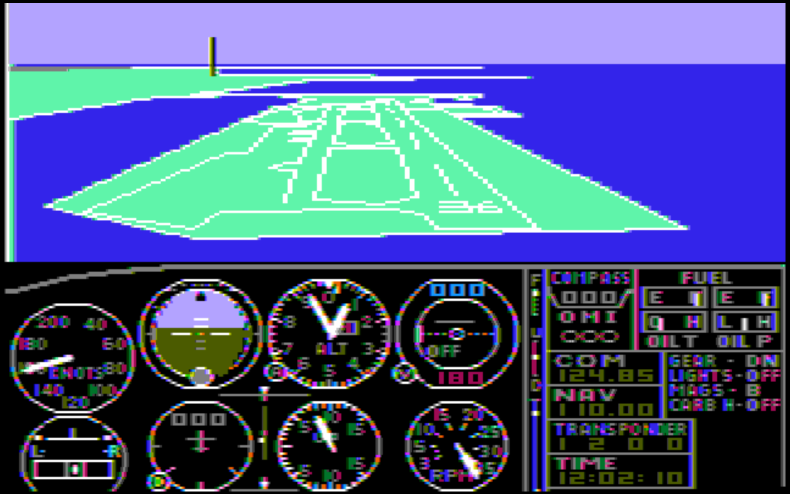
What is a GPU and why do we need it?

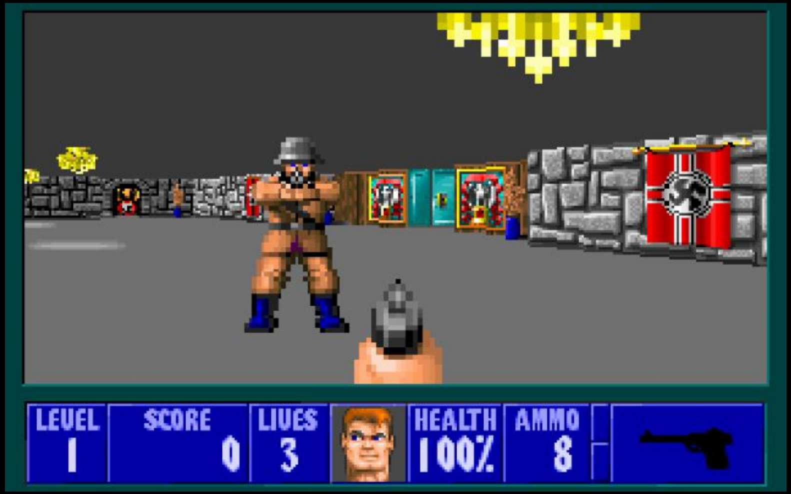
aka: a brief history of graphics hardware

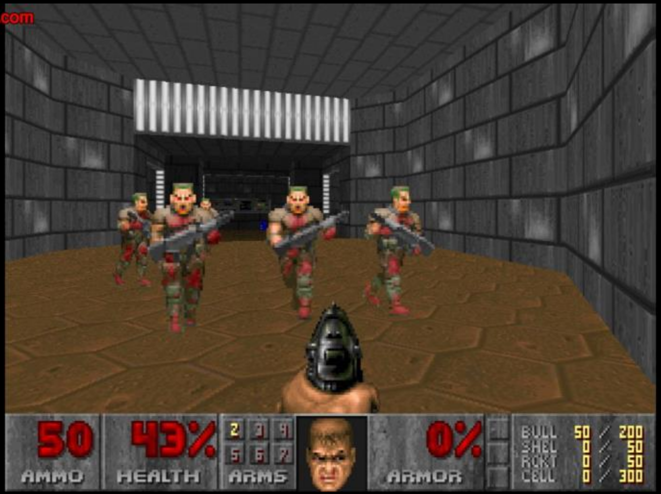
- Having a separate CPU on the cartridge sucks
  - You have to put one on each game, you sell
- Let‘s build it into the console/computer (the GPU was born)

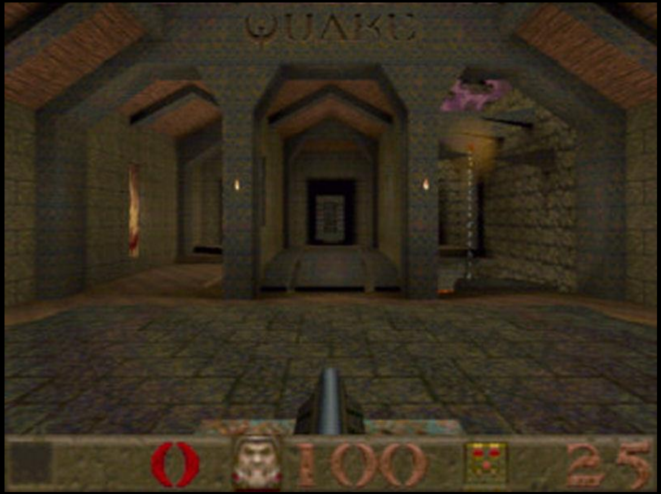
Well… Actually


Score: 00000897
00:03:10
Key


COMPASS
FUEL
OMI
OIL T OIL P
COM 124.85
NAV 110.00
TRANSPONDER 1200
TIME 12:02:10
GEAR - DN
LIGHTS - OFF
MAGS - B
CARB H - OFF
KNOTS
ALT
RPM


LEVEL 1
SCORE 0
LIVES 3
HEALTH 100%
AMMO 8


50 AMMO
43% HEALTH
2 3 4 5 6 7 ARMS
0% ARMOR
BULL 50 / 200
SHEL 0 / 50
RCKT 0 / 50
CELL 0 / 300


QUAKE
0
100
25

What is a GPU and why do we need it?

aka: a brief history of graphics hardware

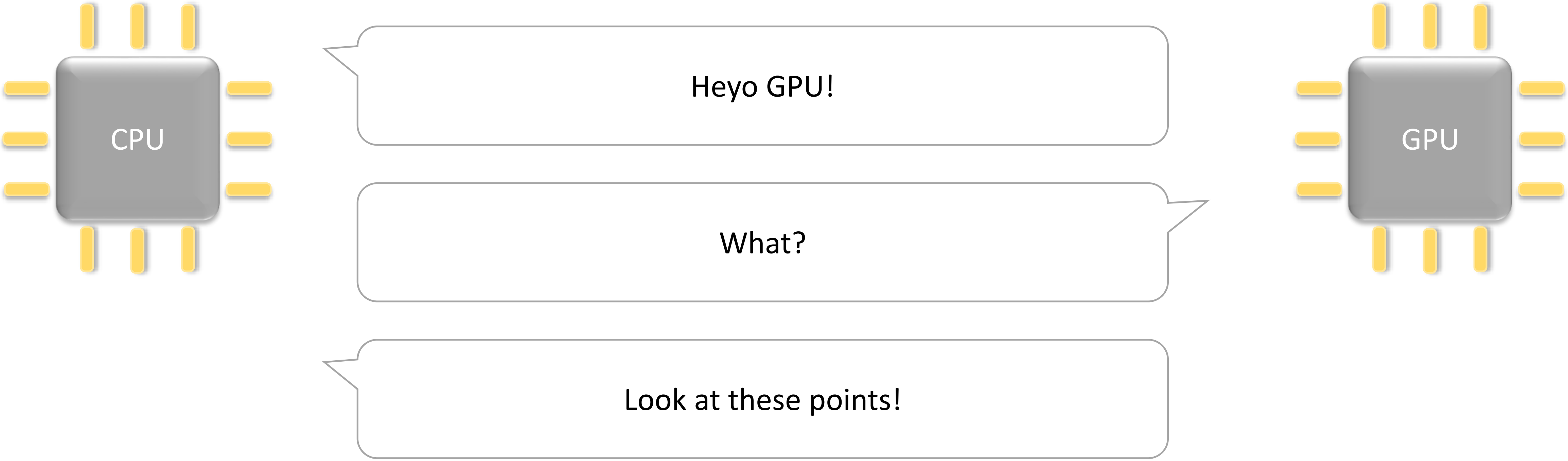
- When consoles came with builtin Graphcis Processing Units there was no going back
- GPUs took care of:
  - Perspective distortion of 3d points (vertices)
  - Calculating the triangles and filling their pixels
  - Mapping textures onto the triangles

What is a GPU and why do we need it?

aka: a brief history of graphics hardware

- Shaders were super simple
- You could:
  - Send a bunch of vertices to the GPU
  - Tell it to connect vertices to triangles
  - And maybe slap a texture onto it

How does 3d hardware work (abridged)

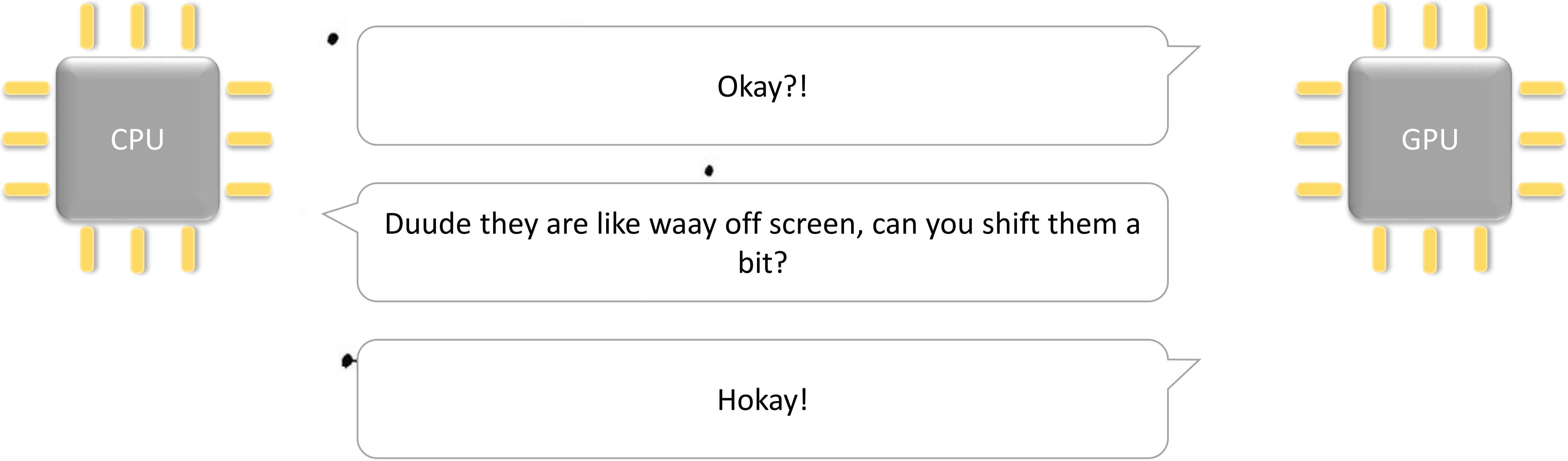


Heyo GPU!

What?

Look at these points!

How exactly does 3d hardware work

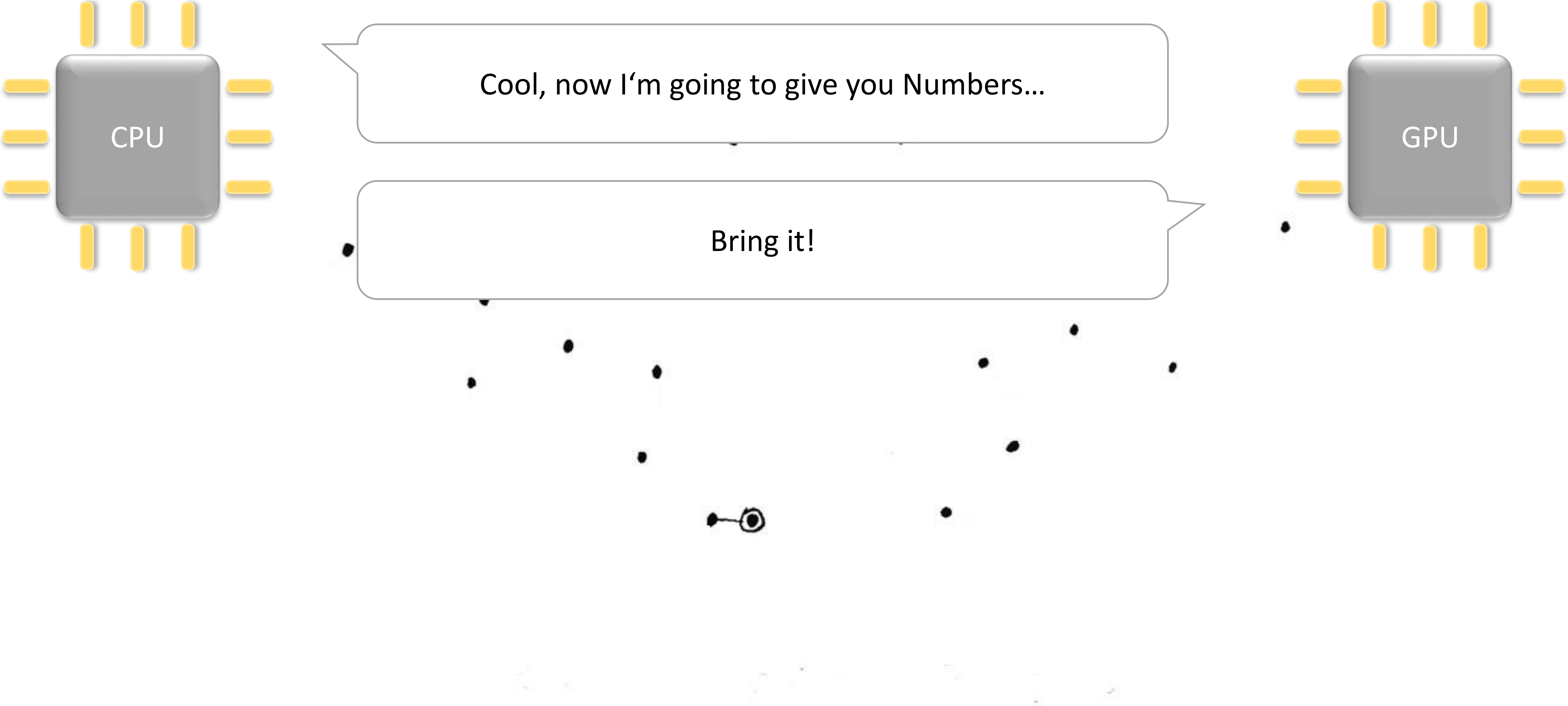


Okay?!

Duude they are like waay off screen, can you shift them a bit?

Hokay!

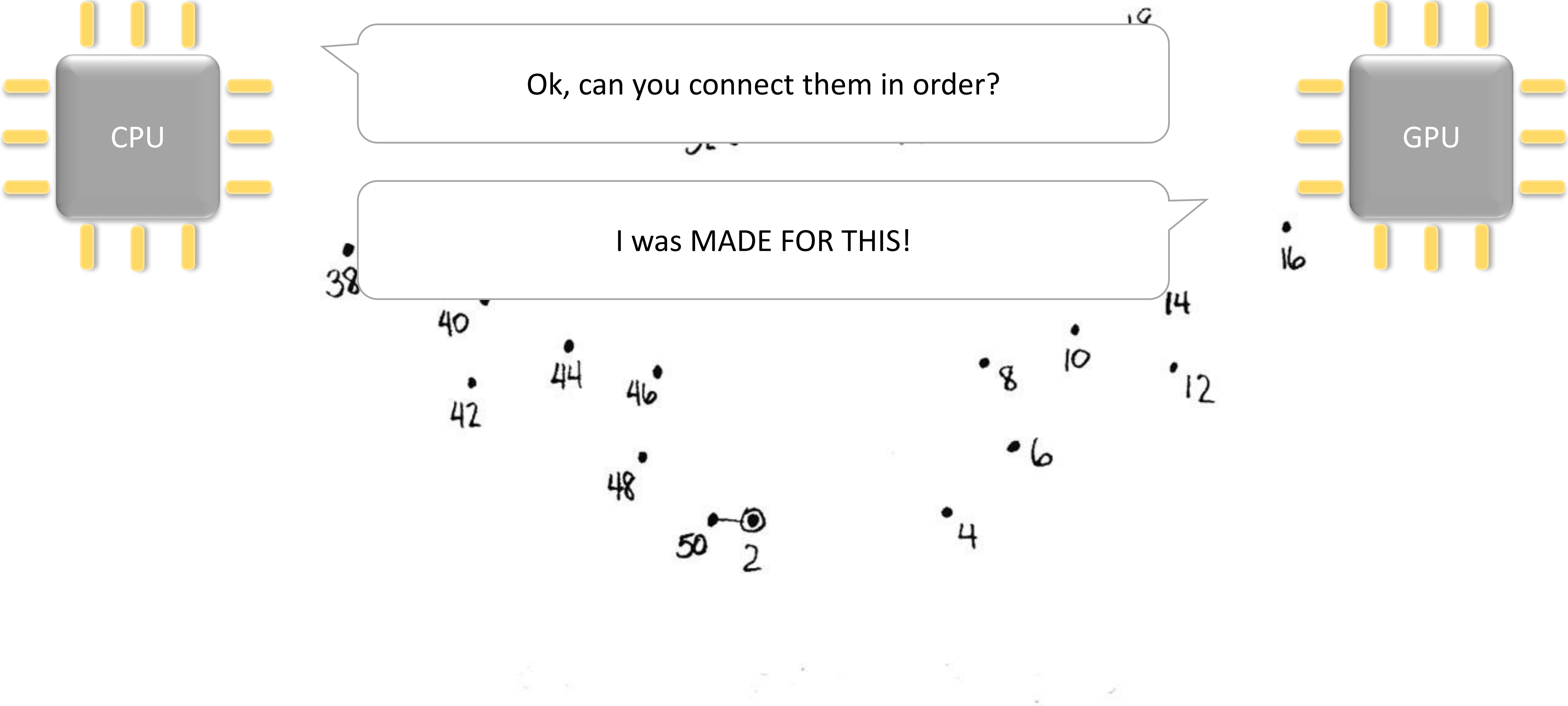
How does 3d hardware work (abridged)



Cool, now I‘m going to give you Numbers…

Bring it!

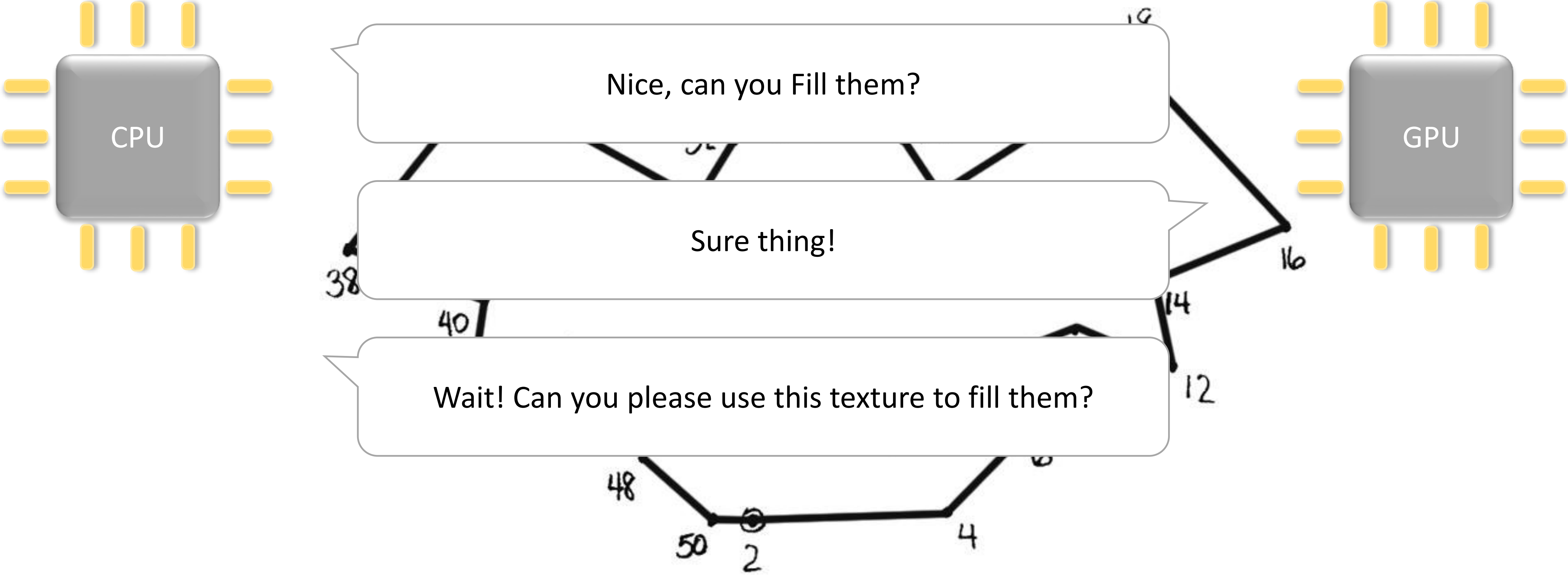
How does 3d hardware work (abridged)



Ok, can you connect them in order?

I was MADE FOR THIS!

How does 3d hardware work (abridged)

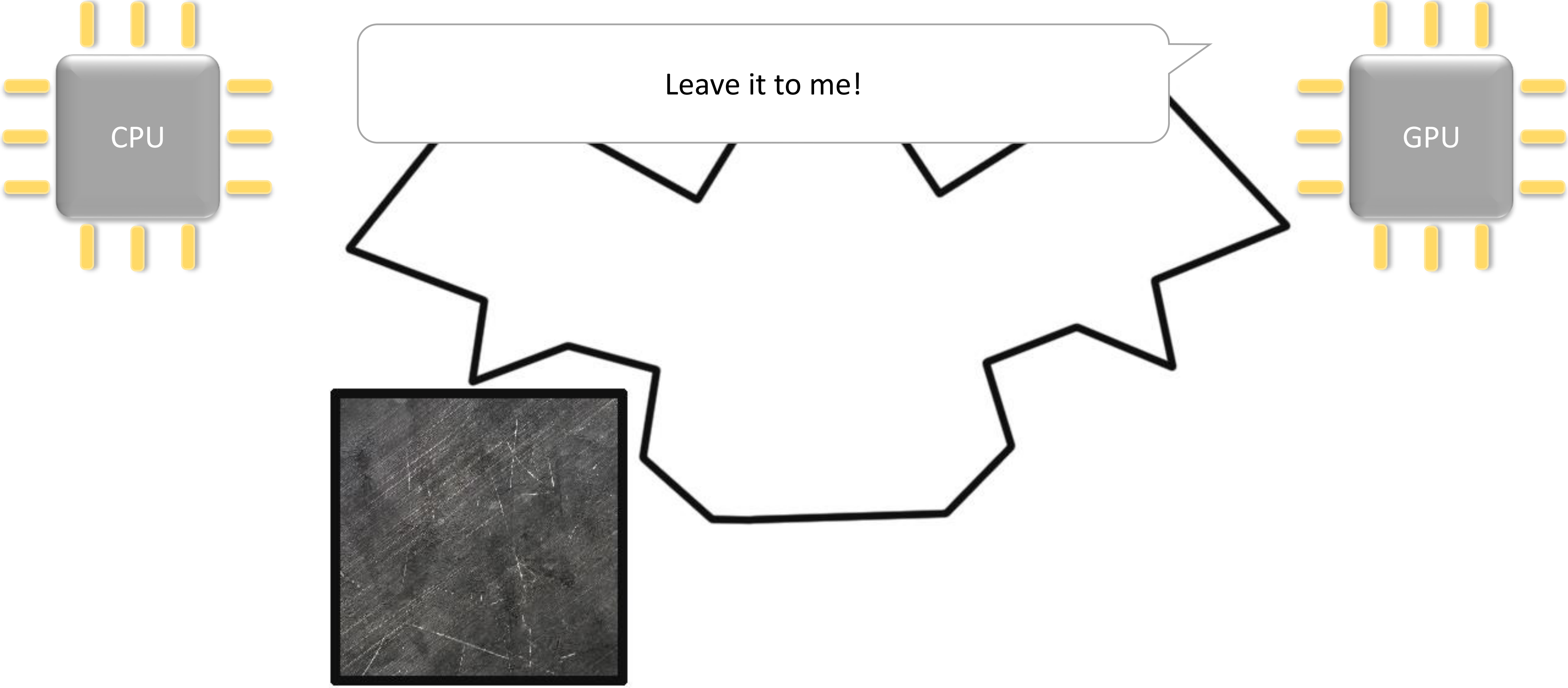


Nice, can you Fill them?

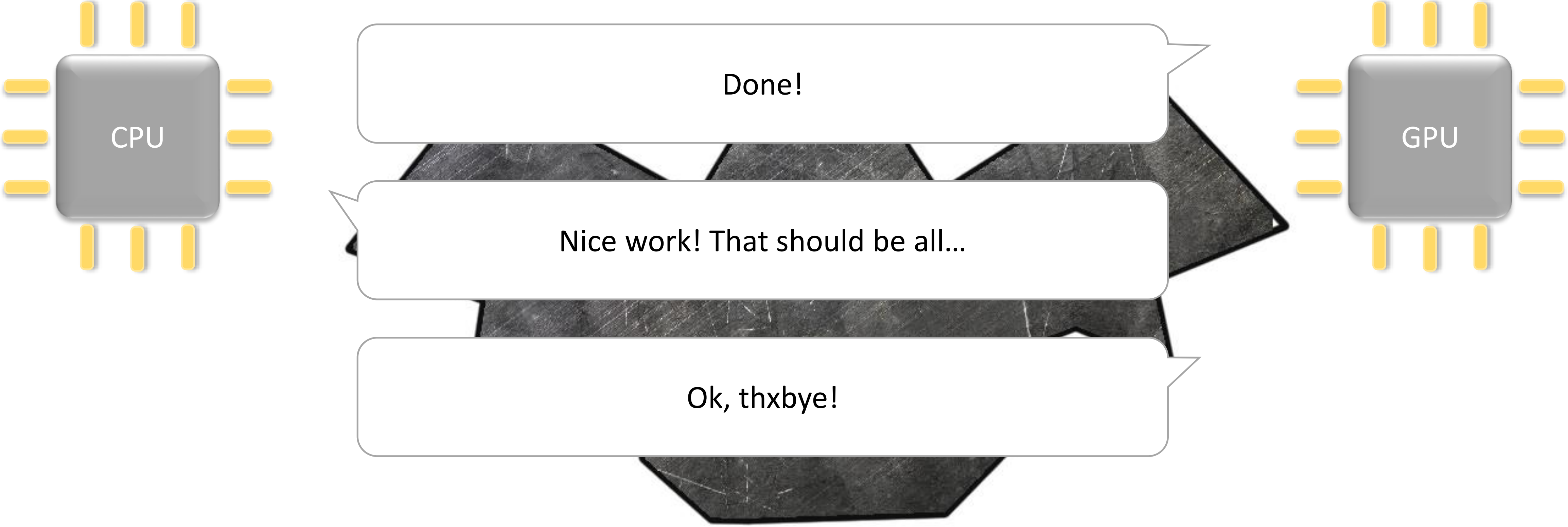
Sure thing!

Wait! Can you please use this texture to fill them?

How does 3d hardware work (abridged)



How does does 3d hardware work (abridged)



Done!

Nice work! That should be all…

Ok, thxbye!

How does 3d hardware work (abridged)

- You could also:
  - Choose different basic lighting methods
  - Sometimes fancy predefined effects, like wobbling the texture

How does 3d hardware work (abridged)

- People wanted more, for example
  - Move vertices around separately
  - Make their own cool unique texture effects
  - Do procedural effects, with the POWER OF MATH!!!

How does 3d hardware work (abridged)

SHADERS WERE BORN!

How does 3d hardware work (abridged)

- Shaders come in two parts:
  - Vertex shader
  - Fragment shader

Vertex Shader

- Moves the vertices to the correct position on screen
- Also does things like:
  - Vertex displacement
  - Precalculation of complex math for fragment shader

```
v2f vert (vertexData v)
{
    v2f o;
    o.vertex = mul(MVP_MATRIX, v.vertex);
    o.uv = v.uv;
    return o;
}
```

Fragment Shader

- Converts „Fragments“ to final pixel colors on screen
- Calculates colors based on:
  - Textures
  - Vertex Colors
  - Math
  - Complex Math prepared inside the vertex shader

```
fixed4 frag (v2f i) : SV_Target
{
        fixed4 col = tex2D(_Texture, i.uv);
        return col;
}
```

How exactly does 3d hardware work

Thats it.

Questions?

Assignment!

- Find cool effects in games and movies you like or come up with your own
- We‘re going to look at how they were done
- You have 15 Minutes!
- GO!