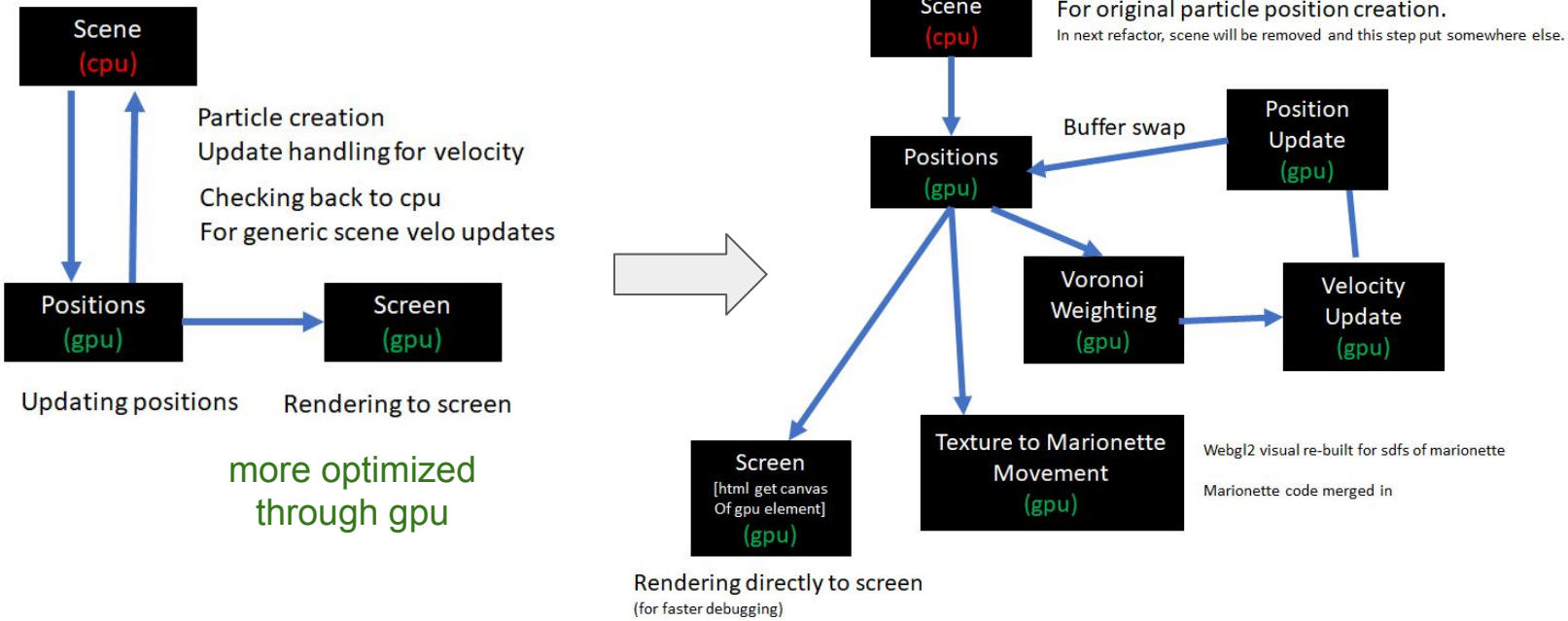
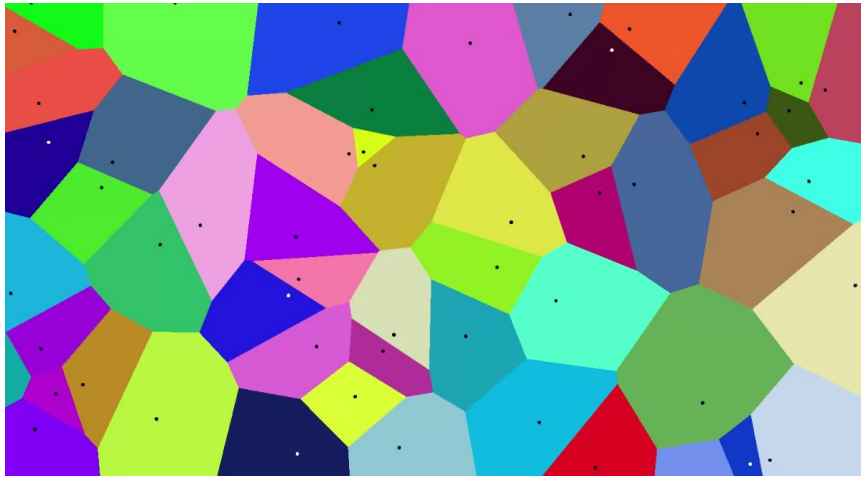
CrowdSim: Milestone 3

Hannah & Eric

gpujs - pipeline changes from last time

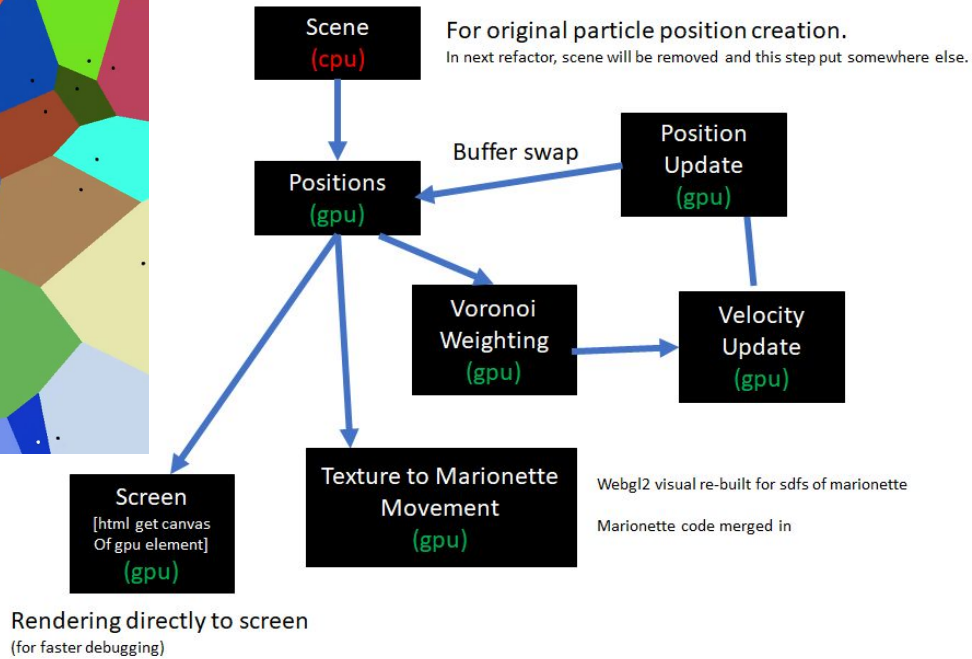


gpujs - pipeline changes from last time

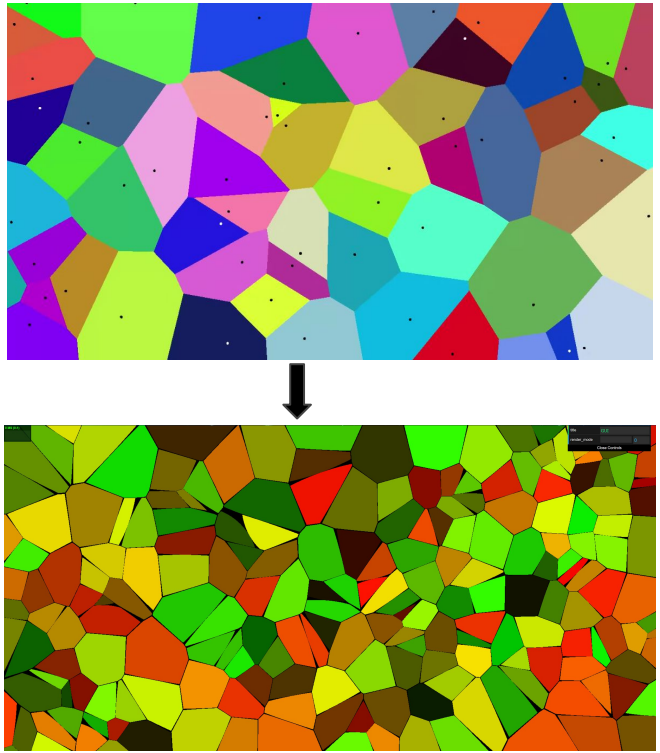


more optimized
through gpu

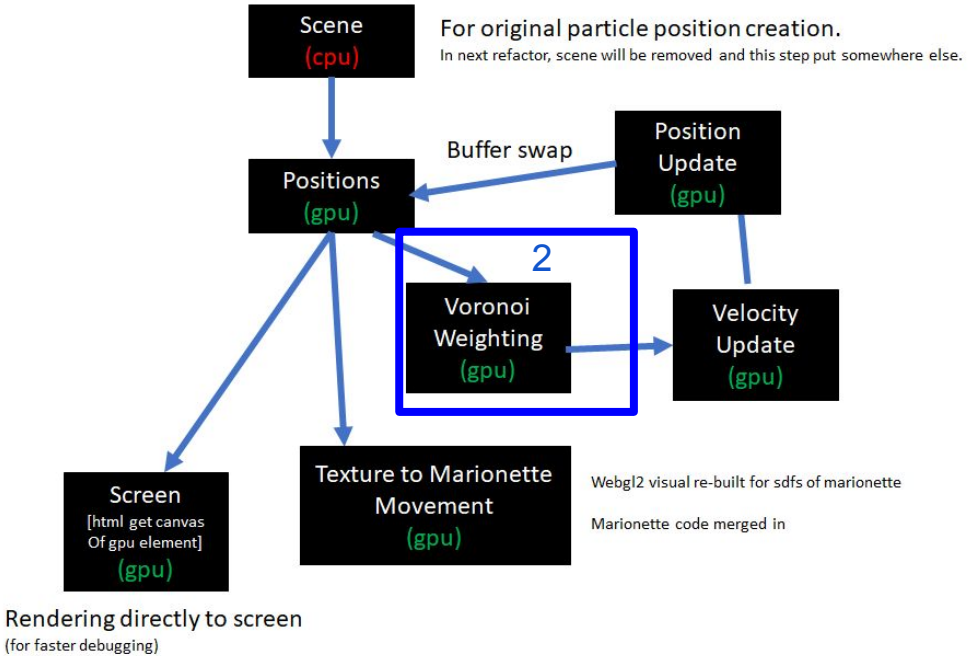
BUT STILL HAD
INDEXING
ISSUE
so no movement



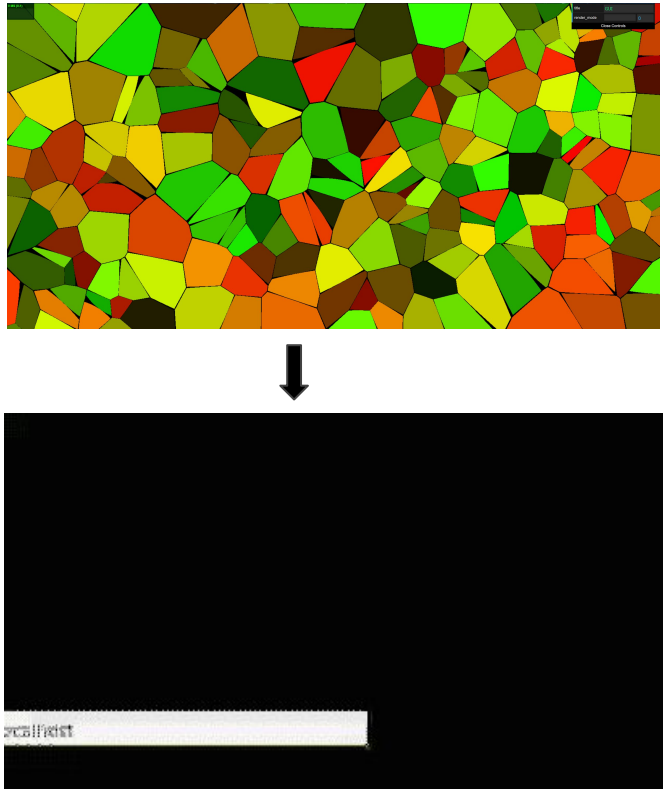
gpujs - pipeline changes for this time



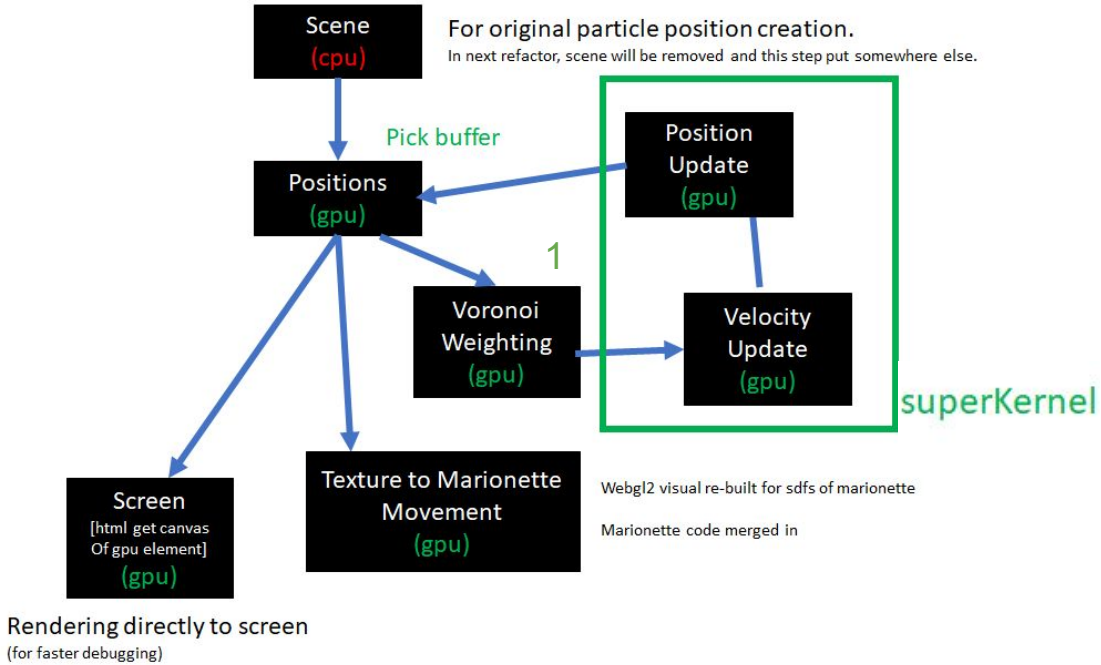
Note: these two images are not from the same initial states but are just a representative example of the change.



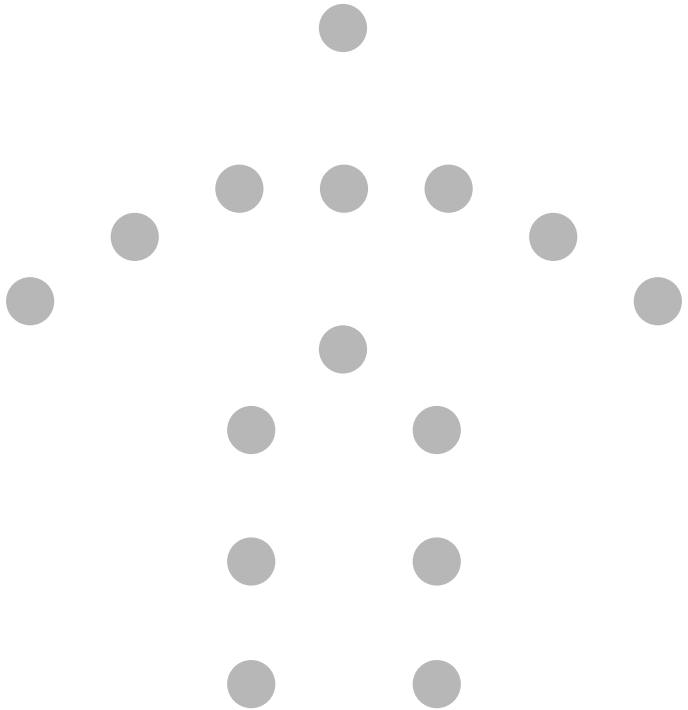
gpujs - pipeline changes for this time



Note: jittery movement is due to getCanvas() output for debug visual



Indexing works!

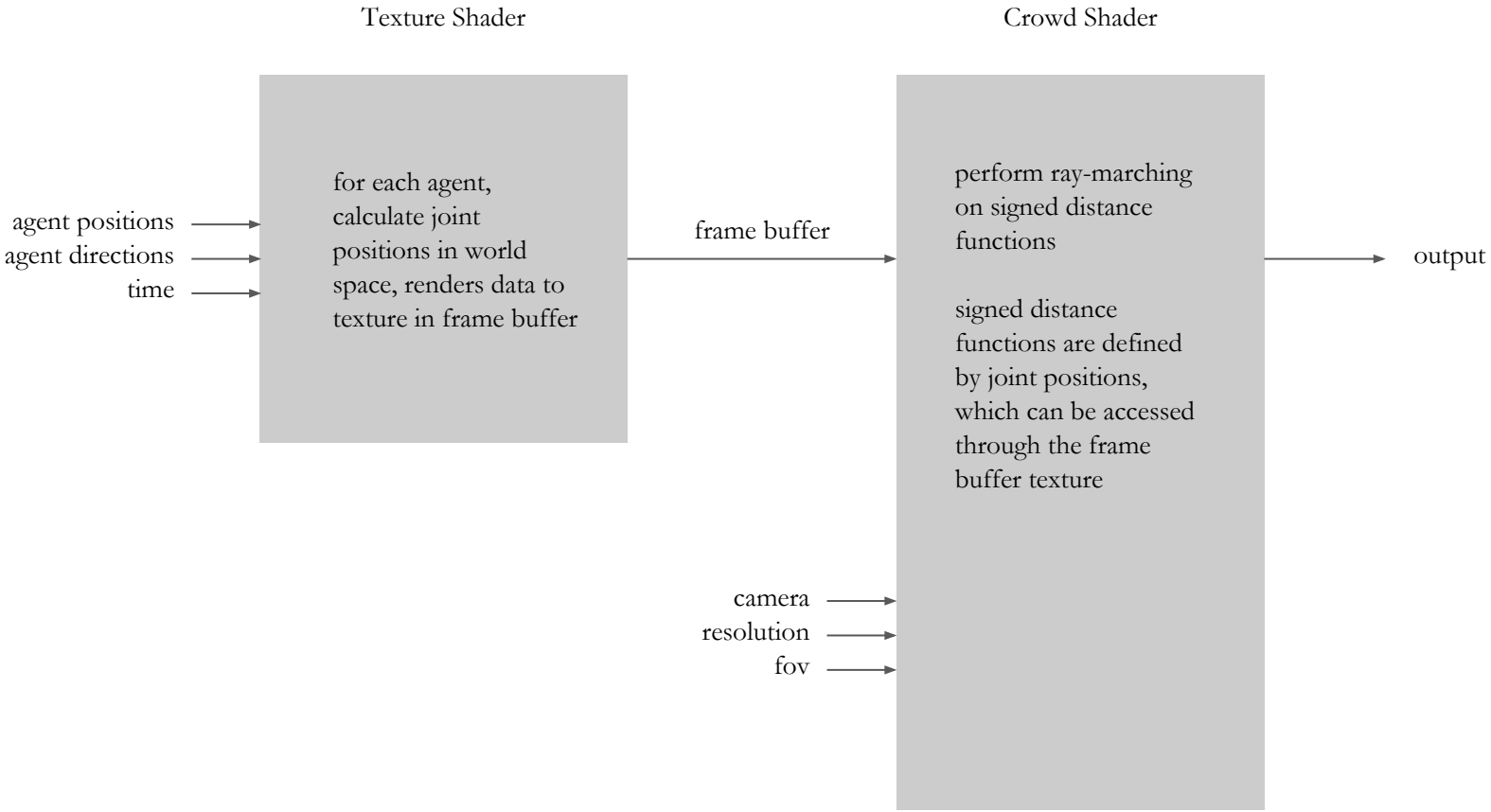


15 joints

now let’s scale this up

15 joints x 100,000 agents = 1,500,000 animated joint positions

have to calculate 1,500,000 animated joint positions based on time, agent position, and agent forward direction before ray-marching

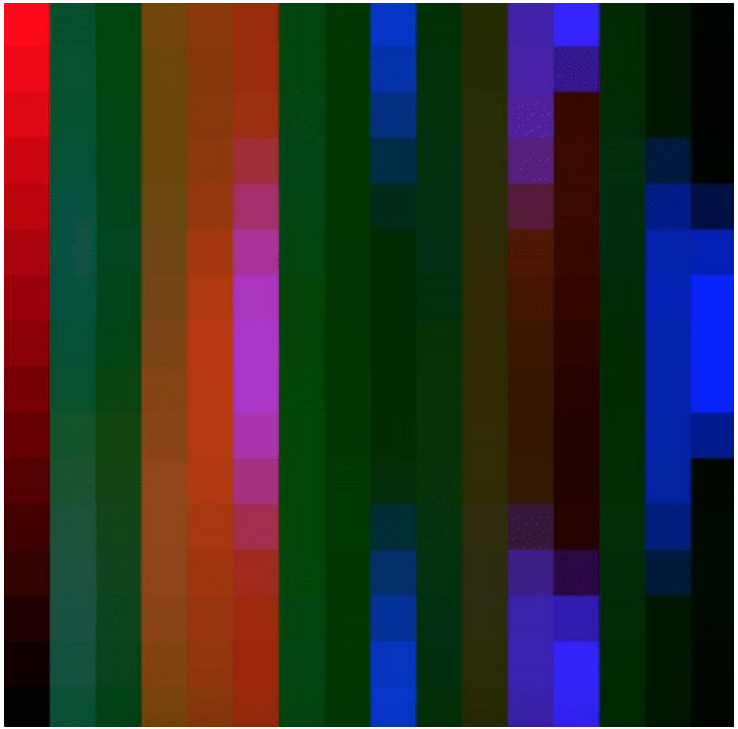

Texture Shader
Crowd Shader
agent positions
agent directions
time
for each agent, calculate joint positions in world space, renders data to texture in frame buffer
frame buffer
perform ray-marching on signed distance functions
signed distance functions are defined by joint positions, which can be accessed through the frame buffer texture
output
camera
resolution
fov

16 pixels per agent

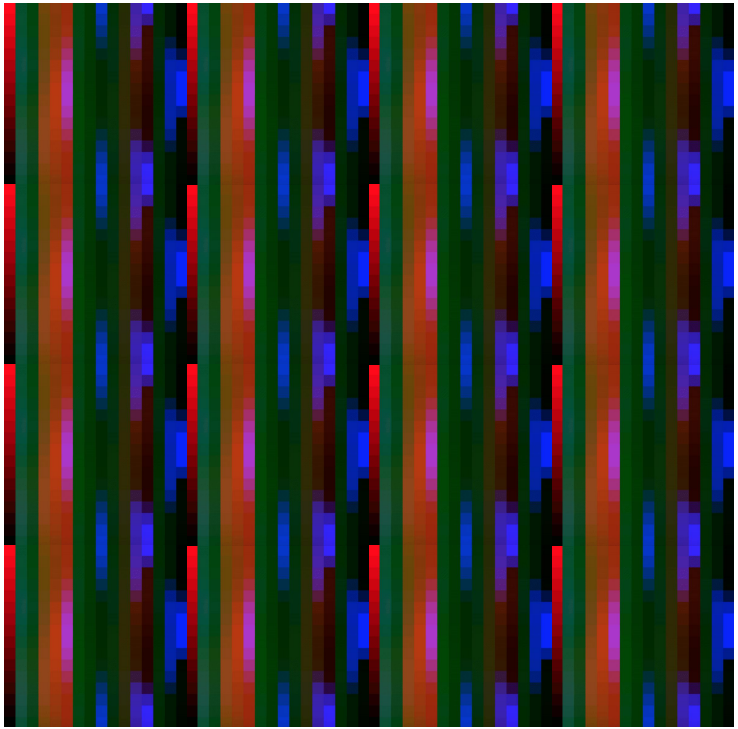
agent pos | joint0 pos | joint1 pos | joint2 pos | ... | joint11 pos | joint12 pos | joint13 pos | joint14 pos

...

agent6	agent7
agent4	agent5
agent2	agent3
agent0	agent1



16 agents



256 agents

Milestone 3 **Issues resolved from last milestone**

- gpu.js
  - Fixed indexing for visual for texture passes (so have velocity update and movement)
  - also doing a gimmicky second canvas to get an additional 2d context so can have debug visual of texture passed through of final voronoi output - might remove this later for speed and/or might find a better way to optimize this.
- Crowd Vis
  - Merging the crowd simulation visuals into the WebGL2 setup of master project
  - Made marionette animation move based on position and direction vector inputs

Milestone 3 **New Features**

- gpu.js
  - Movement through texture-pass
  - shortened number of passes over voronoi from 2 to 1
  - compressed velocityUpdate and positionUpdate into a superKernel
  - instead of doing memory buffer swap by actually copying memory over, just switching reference variables for buffer update.
- Crowd Vis
  - Moved crowd sim shader code to separate files using grunt.js
  - Added agent data texture shader code
  - Utilized frame buffer to store thousands of agent joint positions
  - Passed frame buffer texture to old crowd sim shader code to use

Milestone 3 **For Final Project Demo**

- gpu.js
  - Remove gimmicky second 2d canvas if needed
- Crowd Vis
  - Randomizing character physical traits
- COMBINE THE TWO
  - Difficulty of entire project hasn't been the algorithm itself, but getting the pipelines to work with one another.