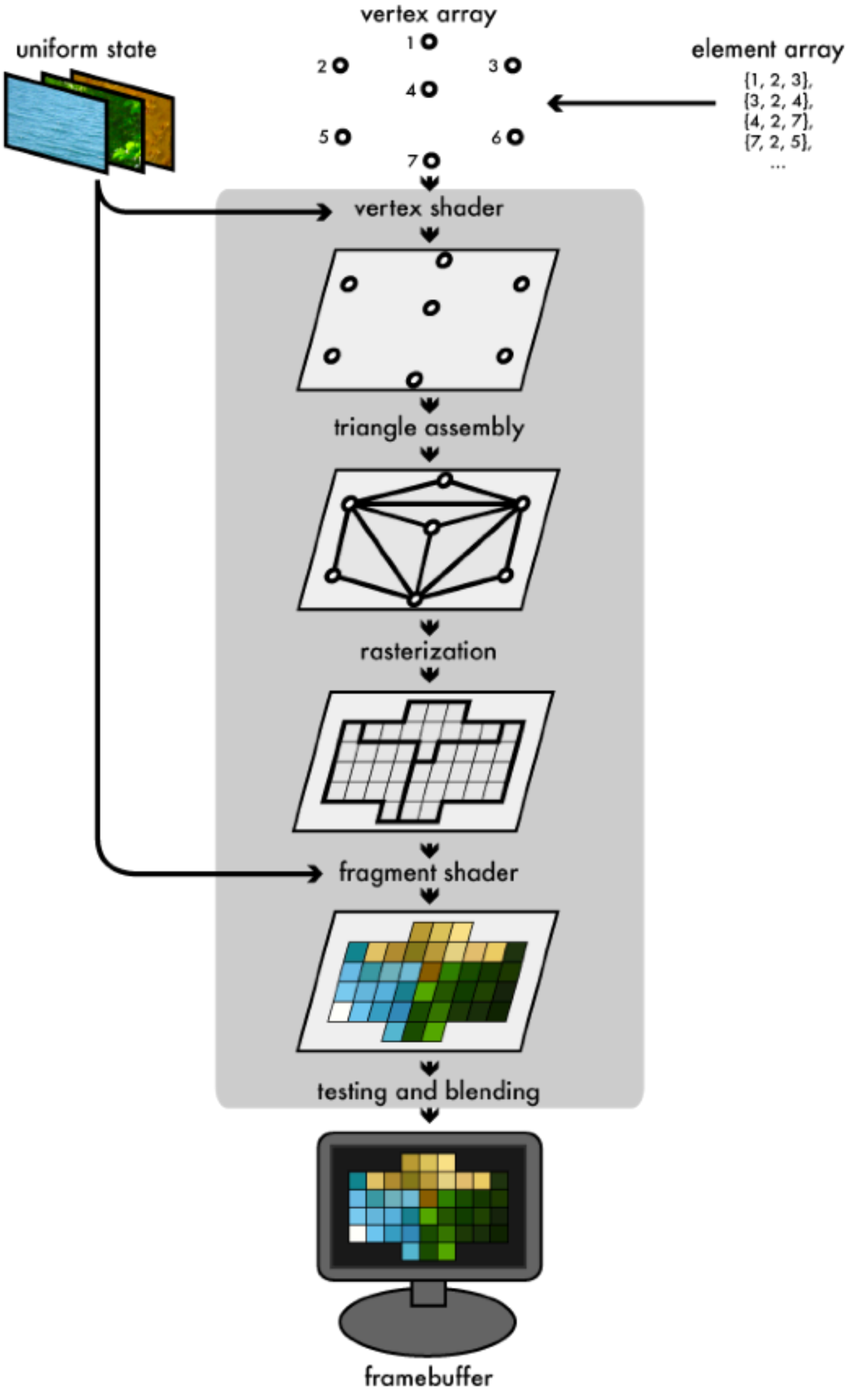
Dynamic Patterns

Realization


vertex array
1
2
3
4
5
6
7
uniform state
element array
{1, 2, 3},
{3, 2, 4},
{4, 2, 7},
{7, 2, 5},
...
vertex shader
triangle assembly
rasterization
fragment shader
testing and blending
framebuffer

WebGL Rendering Pipeline

Source: https://duriansoftware.com/joe/an-intro-to-modern-opengl.-chapter-1:-the-graphics-pipeline

- GPU is designed to efficiently process a big amount simple tasks in parallel

- WebGL allows user to program vertex shaders (geometry) and fragment shaders (pixel coloring) with GLSL

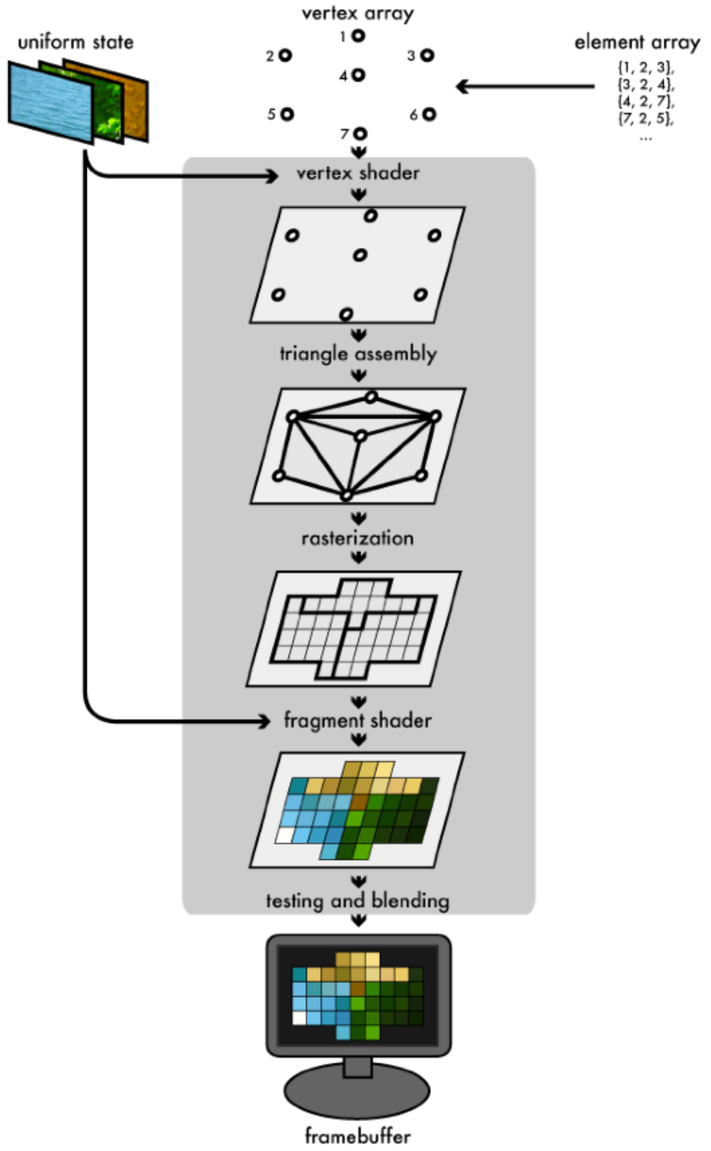
- Programming shaders enables us to fully leverage GPU's power to achieve amazing results

Realization

Dynamic Patterns

- GPU is designed to efficiently process a big amount simple tasks in parallel
- WebGL allows user to program vertex shaders (geometry) and fragment shaders (pixel coloring) with GLSL
- Programming shaders enables us to fully leverage GPU's power to achieve amazing results

WebGL Rendering Pipeline



Source: https://duriansoftware.com/joe/an-intro-to-modern-opengl.-chapter-1:-the-graphics-pipeline

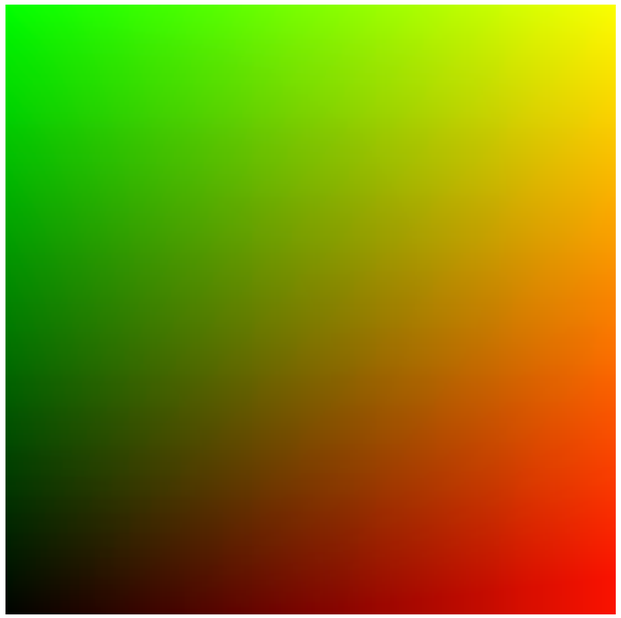
Realization

Dynamic Patterns

```
#ifdef GL_ES
precision mediump float;
#endif

uniform vec2 u_resolution;
uniform vec2 u_mouse;
uniform float u_time;

void main() {
    vec2 st = gl_FragCoord.xy/u_resolution;
    gl_FragColor = vec4(st.x,st.y,0.0,1.0);
}
```



Fragment Shader Example. Source: https://thebookofshaders.com/