

Crowd Particles in Web3d

Implemented as extensions to current web3d v4 standard
in one web3d viewer FreeWRL

Oct 2023

Scene file recordings

- [FreeWRL HAnim transition.mp4](#)
- [FreeWRL ParticleSystem Gravity.mp4](#)
- [top view.mp4](#)
- [ground view.mp4](#)

Changes to web3d nodes and fields:

- ParticlePhysics > new nodes
 - MapEmitter
 - MapPhysics
- Humanoid Animation (Hanim) > new nodes
 - HAnimPermuter
- ParticlePhysics > change
 - SFString geometryType > add “HANIM” to possible types
- HAnim > changes
 - SFTime HAnimHumanoid.transitionTime 0

HAnimPermuter

For geometryType="HANIM" code detects geometry node as one of:

- HAnimHumanoid (one humanoid and one walking motion)
- HAnimPermuter (any number of humanoids and walking motions)

```
HAnimPermuter : X3DChildNode {  
    MFNode [in,out] humanoids [] [HAnimHumanoid]  
    MFNode [in,out] motions [] [HAnimMotion, HAnimMotionData*]  
    SFBool [in,out] compute FALSE  
    MFInt32 [out] permutations // humanoid index *1000 + motion index  
    SFInt32 [in,out] index -1  
    SFNode [out] humanoid [HAnimHumanoid]  
}
```

FreeWRL refactoring of HAnimMotion nodes

Current web3d.org specifications:

- HAnimMotion – contains animation data and animation pointer

Proposed and implemented in FreeWRL and the test scenes:

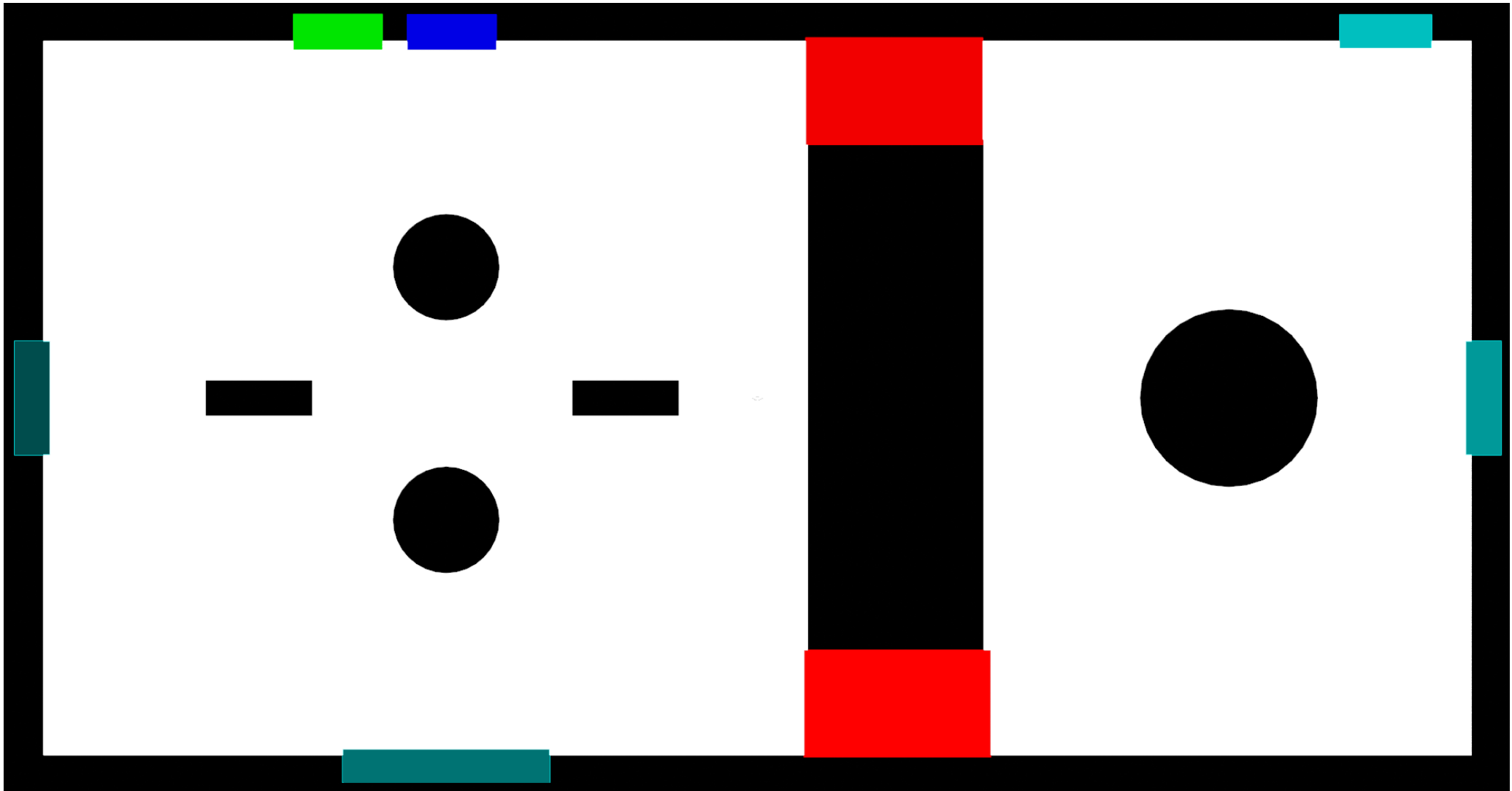
- HAnimMotion : HAnimMotionPlay, HAnimMotionData
- HAnimMotionPlay – contains animation pointer
- HAnimMotionData – contains animation data
- HAnimMotionDataFile : HAnimMotionData – url to an animation file format such as .bvh

MapEmitter

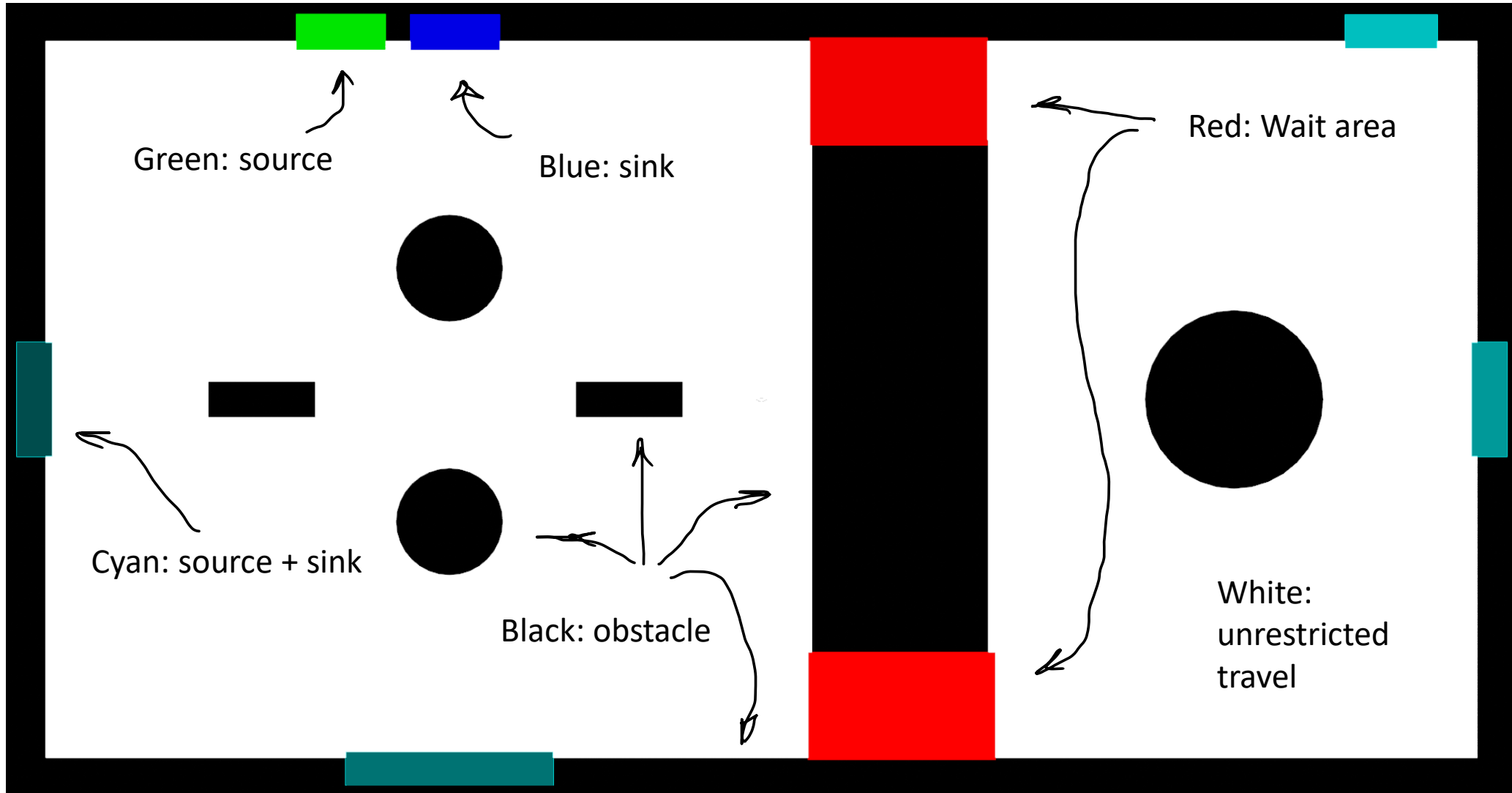
```
MapEmitter : X3DParticleEmitterNode {  
    # standard X3DParticleEmitter emitter fields  
        SFVec3f [in,out] position 0 0 0  
        SFFloat [in,out] speed 0  
        SFFloat [in,out] variation 0.25  
    # MapEmitter specific fields  
        SFNode [in,out] functionMap NULL [X3DSingleTextureNode]  
        SFVec2f [] gridSize 1 1  
        MFColor [] emitterColor []  
        SFFloat [] colorMatchTolerance .01  
        SFBool [in,out] classified FALSE  
}
```

Q. Could have used regular geometry PlaneEmitter?

FunctionMap example



FunctionMap example (annotated)



MapPhysics

```
MapPhysicsModel : X3DParticlePhysicsModelNode {  
    SFNode [in,out] functionMap NULL [X3DSingleTextureNode]  
    SFColor [] obstacleColor 0 0 0  
    MFColor [] sinkColor []  
    SFColor [] pauseColor 1 0 0  
    SFBool [in,out] pauseState FALSE  
    SFFloat [] colorMatchTolerance .01  
    SFVec2f [] gridSize 1 1  
    SFBool [in,out] classified FALSE  
}
```

Q. Could have used shortest path in pre-planned 2D network?

Sinkmap example – stored as uint16 internally



Sink map algorithm – breadth first

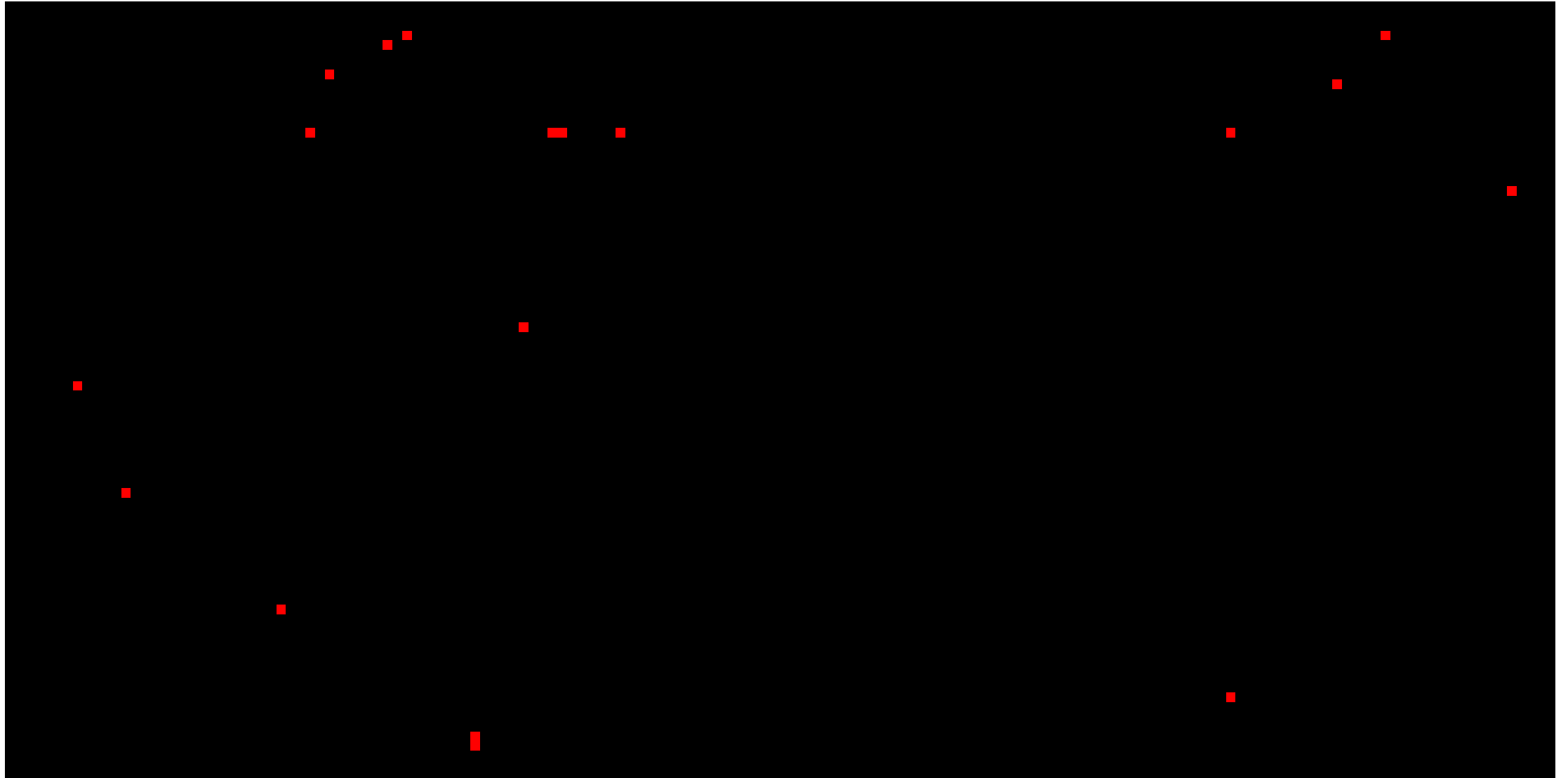
					8	6	4	2	0	2	4	6	8	10					
					9	7	5	3	2	3	5	7	9	11					
					10	8	7	5	4	5	7	8	10	12					
					11	10	8	7	6	7	8	10	11	13					
					13	11	10	9	8	9	10	11	13	14					
											?	13	14	15					

? = (9 + 3) or (10 + 2) or (11+3) or (13 + 2) ?

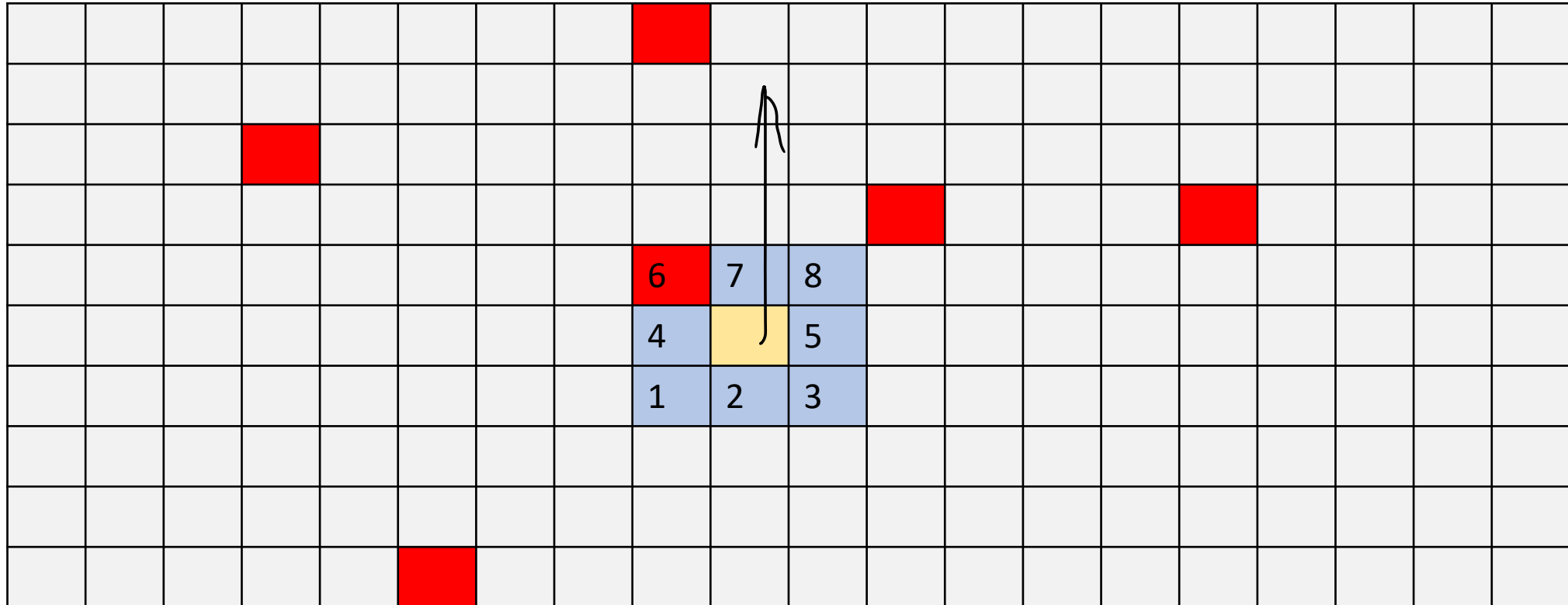
Choose smallest (assuming humans take shortest path)

Re-Used: 2 vectors (orange, green) and alternated

Population map – updated when a particle moves



Collision function



Checks sinkmap surrounding cells for path distance

Checks popmap and eliminates any occupied cells before choosing shortest path

Wait function

- ROUTE to MapPhysics.pauseState (SFBool)
 - TRUE means pause
 - Particles check function map to see if on pause area
 - If not on pause area and next step would be, then wait, else proceed
 - Used for traffic crosswalk pedestrian light

Wait function

								6	7	8									
								4		5									
								1	2	3									

Internal particle structure changes

Enough stored in particle so HAnimHumanoid and HAnimMotion nodes can be recycled / used for multiple instances

```
typedef struct {  
    //... common particle values such as position, velocity  
    //added for HANIM particles, but could be used for other particles:  
        float direction[3]; //normalized last non-zero velocity vector  
    //added for mapemitter method  
        int sink; //assigned after birth in MapPhysics  
        int maplocation[2]; //last popmap location in MapPhysics  
        int paused; //HANIM 0= use first motion 1= use second motion  
        double transitionStart[2];  
        double transitionTime[2];  
        double _startTime[2];  
        int lastMotionsEnabled[2];  
    //added for HANIM particles: mapemitter + HAnimPermuter method  
        int permutationIndex;  
} particle;
```


ParticleSystem loop changes:

- Split between
 - HANIM particles (HAnimHumanoid, HAnimPermuter)
 - Rest of geometry types
- HANIM
 - ignores ParticleSystem : Shape materials
 - Delegates appearance, shader setup, rendering to HAnimHumanoid rendering
 - Recycles HAnimMotion (or HAnimMotionPlay) by saving/restoring parameters to/from particle structure on each Humanoid draw
 - 2 Motion nodes per particle: Stand, Walk
 - If HAnimPermuter node, a combination of (humanoid+motion) is chosen at random for a particle during particle initialization

The End

References:

- [https://freewrl.sourceforge.io/tests/40 Particle systems/crowd/](https://freewrl.sourceforge.io/tests/40_Particle_systems/crowd/)
- <https://stackoverflow.com/questions/367226/flood-fill-algorithms>
- <https://www.web3d.org/standards/hanim>
- <https://www.web3d.org/documents/specifications/19775-1/V4.0/Part01/components/particleSystems.html>