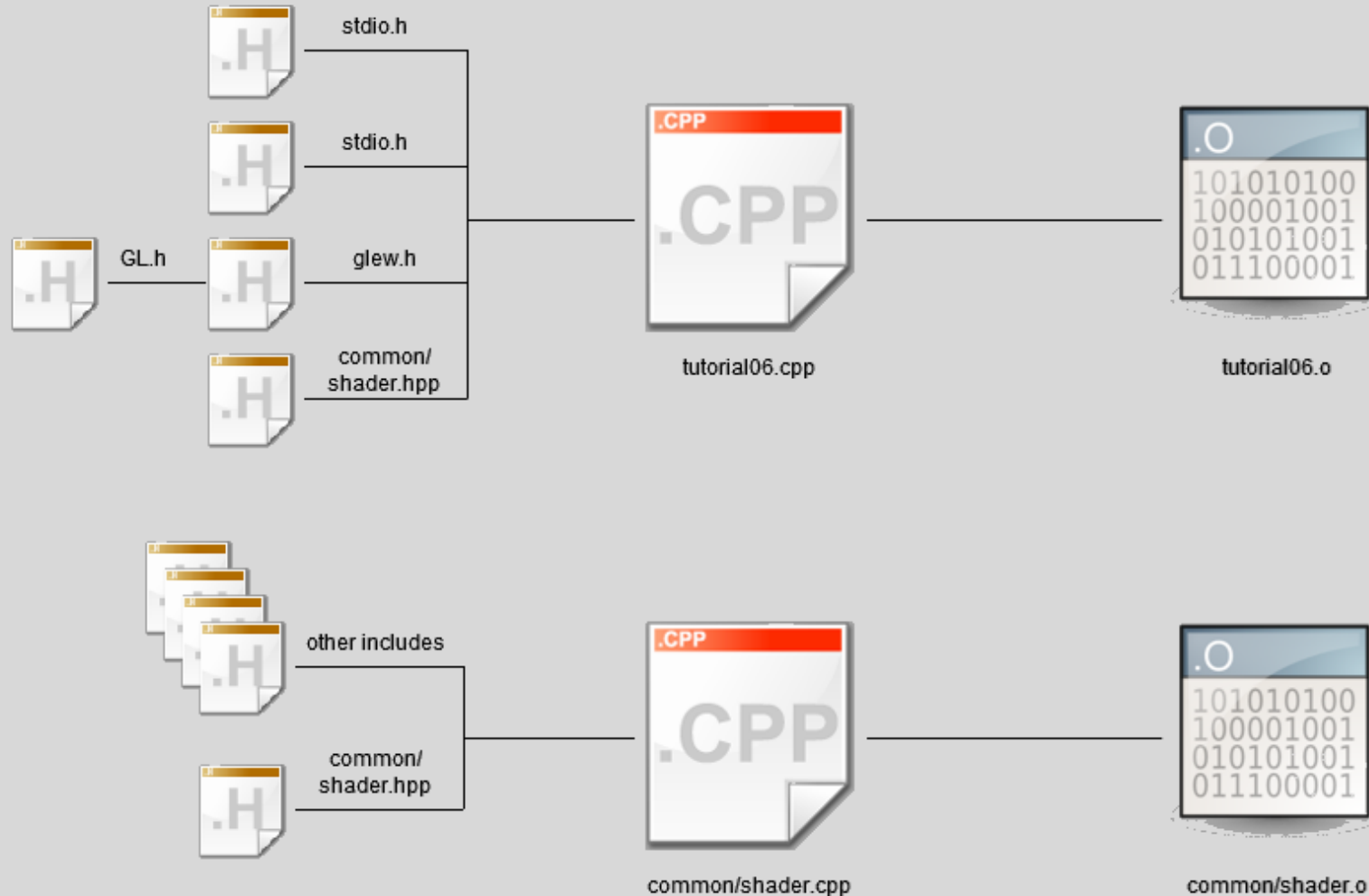




ΓΡΑΦΙΚΑ ΚΑΙ ΕΙΚΟΝΙΚΗ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑ

Εργαστήριο 1

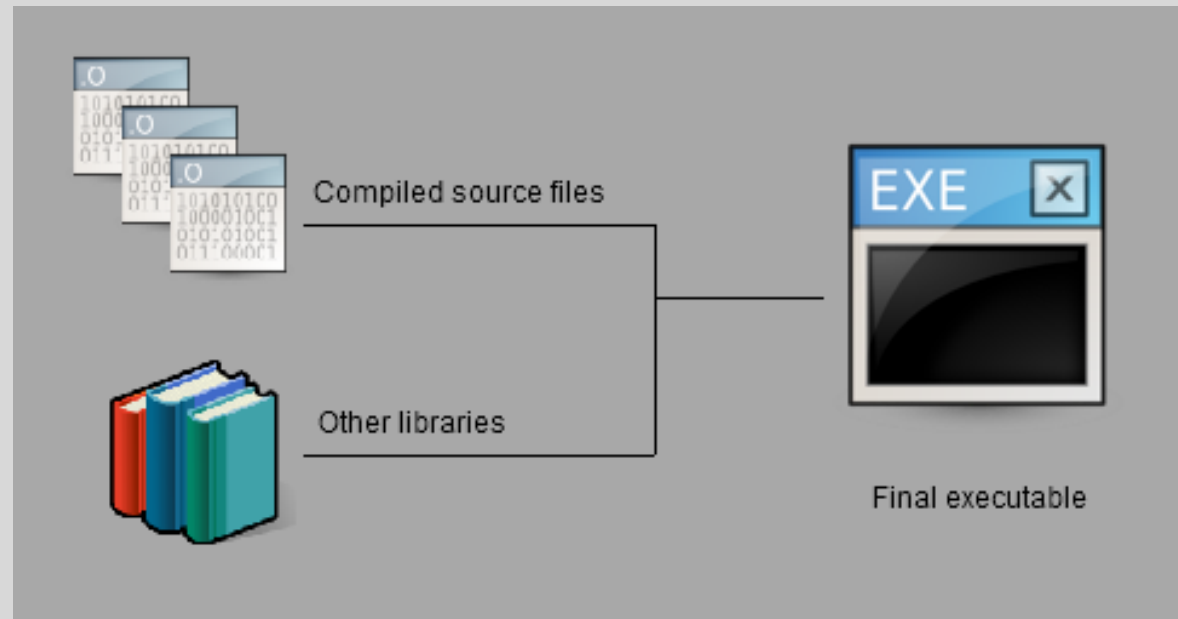
C++ Preprocessing & Translation



C++ Linking

2 library types:

- Static: They contain the assembly code for each function.
- Dynamic: They don't contain any assembly code. The implementation is found inside a .dll file which is read at runtime.

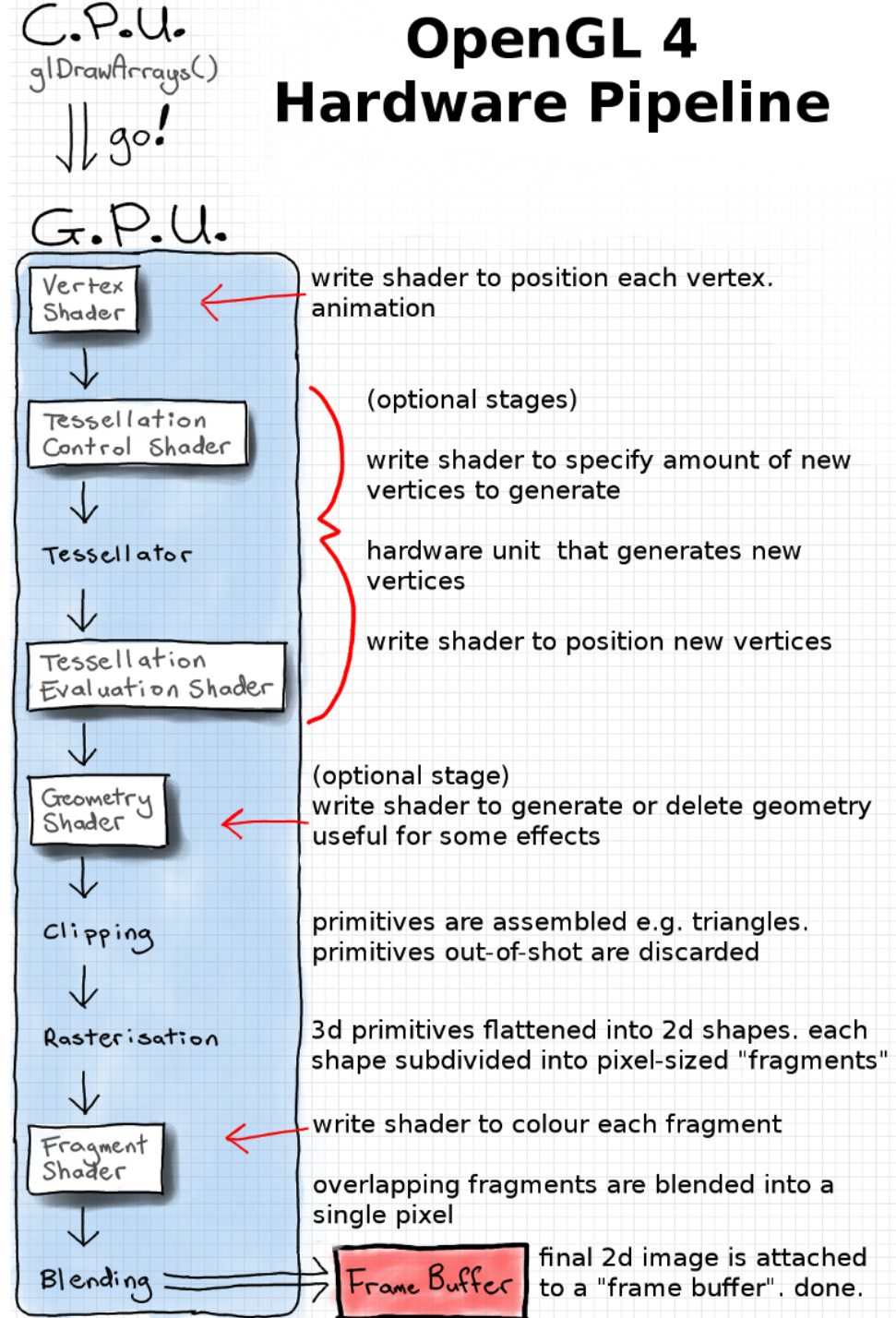


OpenGL

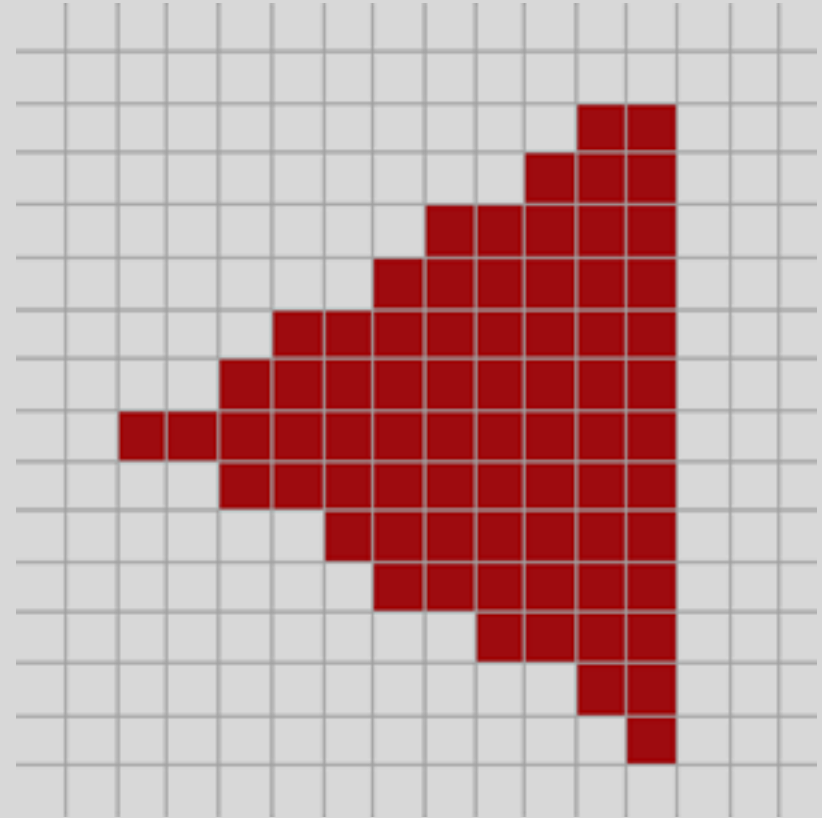
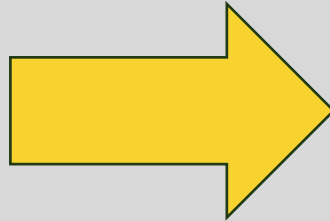
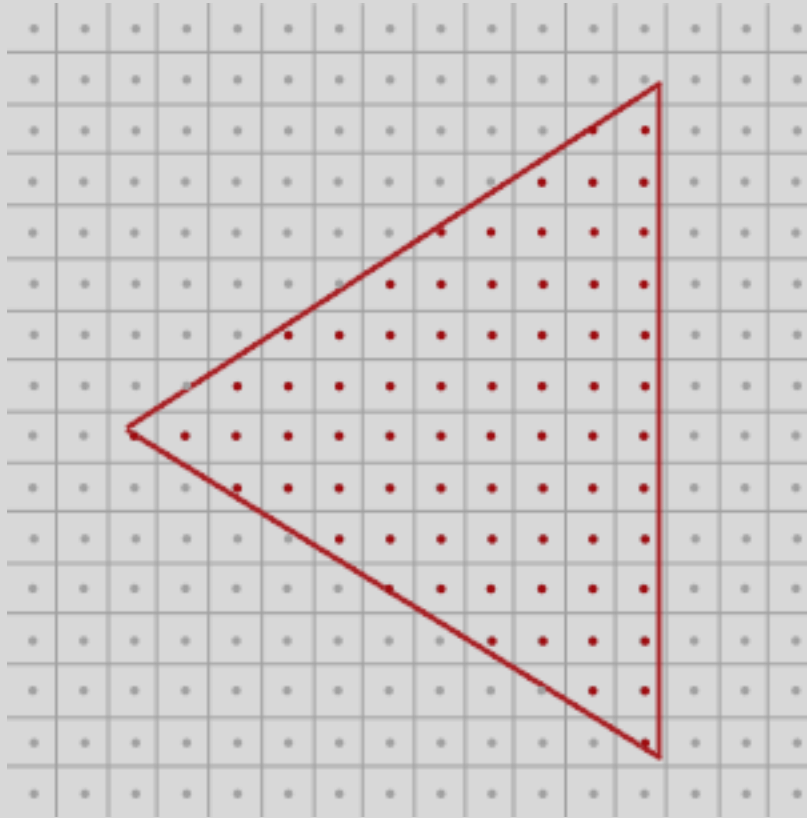
- Βιβλιοθήκη για τη δημιουργία γραφικών.
- Διάδραση μέσω του API σε όποια γλώσσα προγραμματισμού θέλουμε.
Συνηθέστερα C/C++ ή Java.
Διαθέσιμα bindings για πολλές γλώσσες: Python, C#, Lua, Objective-C, ...

OpenGL – pipeline

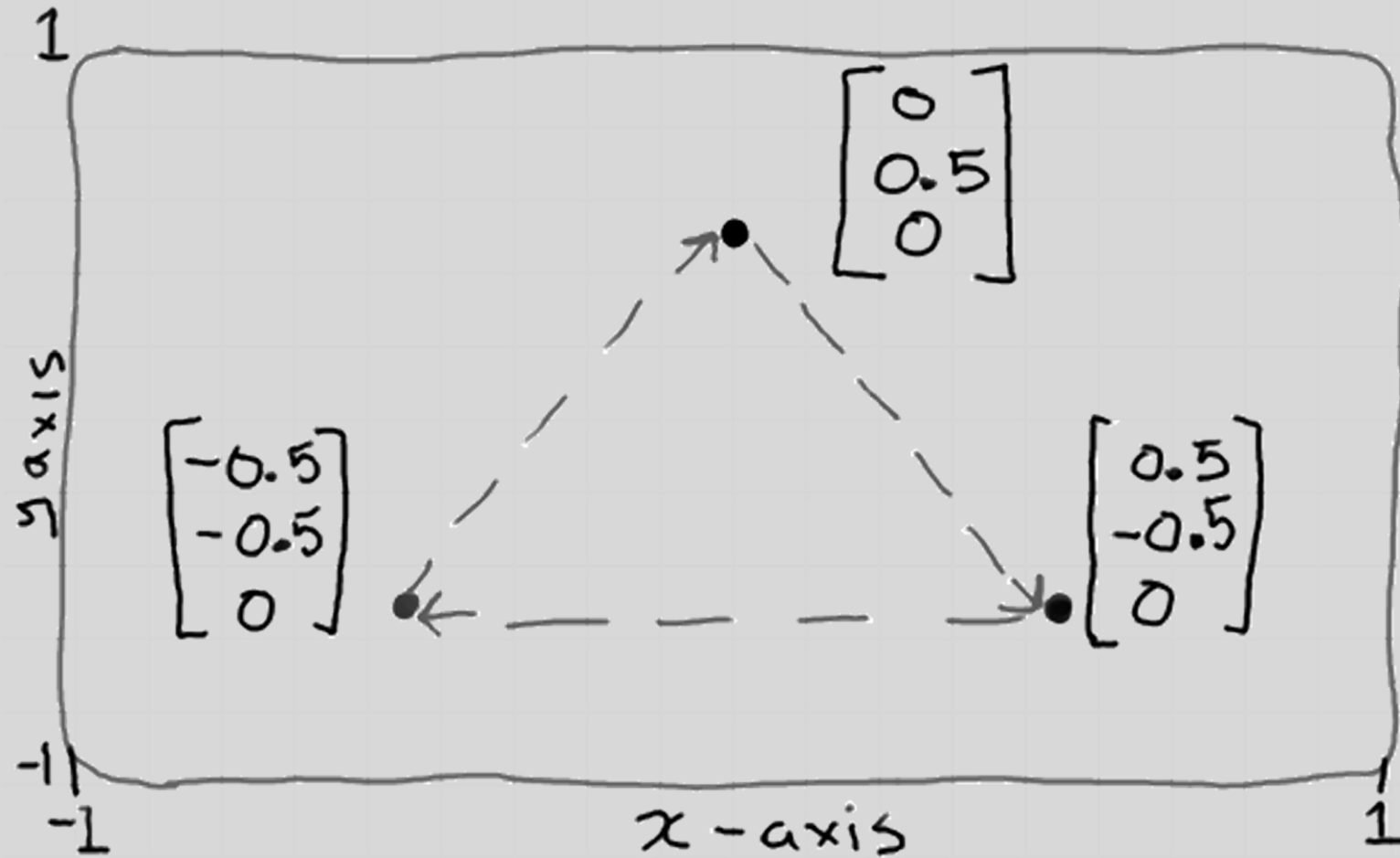
- Οι shaders είναι γραμμένοι σε μία γλώσσα που ονομάζεται GLSL (GL Shading Language) και μπορούμε να τους **τροποποιήσουμε**.



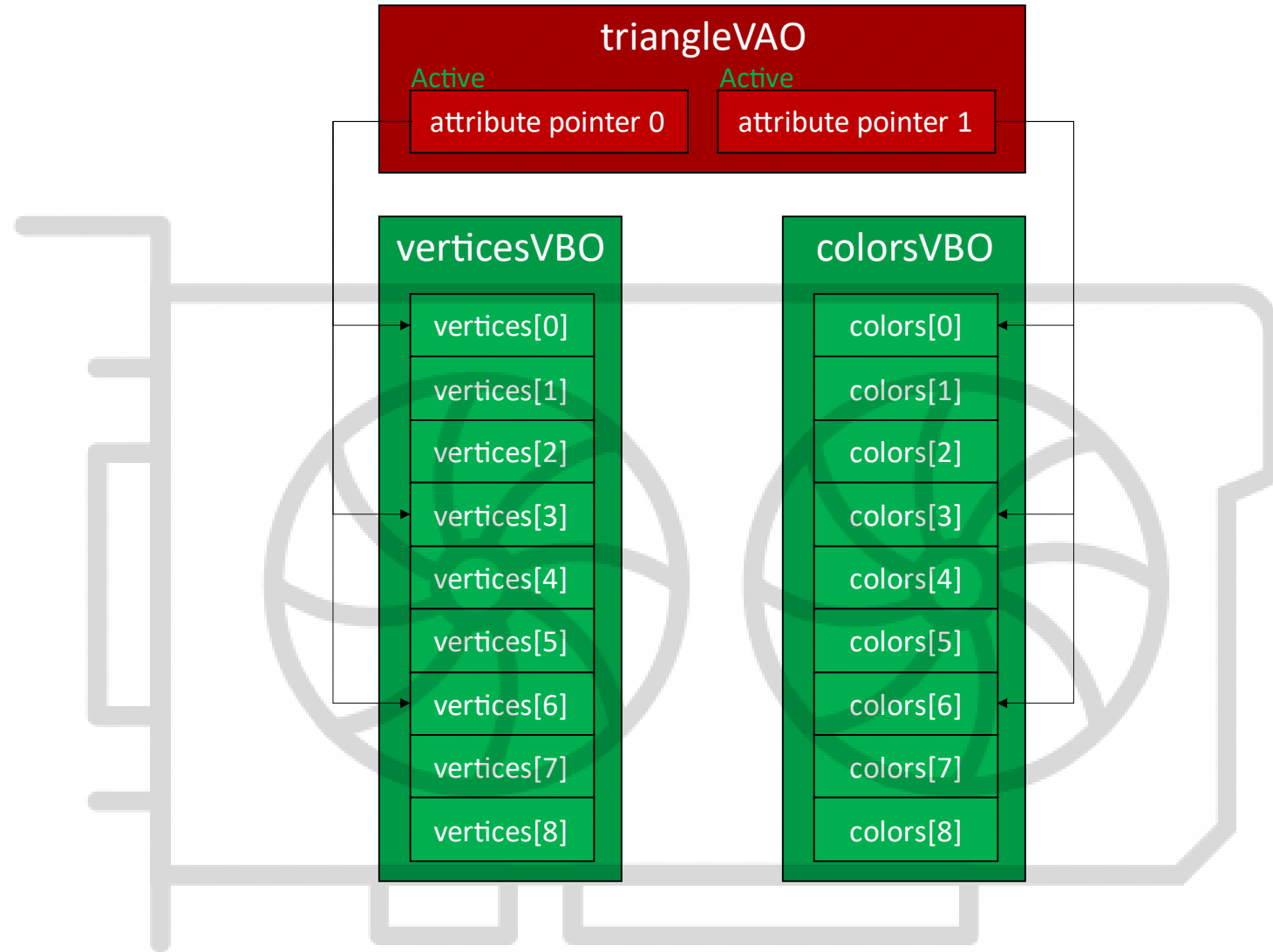
Rasterization (no anti-aliasing)



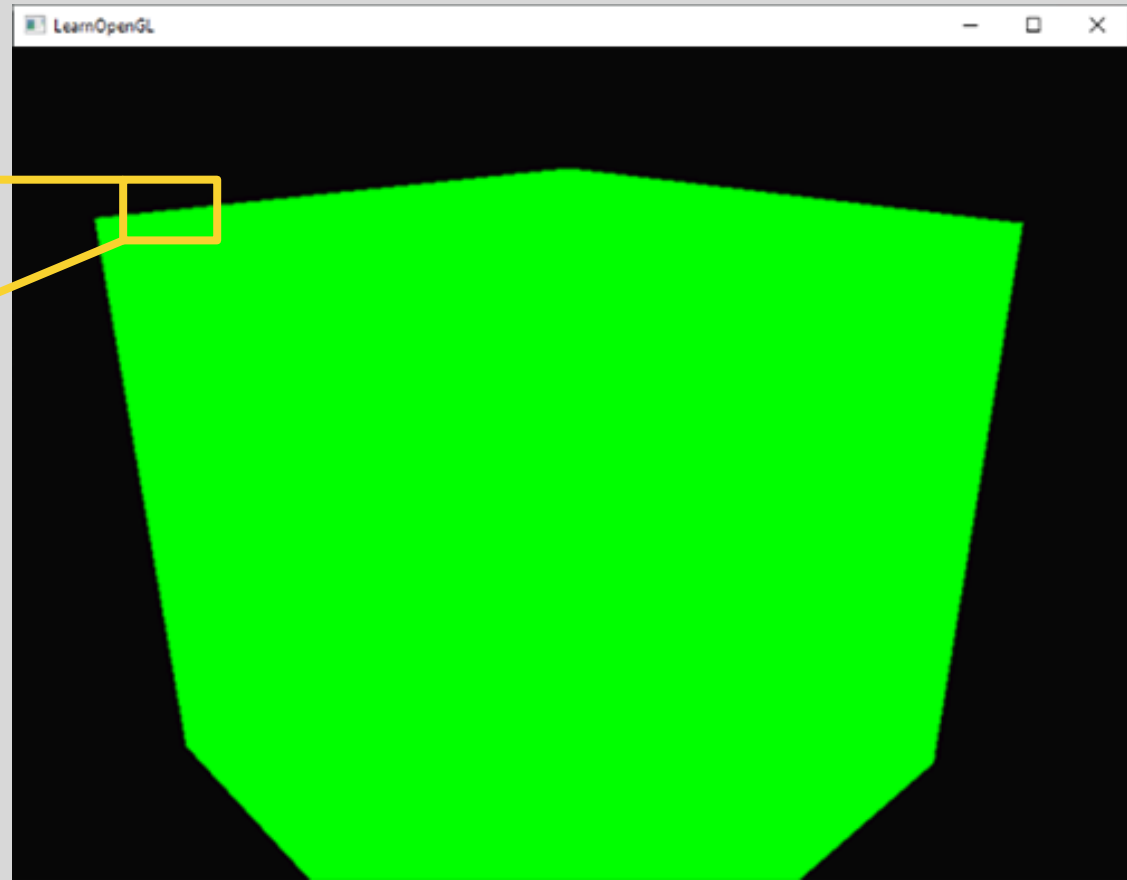
Display a triangle



VAOs & VBOs

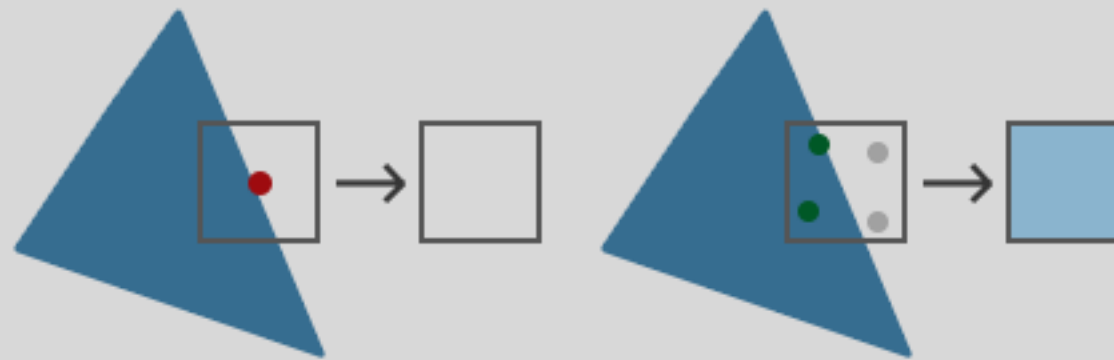


Aliasing



(Spatial) Anti-Aliasing

- Πολλοί διαφορετικοί αλγόριθμοι
- Tradeoff μεταξύ ποιότητας καρέ και χρόνου καρέ
- Οι περισσότεροι βασίζονται σε **multisampling**



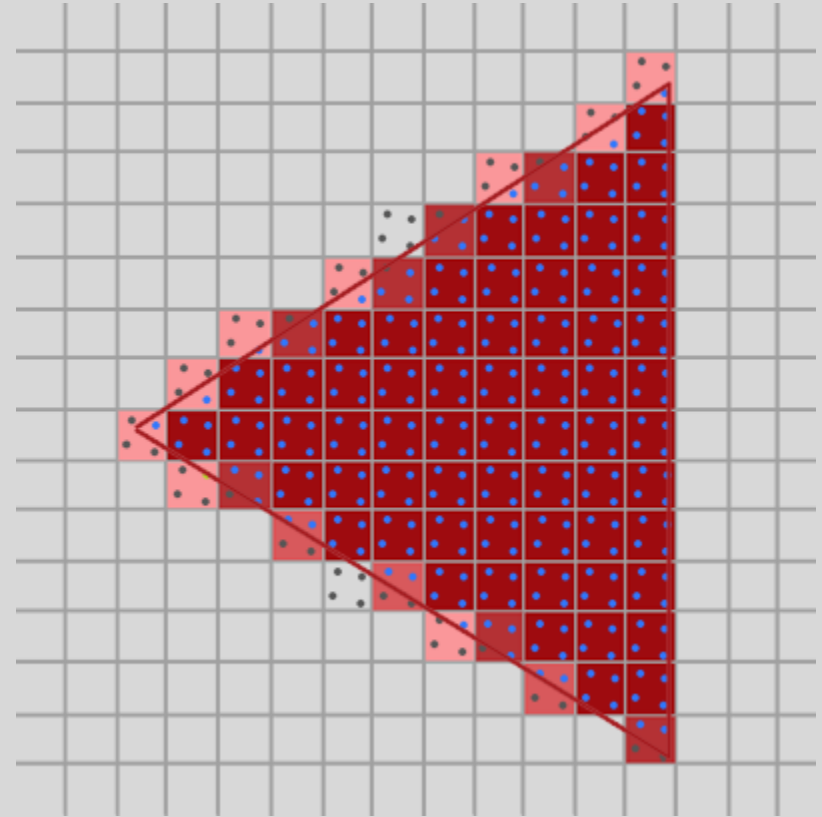
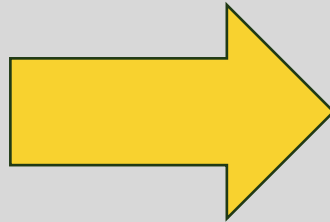
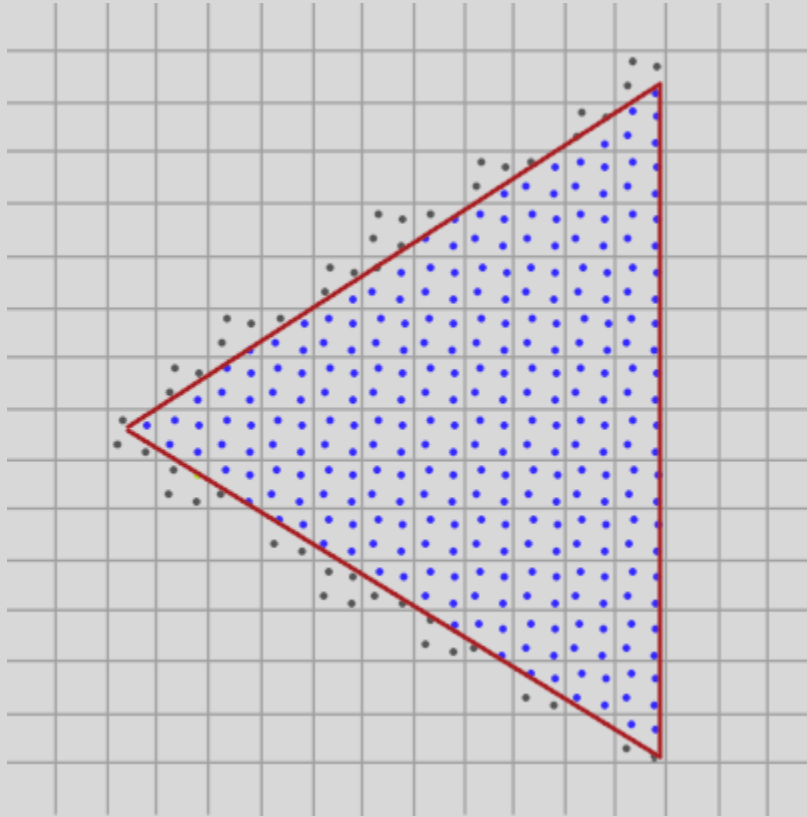
Multisample Anti-Aliasing (MSAA)

- Ενεργοποίηση 4xMSAA στην OpenGL (μέσω της GLFW):

```
glfwWindowHint(GLFW_SAMPLES, 4);
```

- Οι υπολογισμοί δε γίνονται 4 φορές για κάθε pixel!
Γίνονται με interpolation στο κέντρο του pixel, η τελική τιμή, όμως, αποφασίζεται ανάλογα με το πόσα samples βρίσκονται εντός του primitive.

Rasterization (4xMSAA)



Σημείωμα Χρήσης Έργων Τρίτων

Το έργο αυτό κάνει χρήση των ακόλουθων έργων τρίτων:

- <https://antongerdelan.net/opengl/shaders.html>
- <https://antongerdelan.net/opengl/hellotriangle.html>
- <https://learnopengl.com/Advanced-OpenGL/Anti-Aliasing>