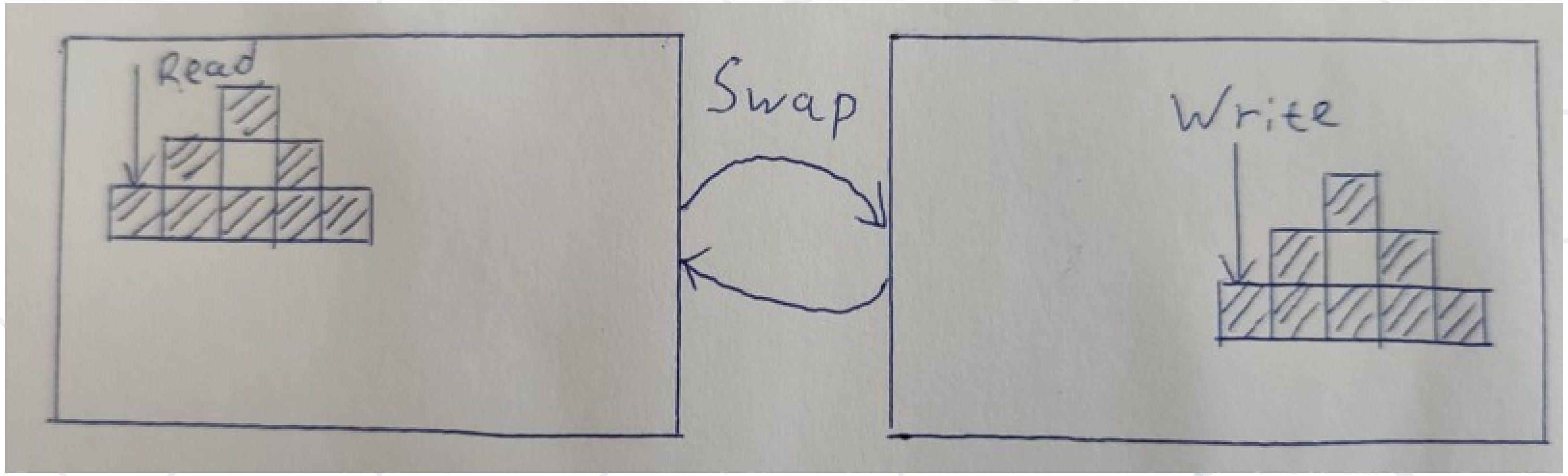
Double Buffering



Zhivko Petrov

A guy that knows C++

Software University

SoftUni

Software University

https://about.softuni.bg

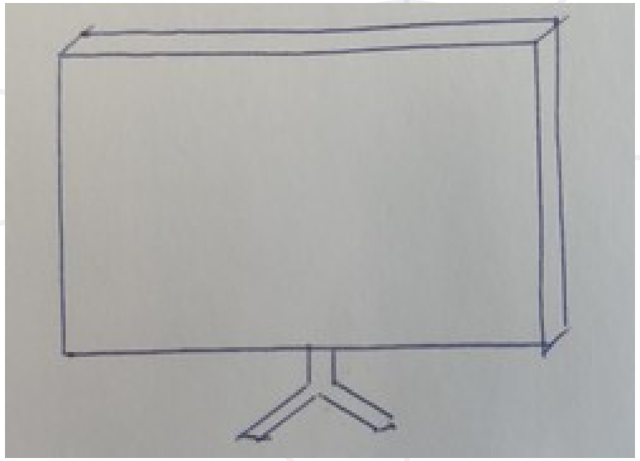
Have a Question?

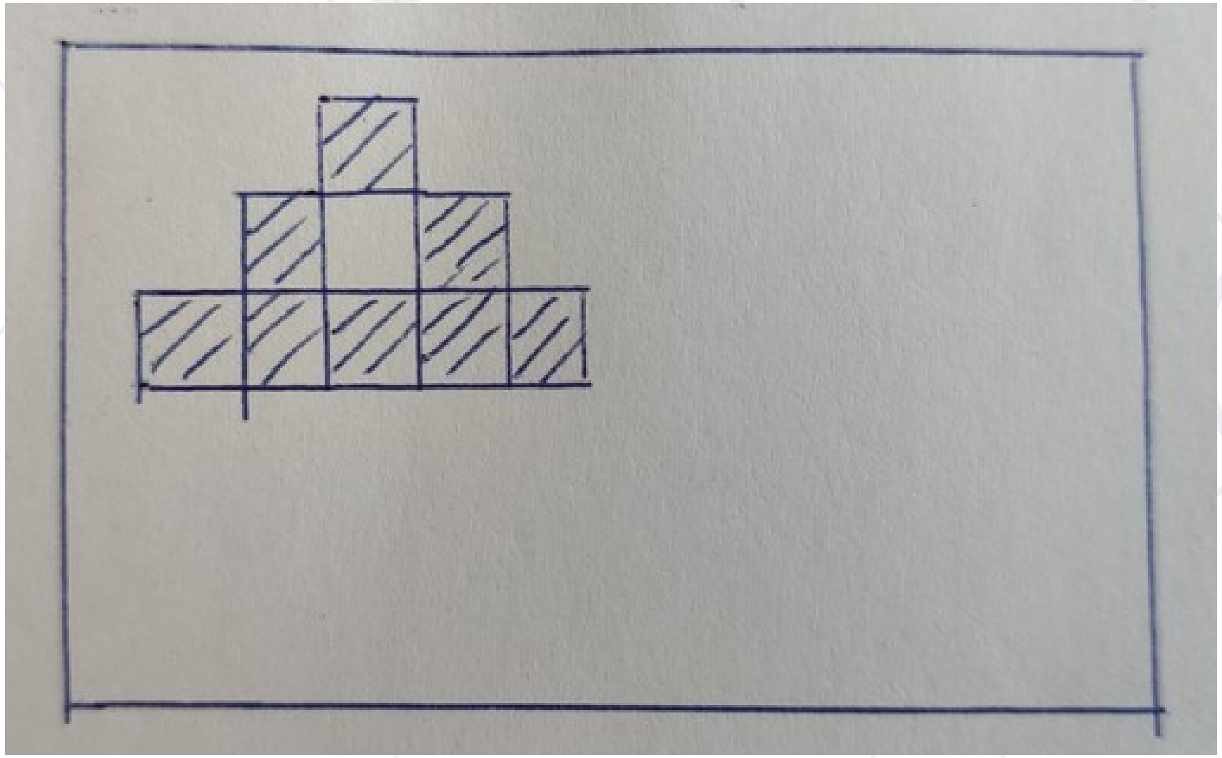
sli.do

#app-dev-cpp

Screen Rendering

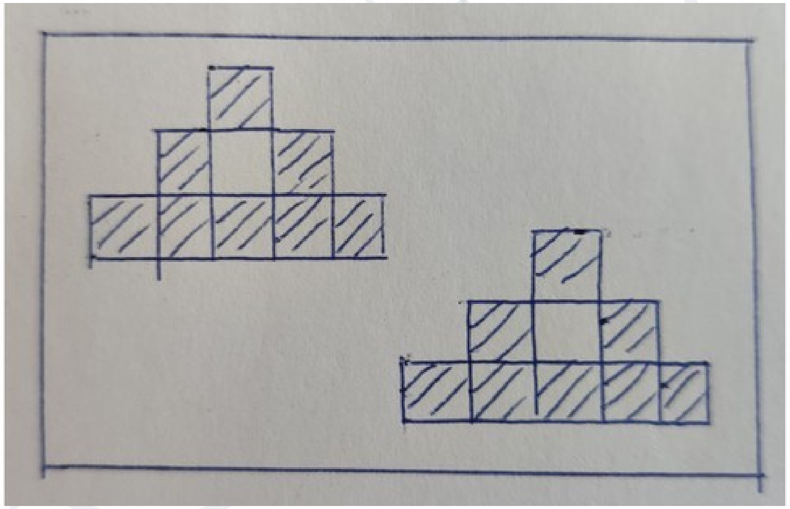
- The purpose of the UI is to visualize graphics on the screen
- Screens are constructed as a matrix of **pixels**
- Pixels are drawn from top-left to bottom-right

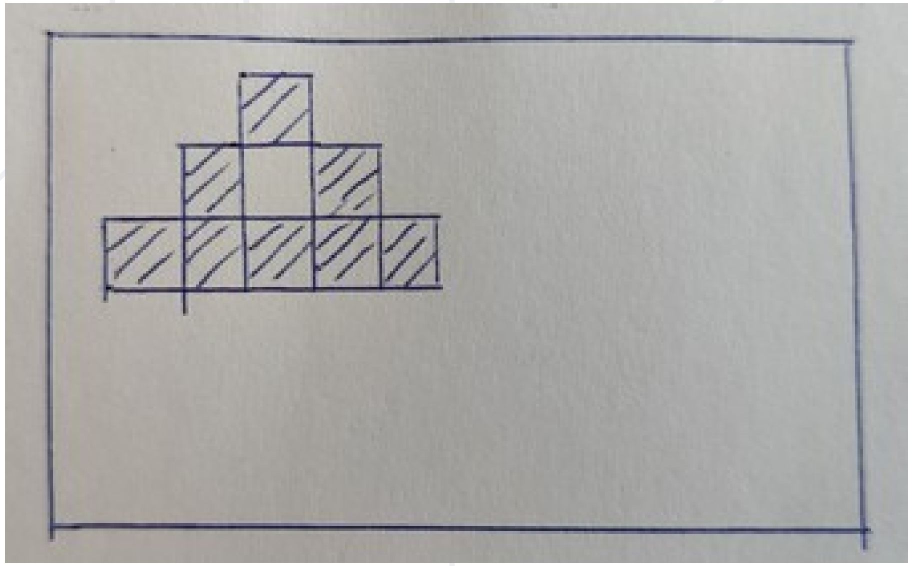




Rendering order

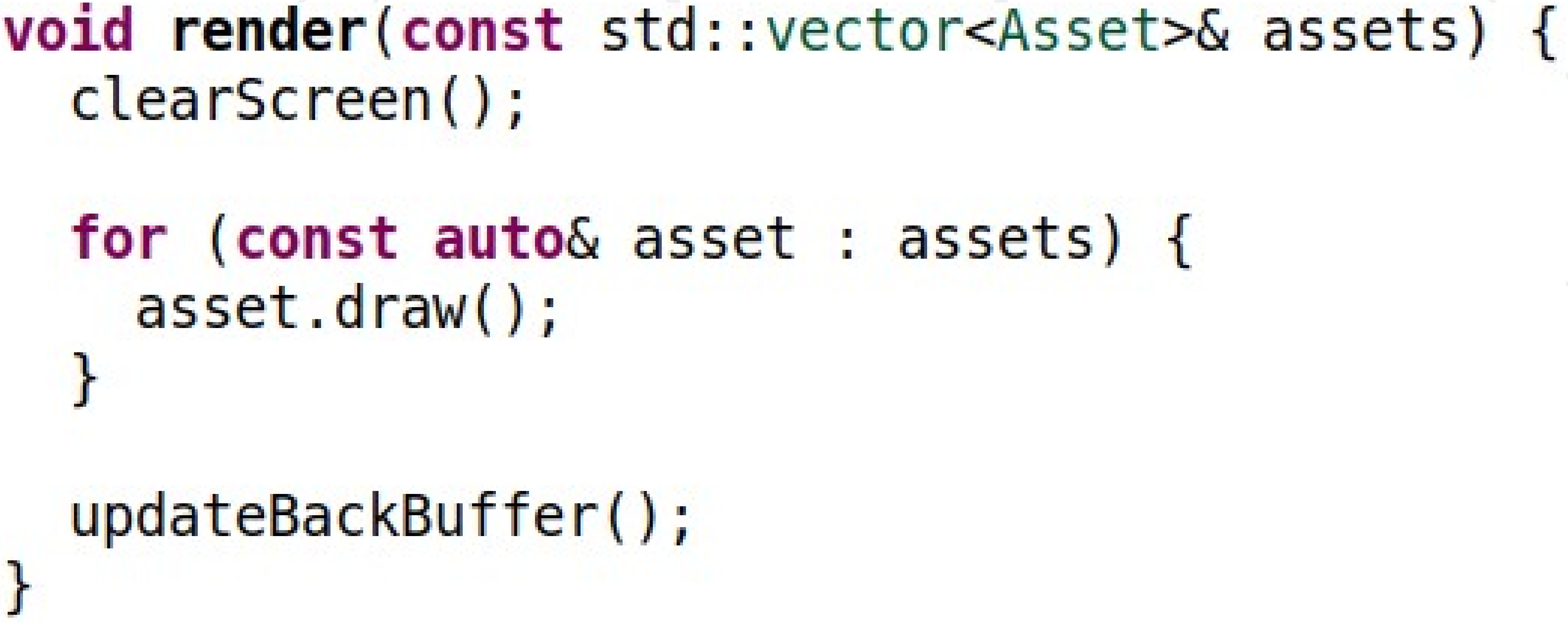
- Screen Rendering is performed **on each frame**
- When an asset is drawn – it should appear on the screen
- On the next frame that asset could be in a different position
- We draw it like normal
- Ouch – **a duplicate**!





Rendering order

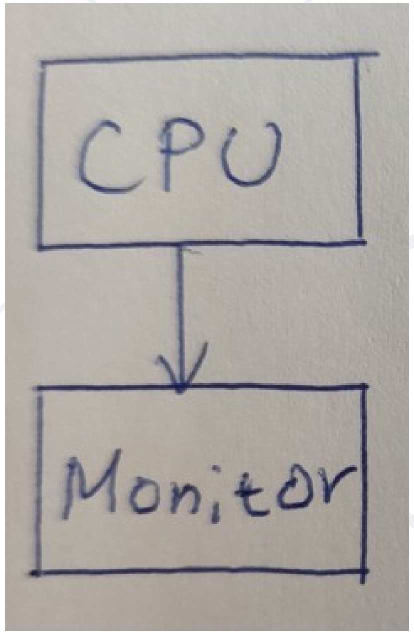
- Modern hardware relies on the concept of clearing, drawing assets and pushing the final frame (called the **back buffer**) to the screen
- This will ensure the screen is properly cleared for each new frame

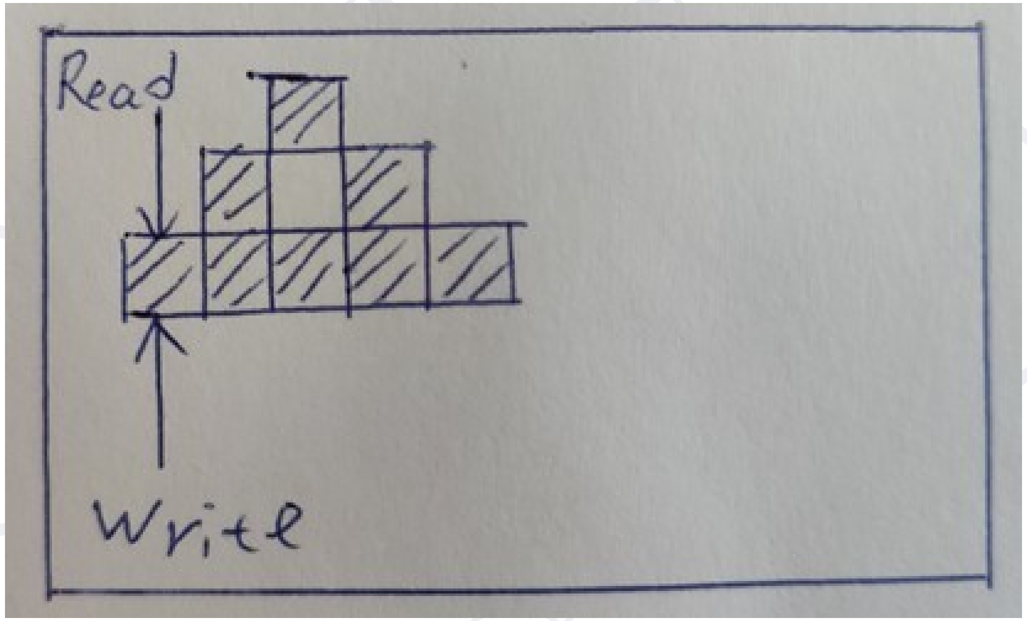


```cpp
void render(const std::vector<Asset>& assets) {
  clearScreen();

  for (const auto& asset : assets) {
    asset.draw();
  }

  updateBackBuffer();
}
```

Single Buffering

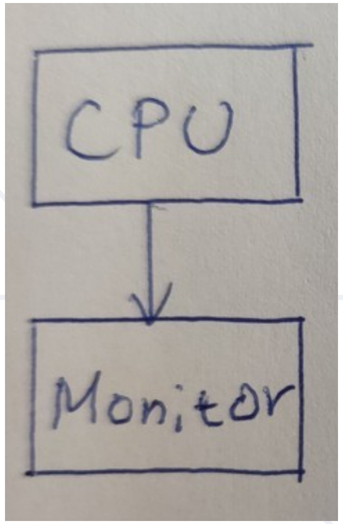
- In the current scenario the CPU is crunching all of the work
- It is responsible for both reading and writing to the back buffer
- “Reading” is sending the final frame from the back buffer to the screen
- “Writing” is populating the back buffer with asset information

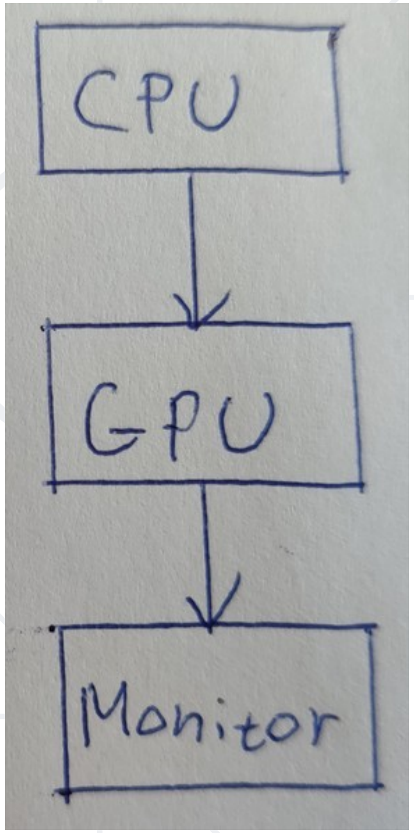




CPU limitations

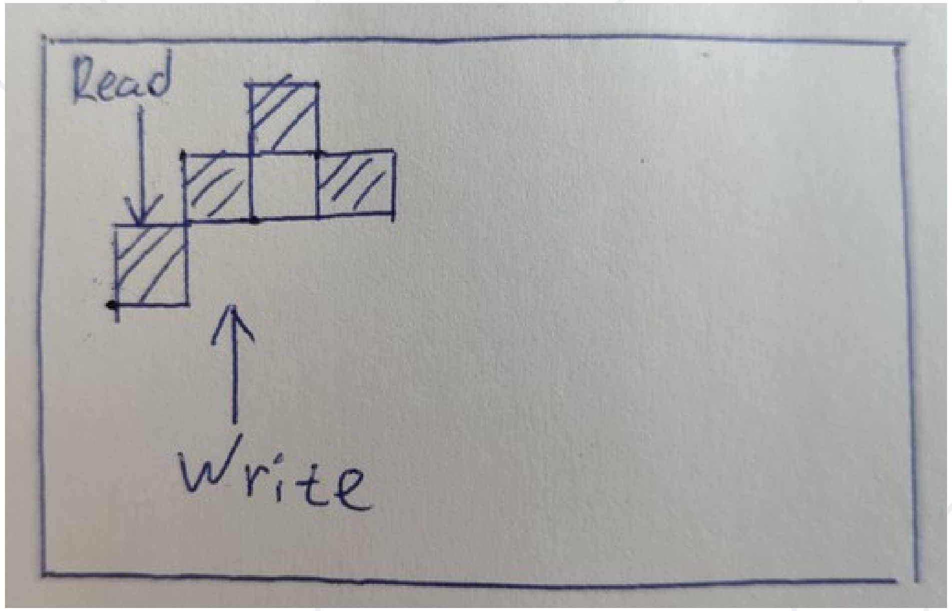
- Graphical work is **quite intensive**.
- The CPU is not able to handle the load and quickly becomes a **bottleneck**
- Here comes the GPU – **blazing fast performance**
- Asset information is **transferred** from the CPU over to the GPU, which handles the actual drawing
- GPU then sends the **final frame** from his back buffer to the **screen**

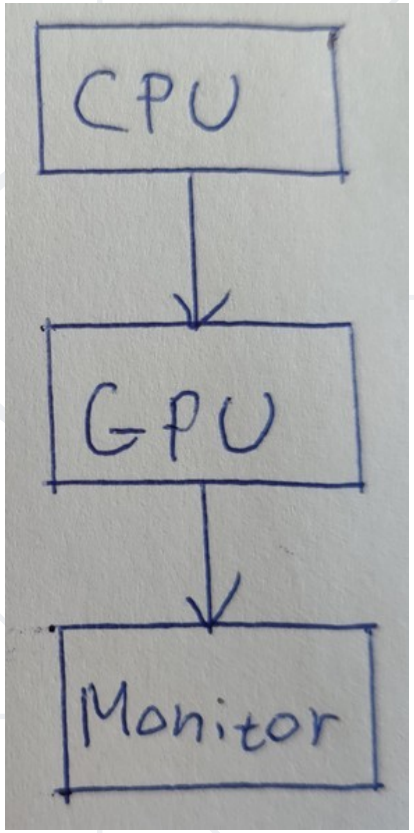




Hardware Acceleration - GPU

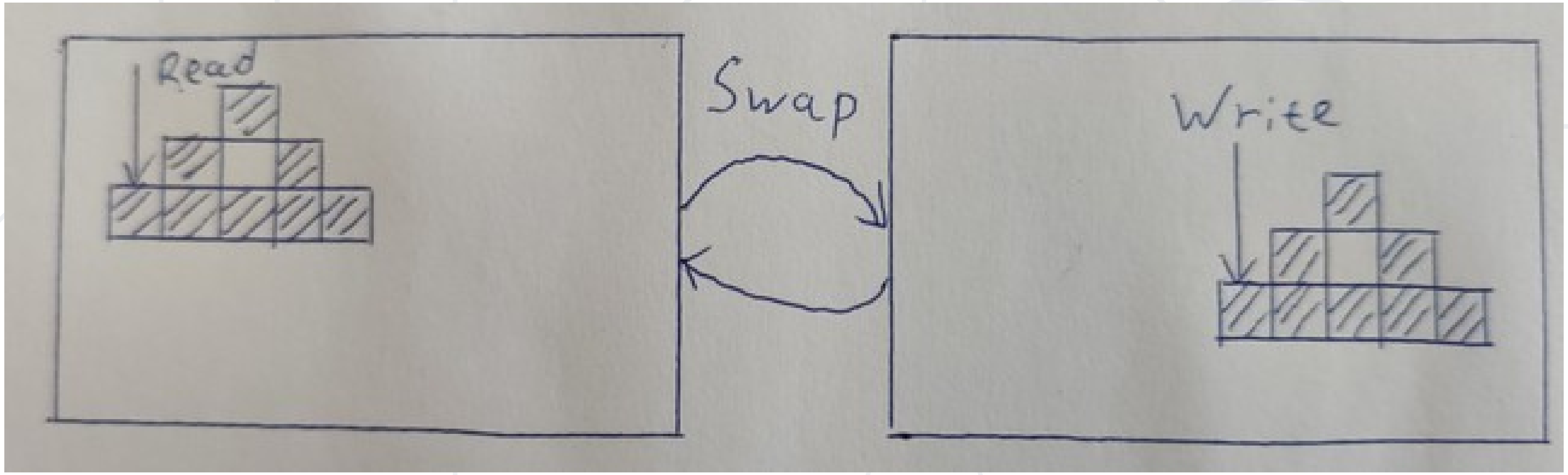
- We now have the required **speed**!
- Sadly, the fight is not over yet.
- Single Buffering suffers from a big problem – reading and writing are **no longer in sync**!
- This phenomenon is referred to as "**tearing**"



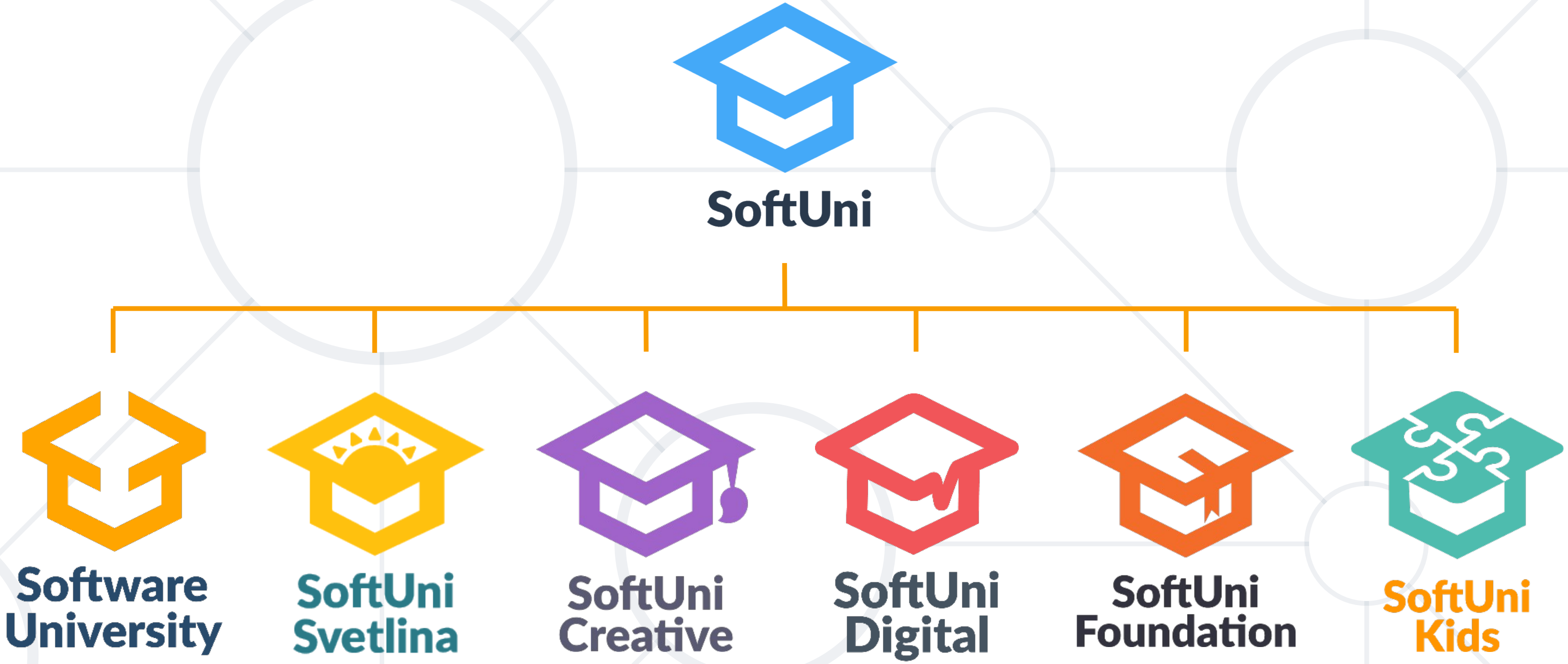


Double Buffering

- This problem is addressed with the addition of **another back buffer**
- GPU is “reading” information from the first buffer, which is sent to the screen
- In the meantime the “write” buffer is being populated by drawn assets
- When the frame is “ready” only a single “**pointer swap**” is performed
- A single pointer swap is again – **blazing fast**



Questions?



SoftUni

Software University

SoftUni Svetlina

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

Diamond Partners

SUPER
HOSTING
.BG

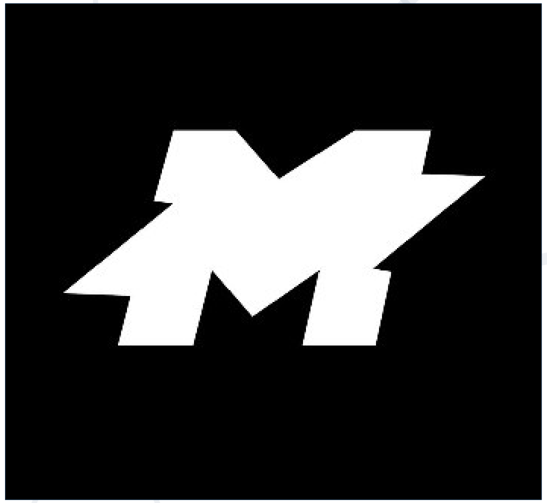
INDEAVR
Serving the high achievers

XS
SOFTWARE

zühlke
empowering ideas

INFRAGISTICS®

Coca-Cola HBC
Bulgaria



DRAFT
KINGS

Postbank
Решения за твоето утре

SOFTWARE
GROUP

Educational Partners

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://about.softuni.bg/
- © Software University – https://softuni.bg

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg