CrowdSim: Milestone 2

Hannah Bollar & Eric Chiu

Biocrowds

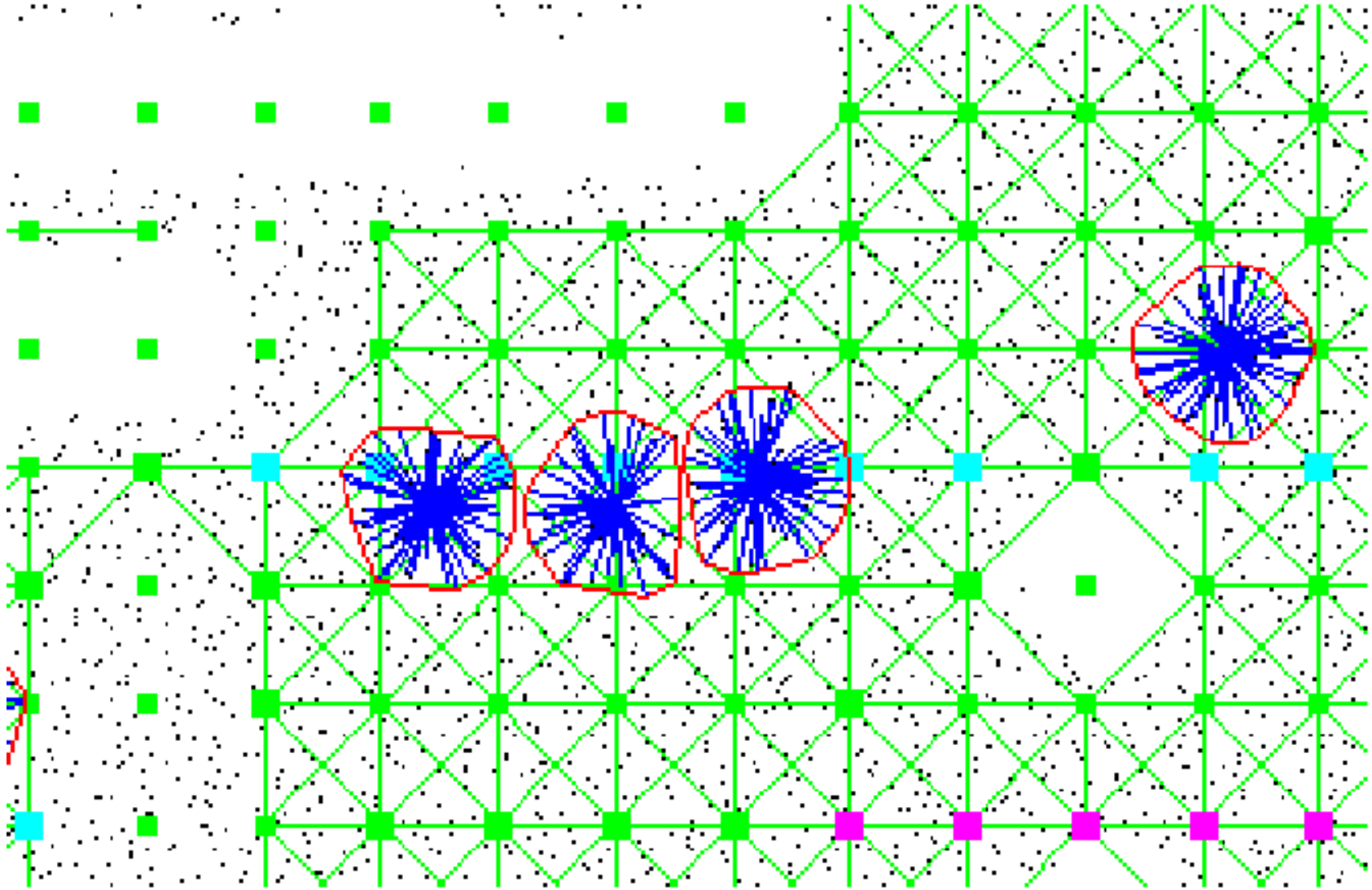
For each agent

find closest available markers (can claim if priority)

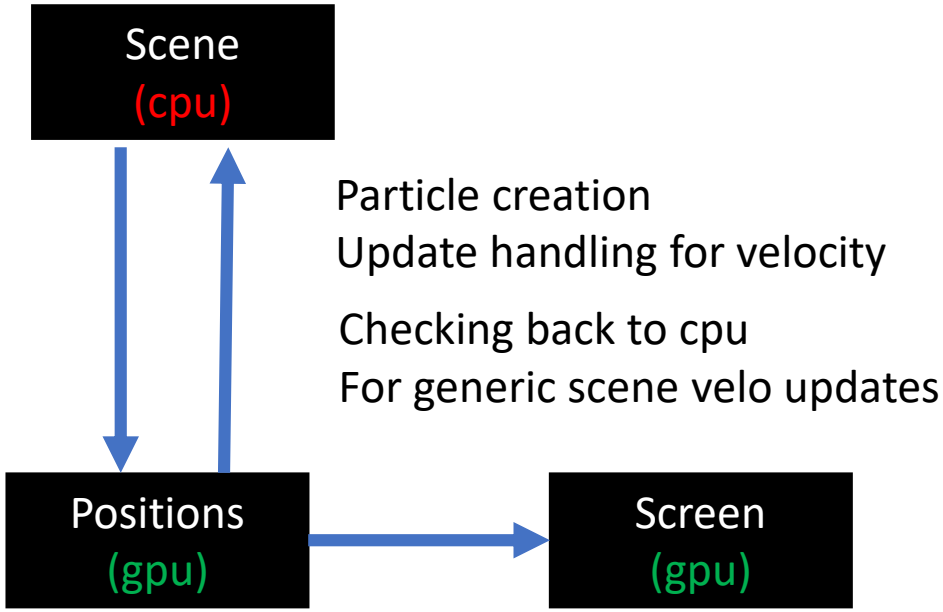
cone check (useful in depth buffer of fshader but we're faking the depth buffer here)

update velocity based on ave of marker influence towards target

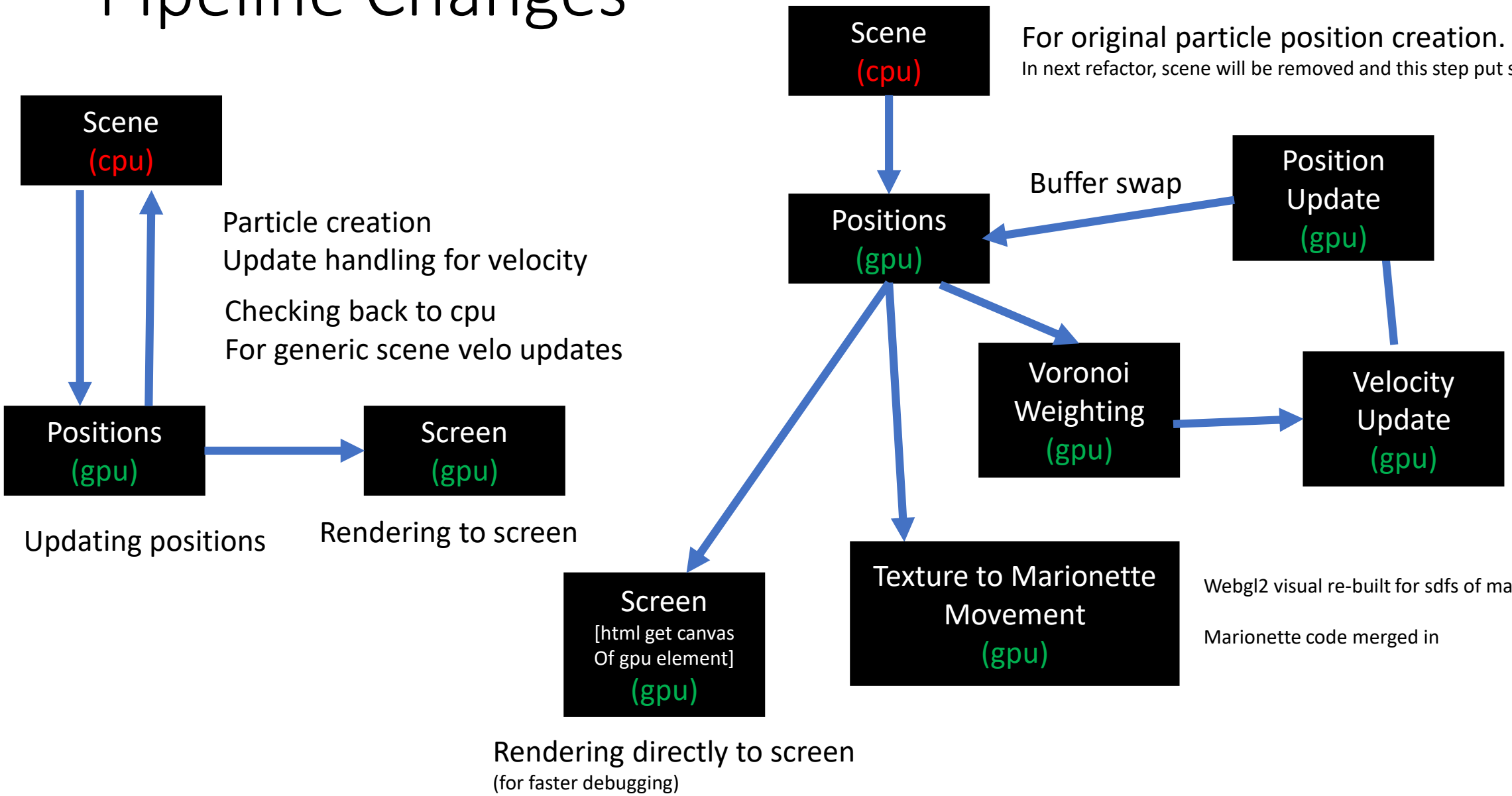
update position based on velocity change



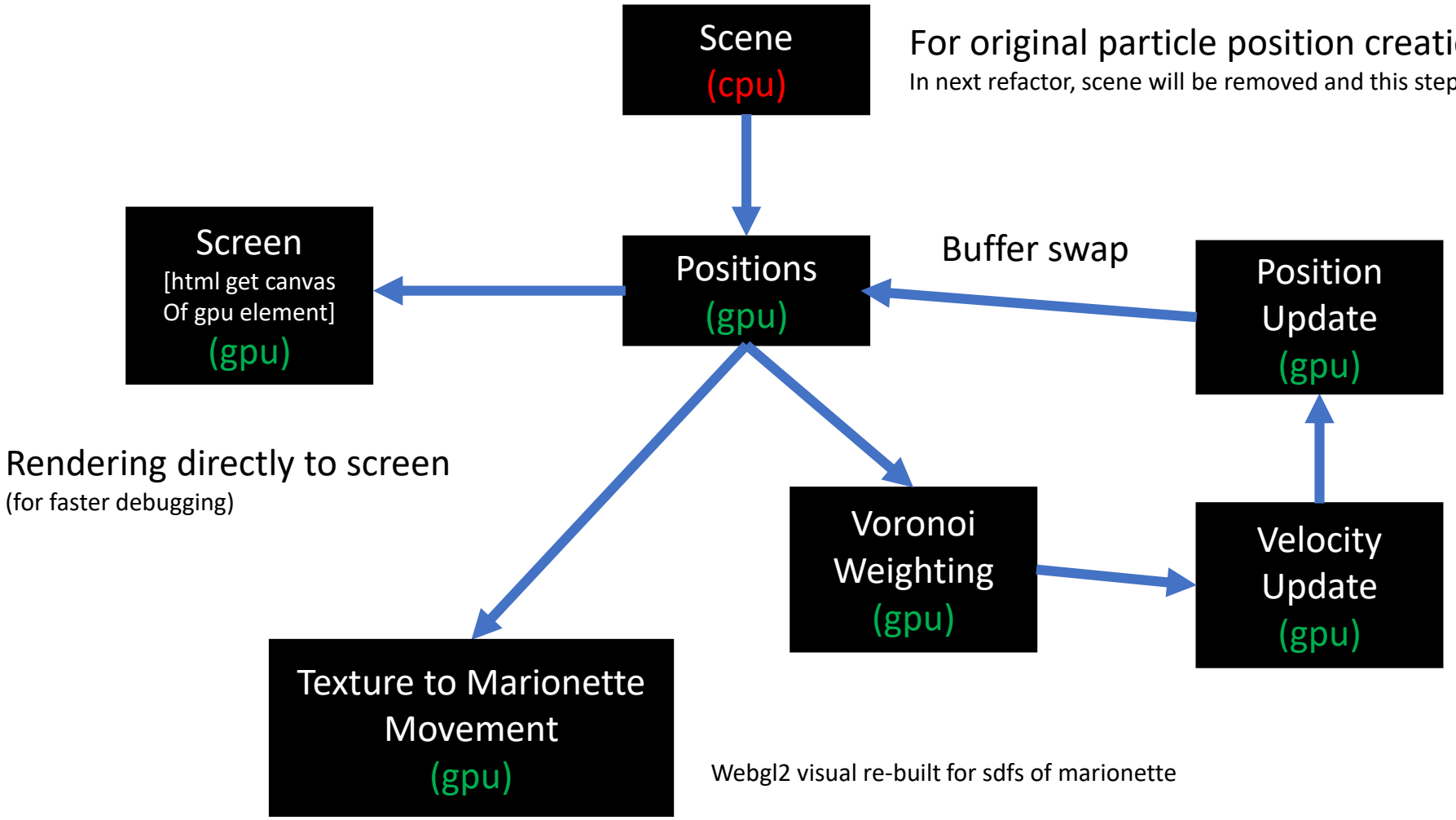
Pipeline Changes



Pipeline Changes



Pipeline Changes



Webgl2 visual re-built for sdfs of marionette

Marionette code merged in

setGraphical vs outputToTexture

- setGraphical
  - Using setGraphical before
  - currently no full support for canvas as input (gpu -> cpu -> gpu -> ...)
  - Useful for debugging since an actual obj?

setGraphical vs outputToTexture

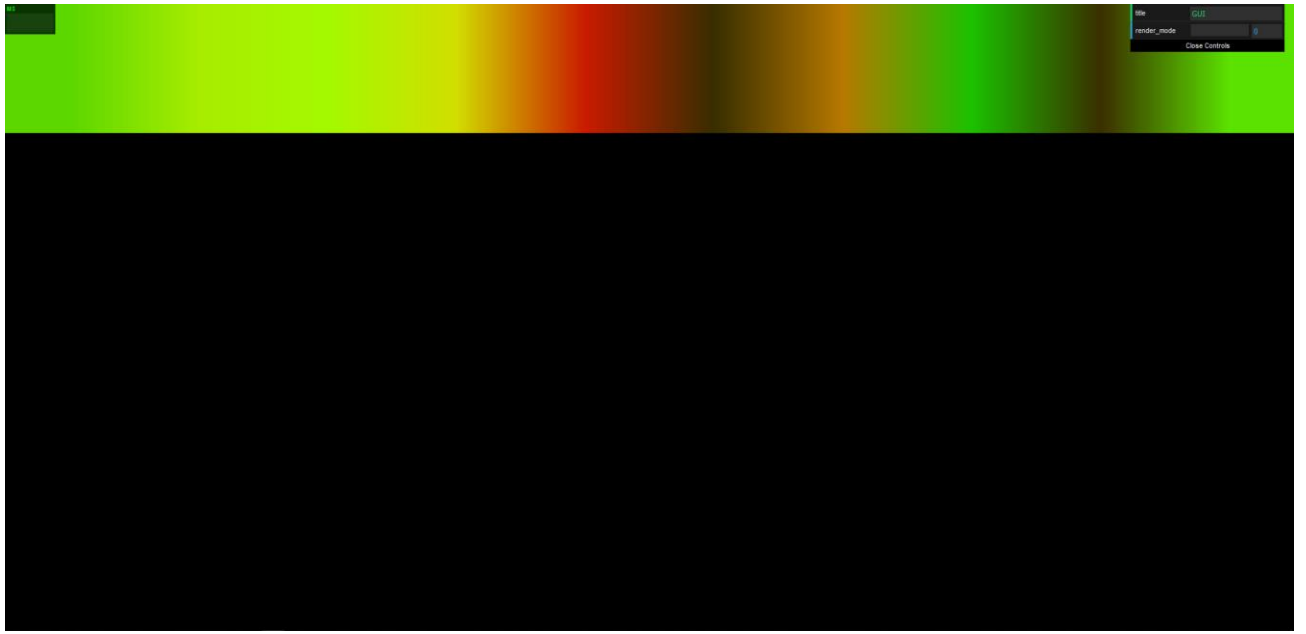
- setGraphical
  - Using setGraphical before
  - currently no full support for canvas as input (gpu -> cpu -> gpu -> …)
  - Useful for debugging since an actual obj?

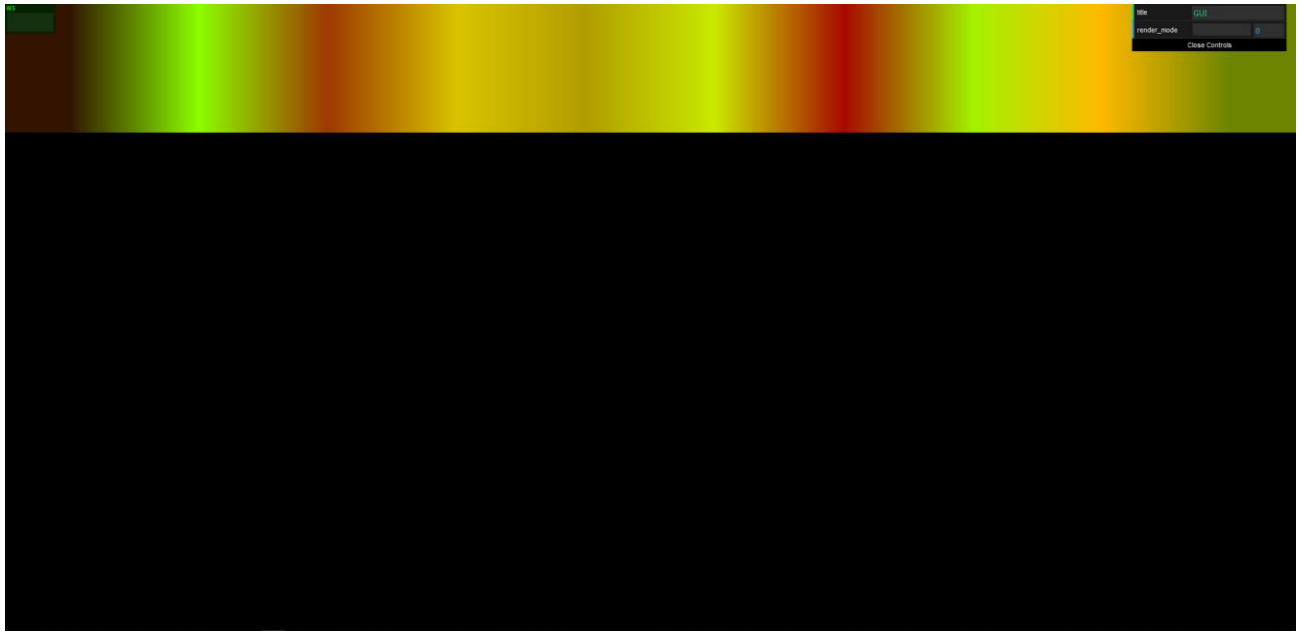
- outputToTexture
  - Maintains on gpu bc it’s an actual texture in gl context
  - Can act as gpujs input
  - Read as an array of arrays (no javascript object element)
  - Remains fully on gpu even with function transfers
  - Vis directly to screen if nothing set to render (since it’s the context’s texture) so actually better for debugging

outputToTexture

If no current vis on canvas
Renders directly to screen

Regardless of screen vs texture dim



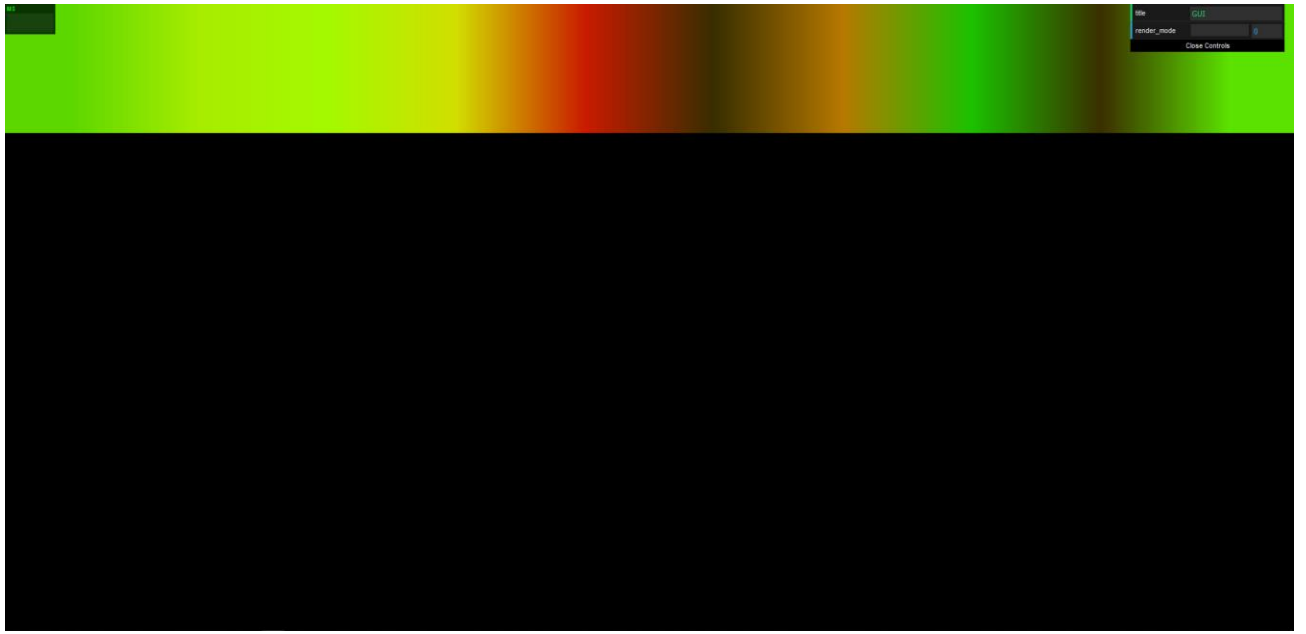


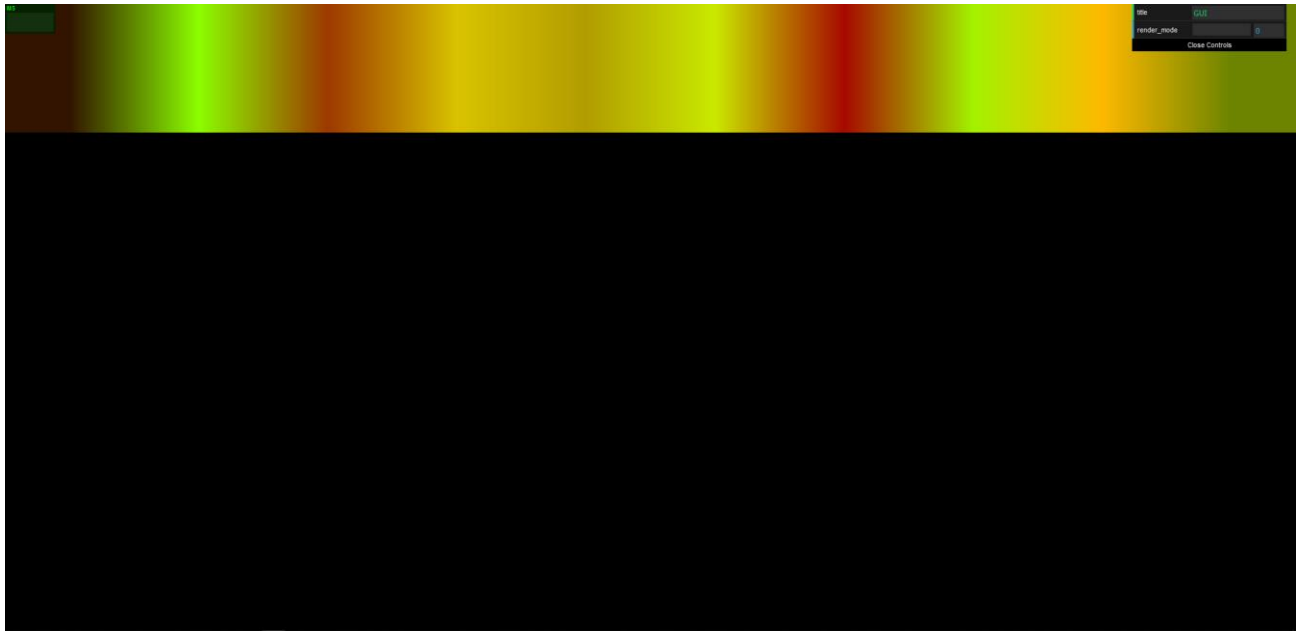
outputToTexture

If no current vis on canvas
Renders directly to screen

Regardless of screen vs texture dim

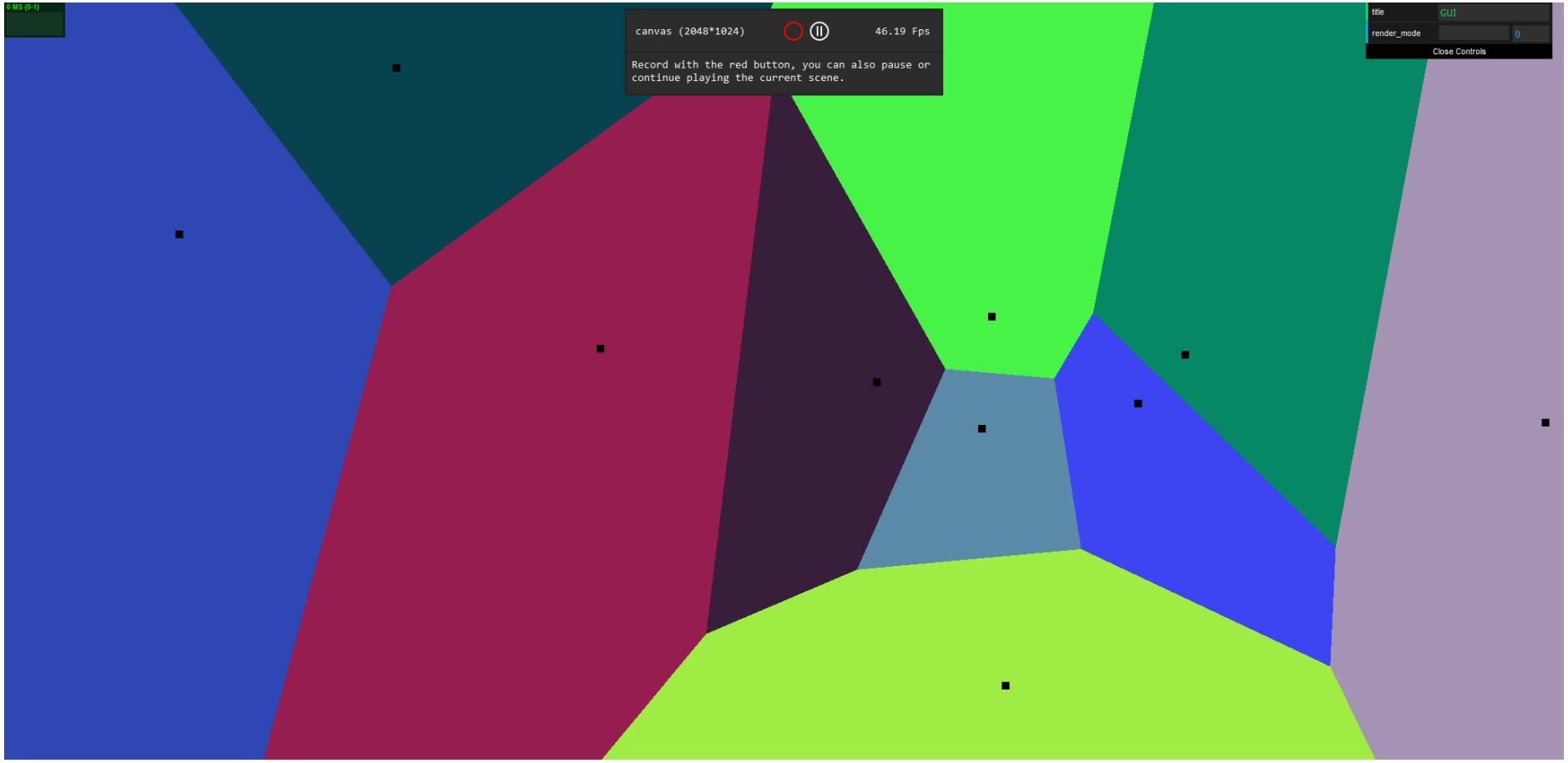
These are actual position locations in a linear array. Coloring right now isn't per pixel bc im doing my indexing wrong for how to store the info into the texture (they do array of arrays in z y x order for indexing instead of usual – looking into)





setGraphical

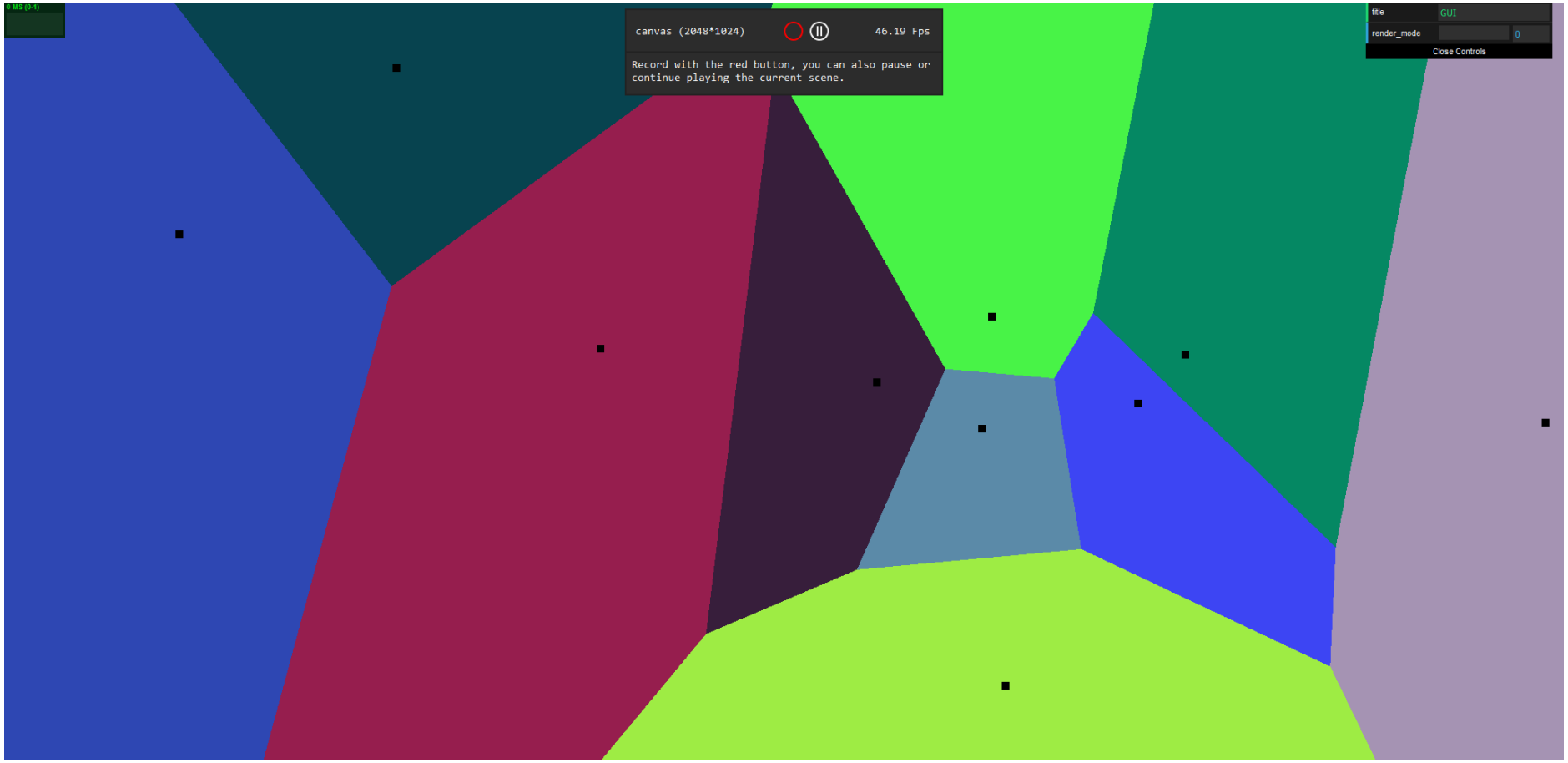
Calling toCanvas()
On the kernel later



setGraphical

Calling toCanvas()
On the kernel later

Should move with the new updated pipeline bc of indexing issue still resolving



Marionette

- Baking movements for sdf updates
- Merged this code from prev milestone into shader section of webgl2 of this project for visualization
- Added in walker class to handle baked movement.

Current Progress (this milestone)

- Large refactor of pipeline
  - Now all on gpu
- Velocity updates properly (not just generic stepping)
- Using cone checking instead of pure pixel dist
  - Fakes the “depth buffer”
- Marionette merged in

Current Progress (this milestone to next)

- Large refactor of pipeline
  - Now all on
- Velocity updates properly (not just generic stepping)
- Using cone checking instead of pure pixel dist
  - Fakes the “depth buffer”
  - TODO (for speed optimization)
    - Currently doing for each pixel, check
    - Need to update to for each point, check pixels in distance [indexing and kernel change]
  - TODO (for movement)
    - indexing issue bc of swap from regular array to outputToTexture – so no movement in visual
- Marionette merged in
  - TODO (for movement)
    - Get baking to work properly in shader
    - Map marionettes atop positions designated by gpu pipeline