ASSASSIN'S
CREED®
UNITY

Massive crowd on ACU: AI Recycling

Talk by François Cournoyer and Antoine Fortier

The goal

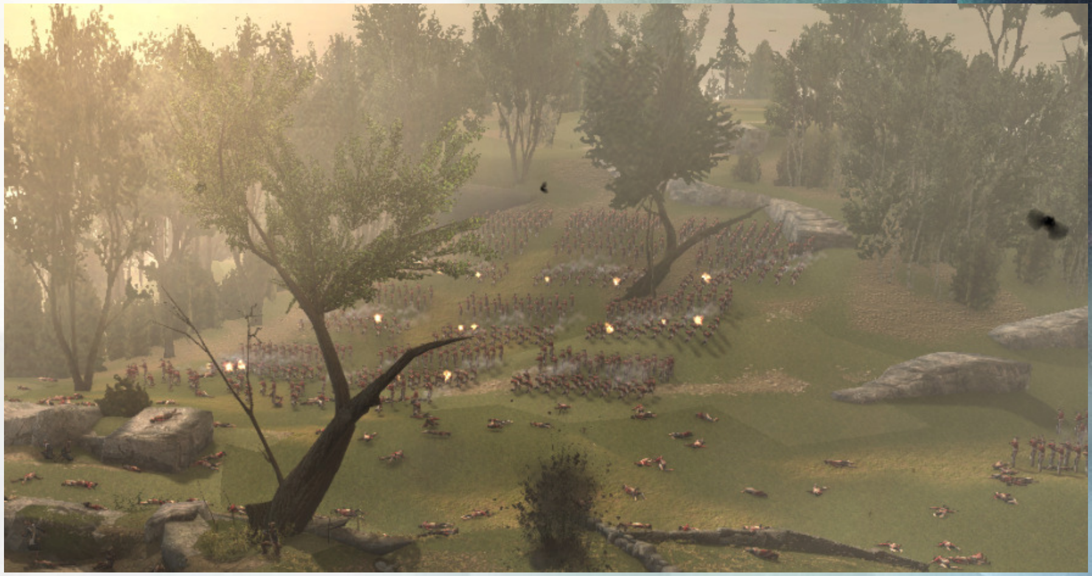
- Goal since AC1
- Social stealth ingredient
- City full of life

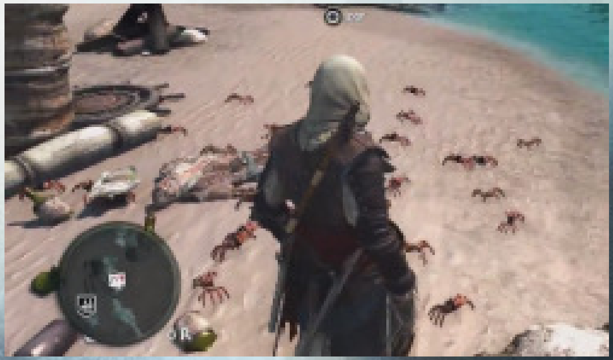
The legacy constraints

- CPU limit of 100 NPCs
- Limit of 20 civilians
- Around 4 around the player
- No systemic crowds

Phase 1

- Rats, crabs and birds
- Far away armies
- No interaction possible
- Lead to AI LOD



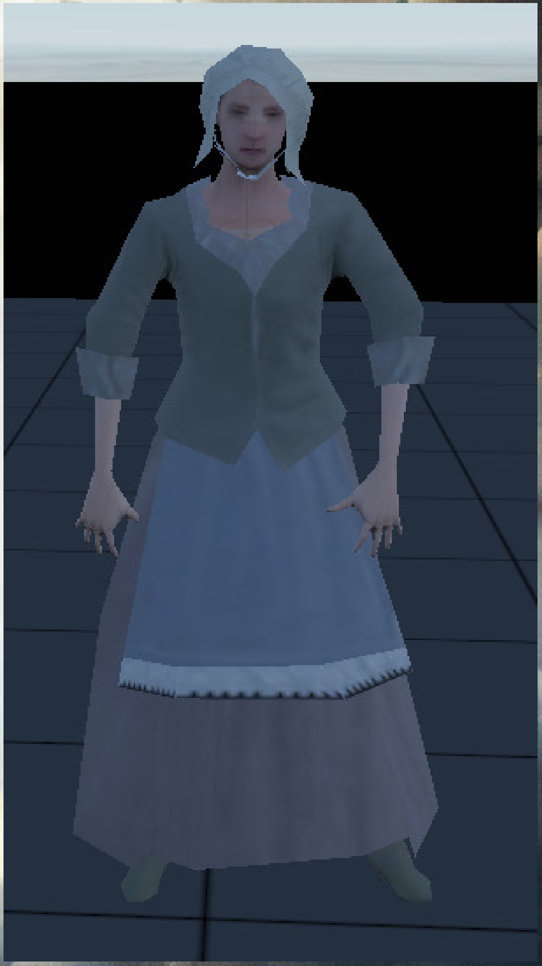


Bulk system overview

- Member is called a bulk
- Thousands of cheap meshes
- Pool of full NPC
- AI LOD based on distance
- Three states for the bulk

Low res bulk

- > 40m
- No entity
- Around 2000 Polygons
- 11 animation bones
- Basic reaction system



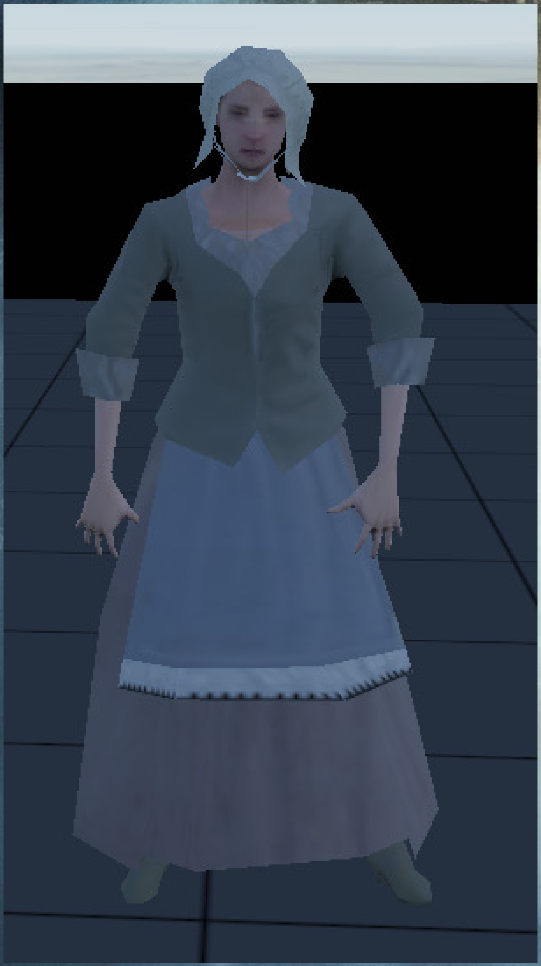
Autonomous bulk

- < 12m
- Entity
- All components are ON
- Costly
- Max of 40

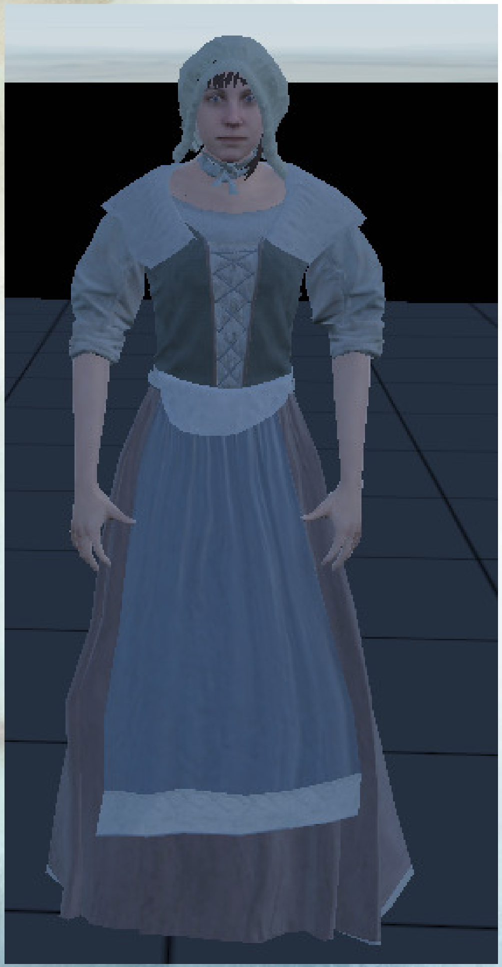
Puppet bulk

- 12 to 40 m
- Real NPC visual mesh
- Low res behavior
- Entity
- Most components are OFF

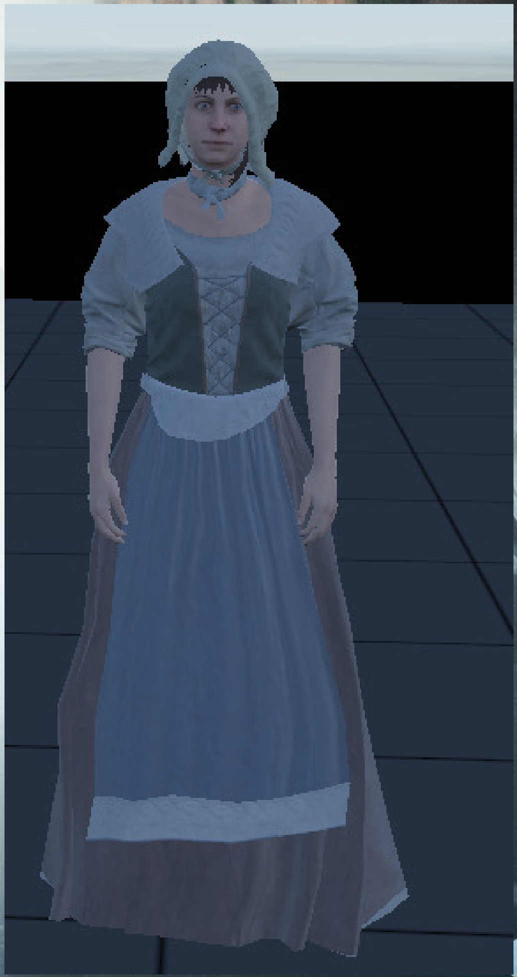
Low Res Bulk

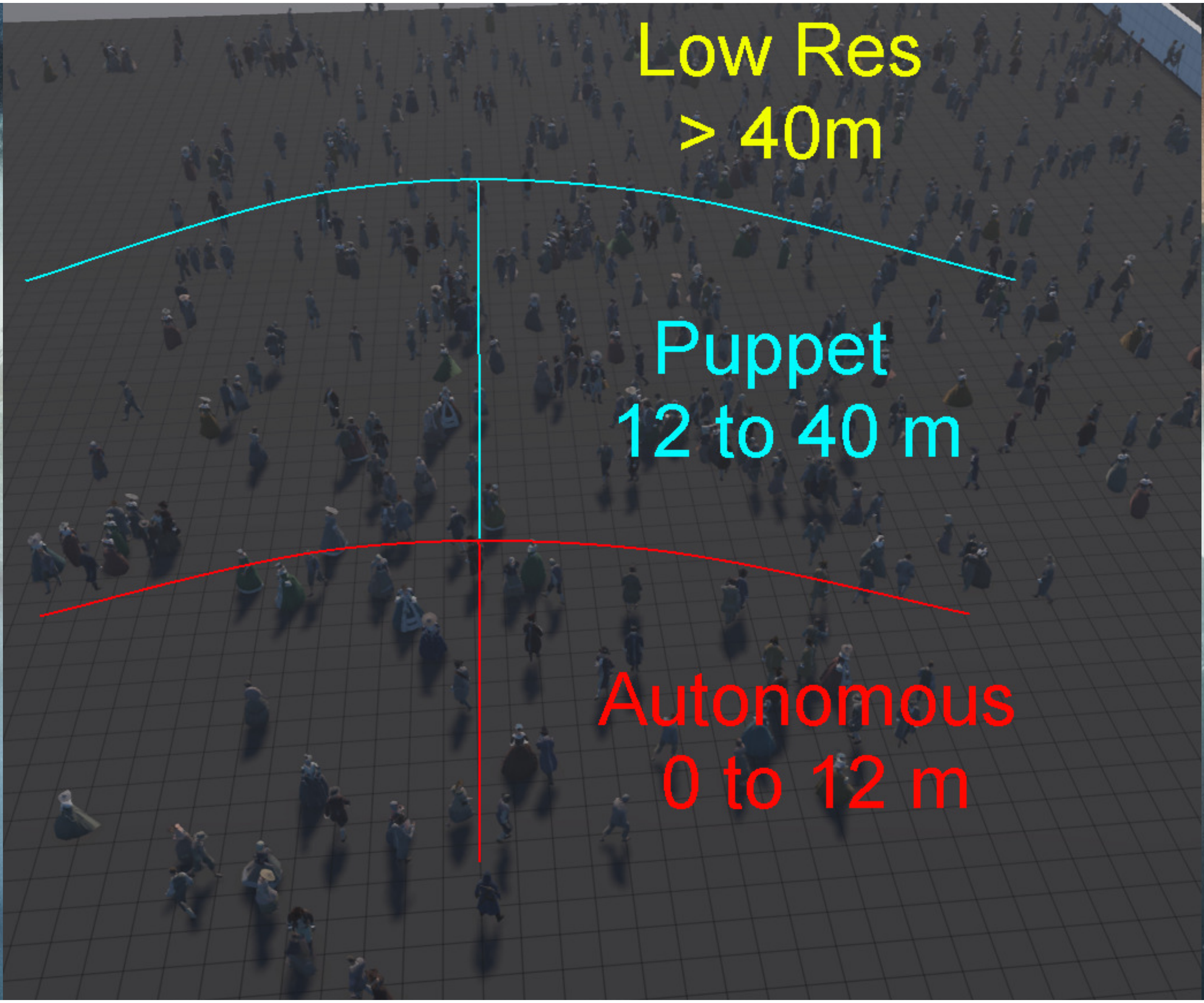


Bulk Puppet



Bulk Autonomous





Low Res
> 40m

Puppet
12 to 40 m

Autonomous
0 to 12 m

Performance metrics

- CPU Costs per bulks:

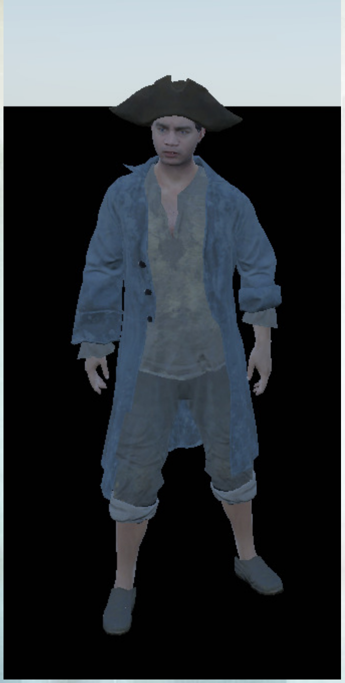
Low Res	Puppet	Autonomous
~25 us	~150us	~500 us to ~5 ms

- Factor ~ 100:1
- Hardware instancing

Low res vs real

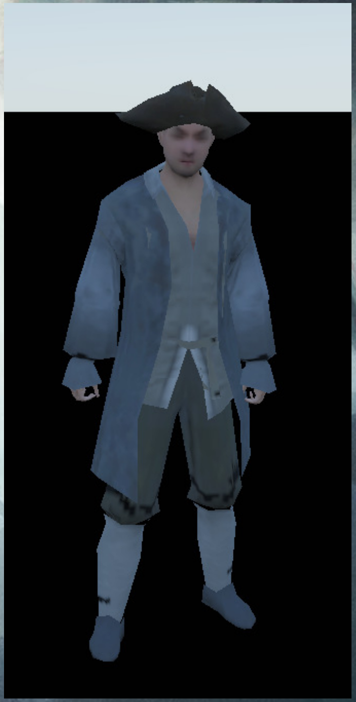
Autonomous Bulk

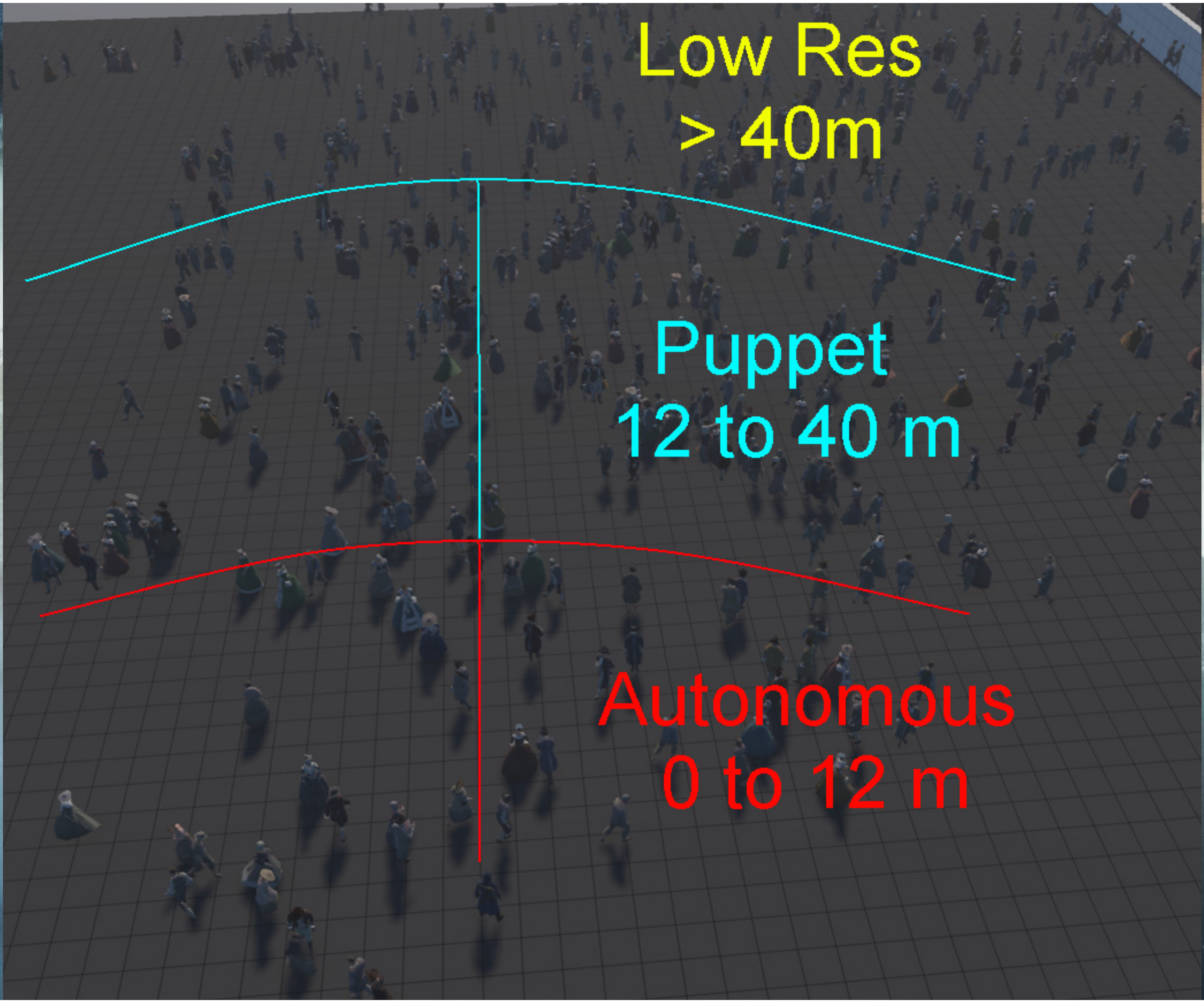
- Infinite Permutations
- Up to 20000 polygons
- 300 animation bones
- No polygon Limit



Low Res Bulk

- 29 different meshes
- Up to 1700 polygons
- 11 animation bones
- 300000 polygons limit





Low Res
> 40m

Puppet
12 to 40 m

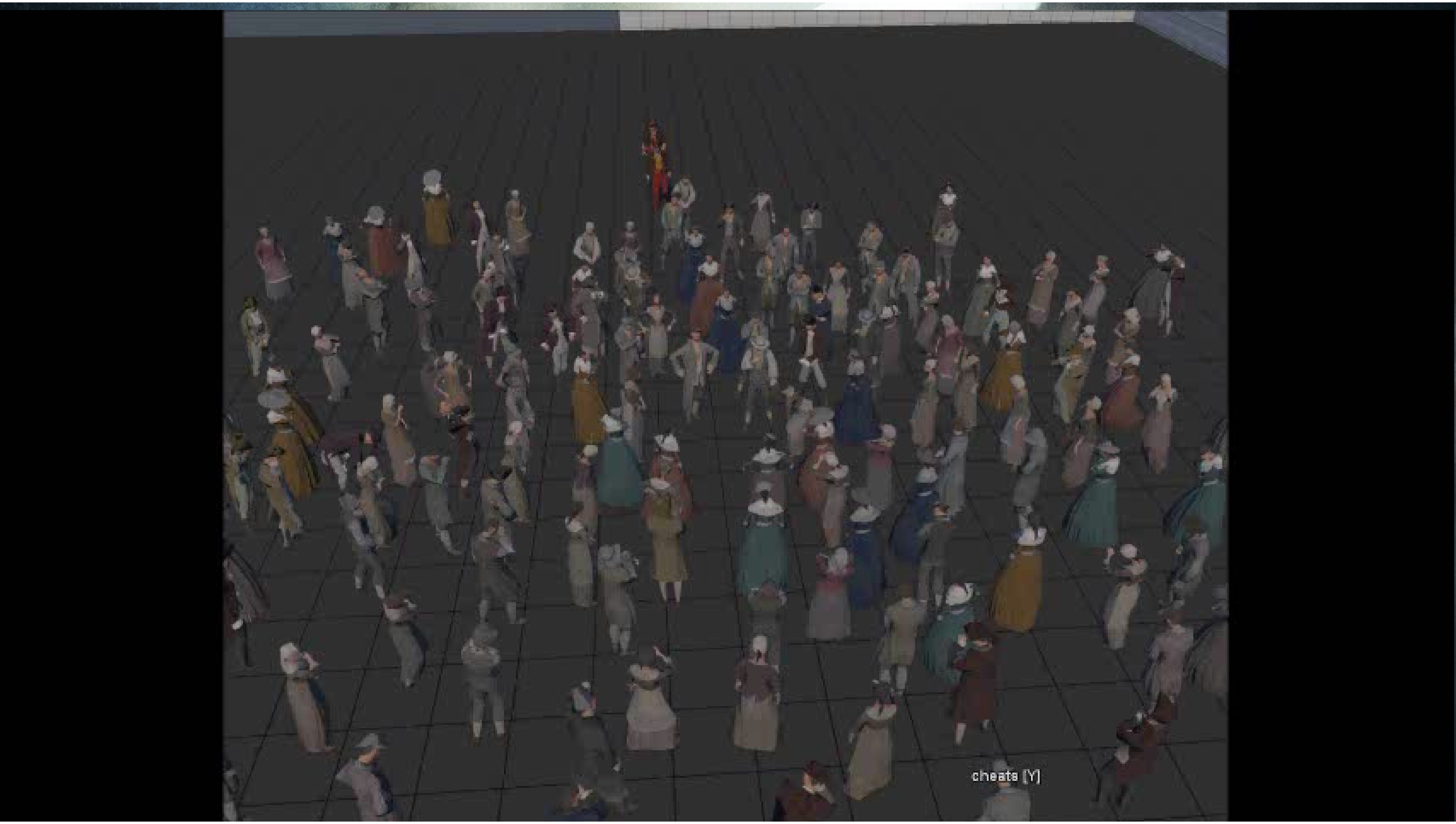
Autonomous
0 to 12 m

Animations

- Custom fast animation system
- Custom animation blender
- Scaling of regular animations
- Stiff hands

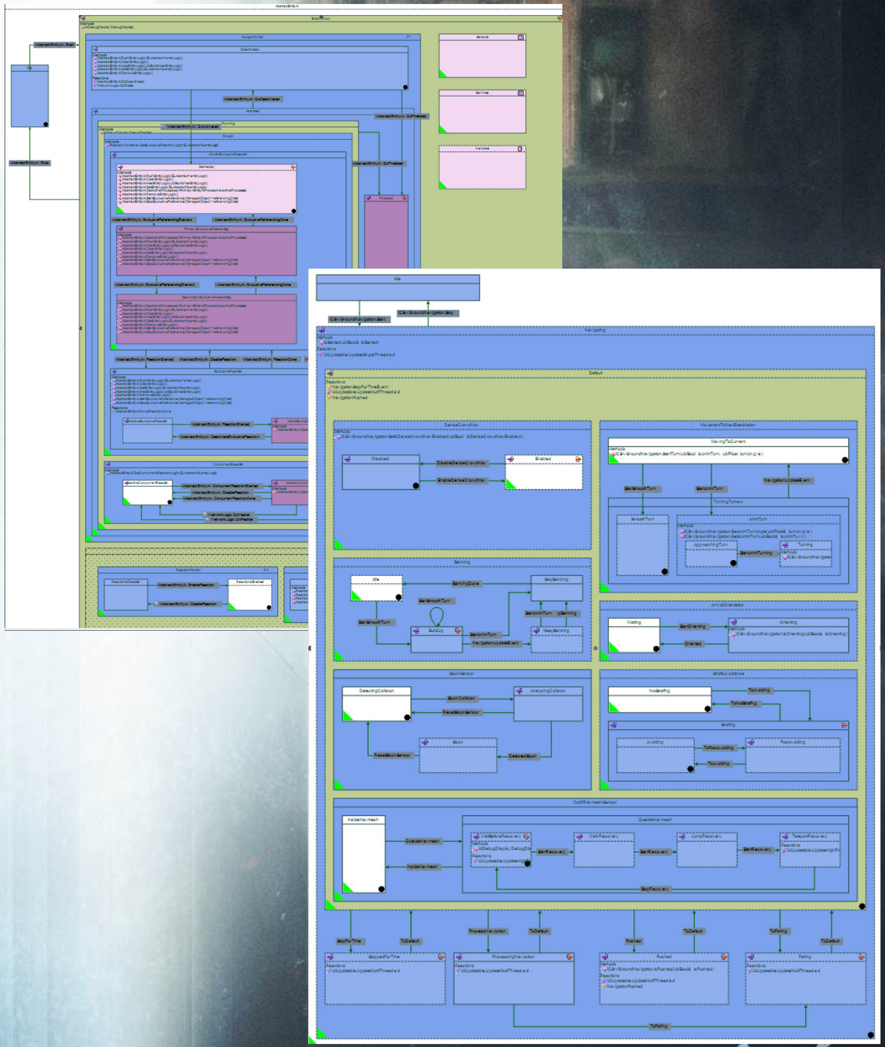
Collision system

- No Havok
- Uses a 2D partition map for queries
- Slides when colliding
- Always clamped on navmesh


cheats [Y]

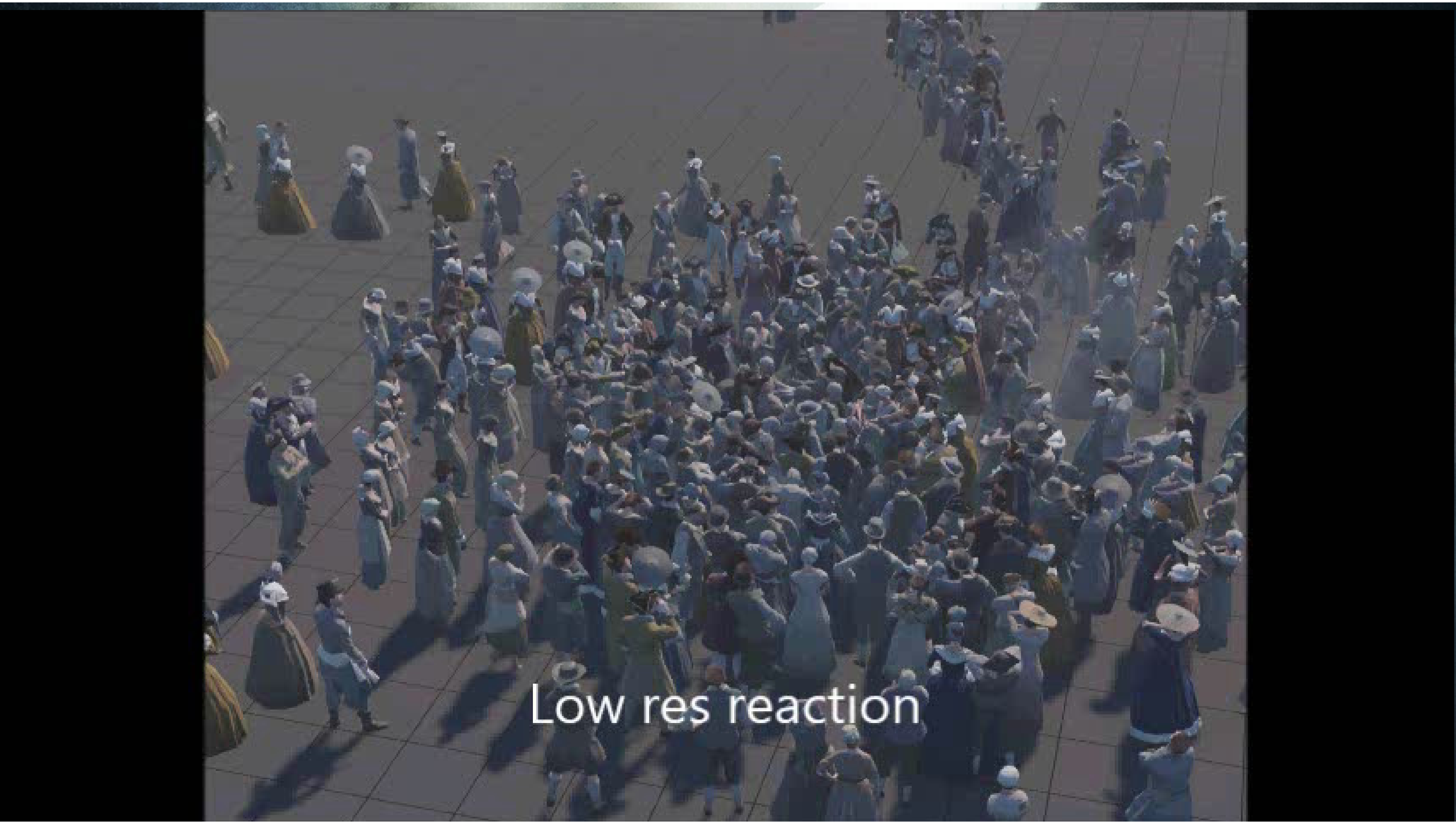
Legacy AI

- Too complex
- Supports all reactions
- All event based
- Very slow



Bulk AI

- Small modular scripts
- No complex reactions
- Tries to follow legacy behavior



Low res reaction

Shepherds

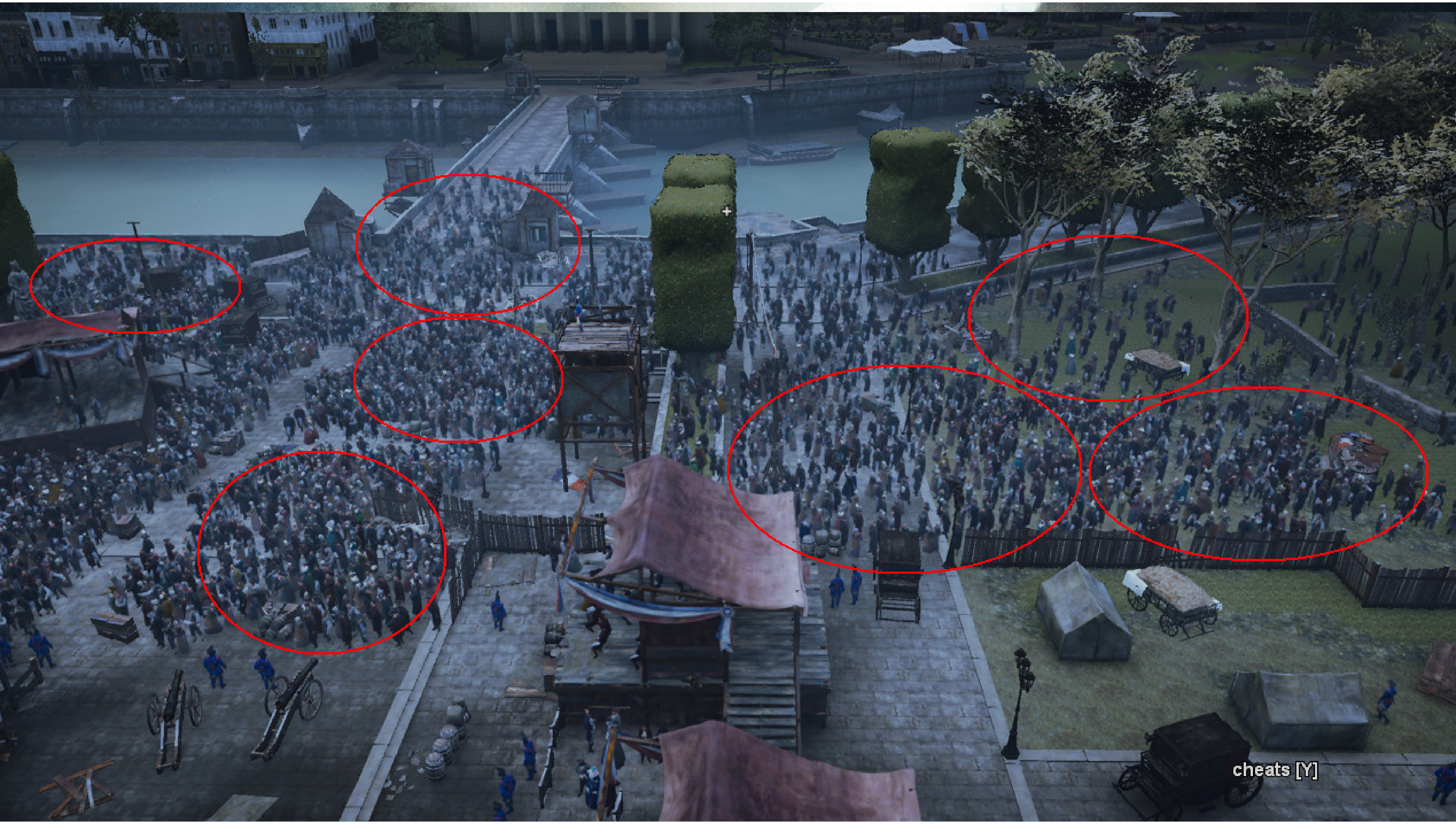
- Holds bulks
- Contains all AI data
- Keeps track of bulk state
- Computes bulk logic

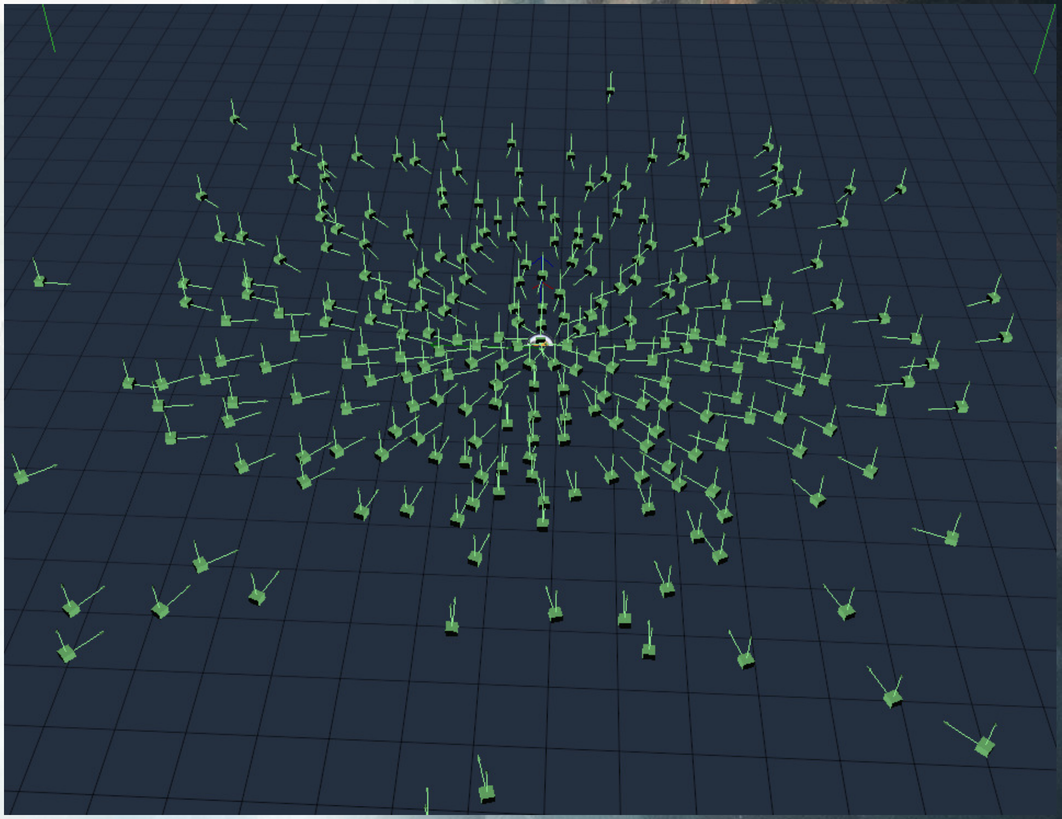
Shepherds

- Unique ID for bulk
- Memory allocated once
- Typically around 30 mb
- Placed by LDs

Level design

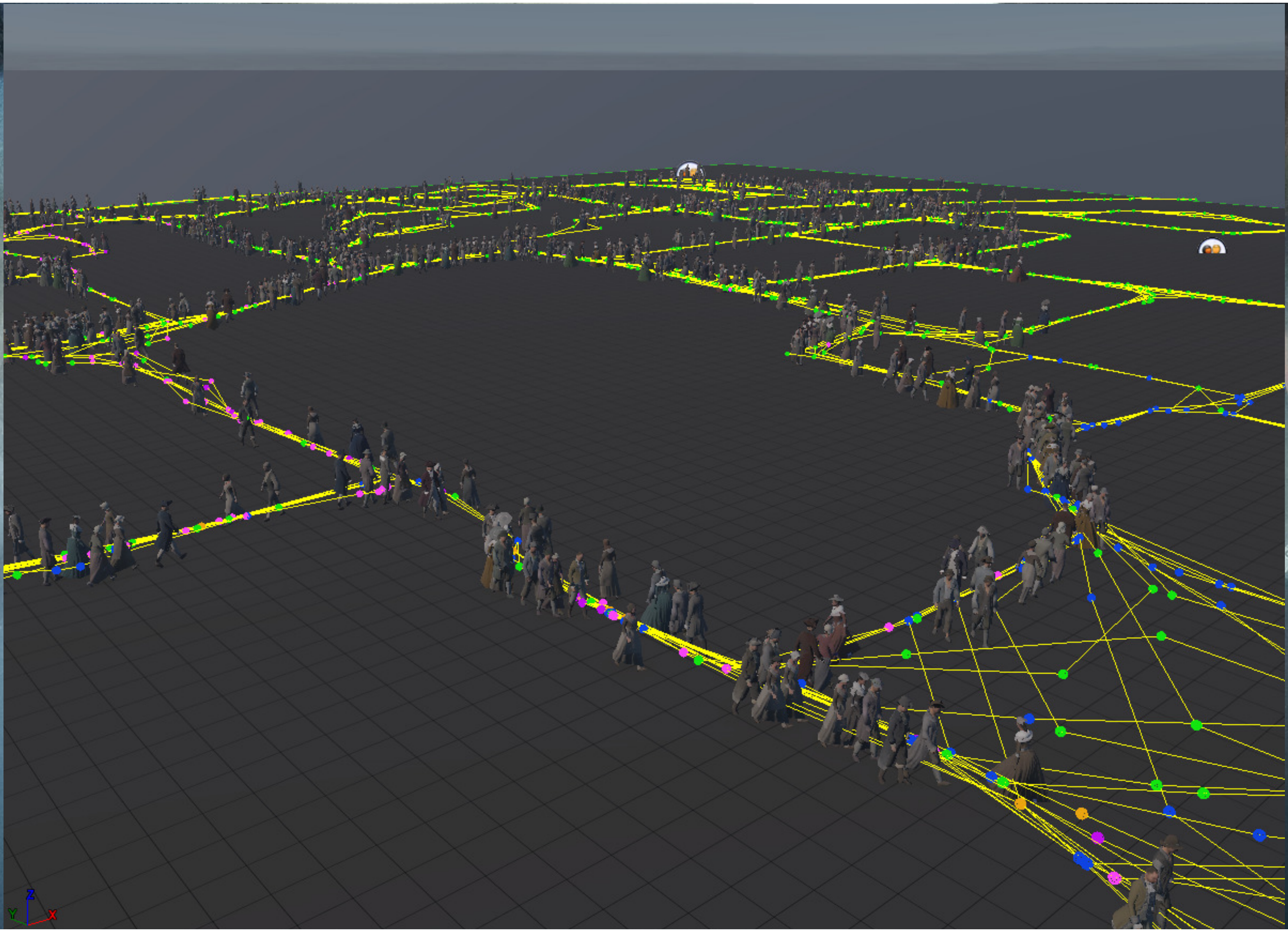
- Can select bulk count and density
- Auto-generated positions
- Can edit specific positions
- Scripting tools for shepherd


cheats [Y]



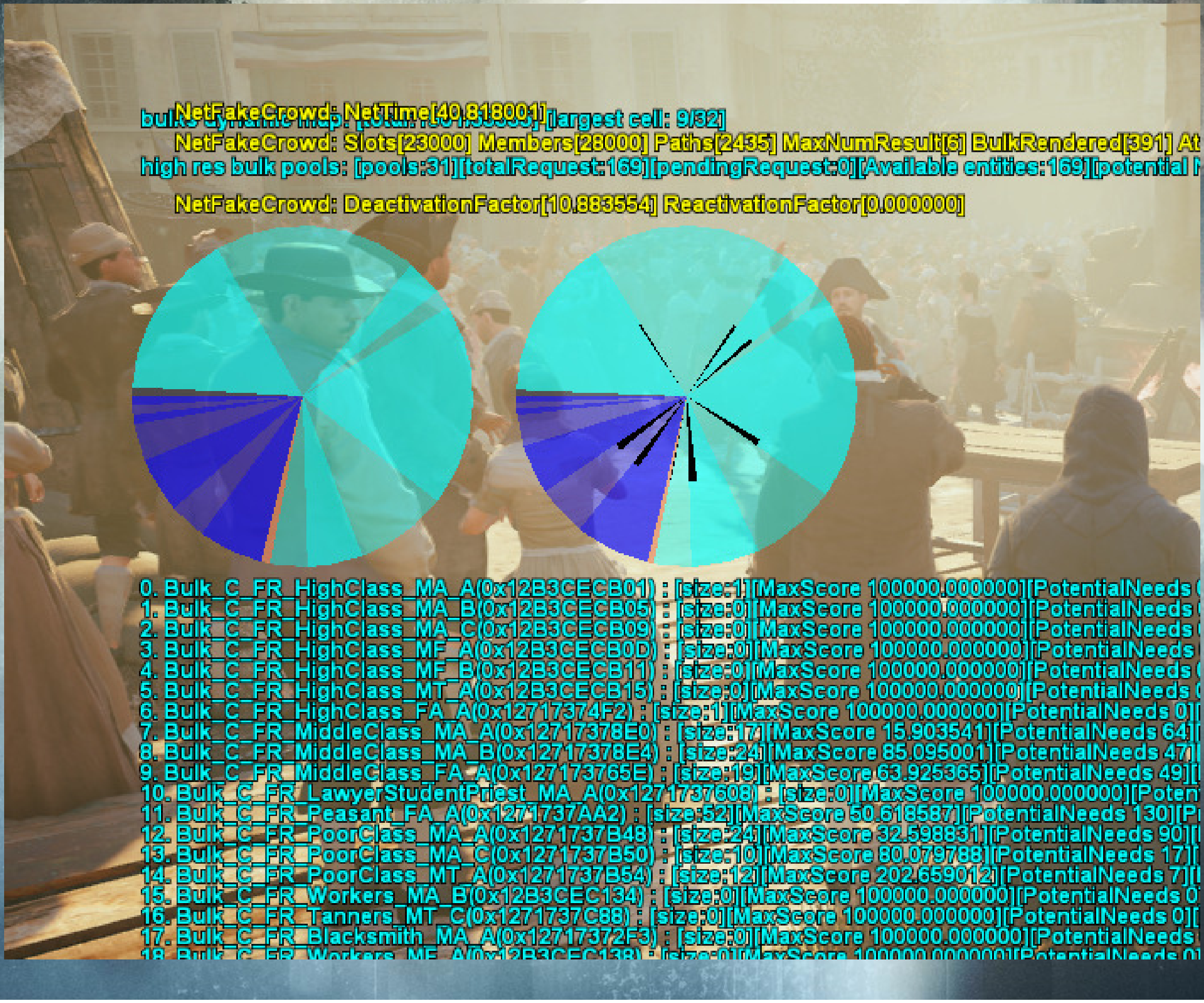
Wandering crowd

- Unique shepherd
- Randomly generated closed paths
- Deterministic
- All of Paris
- Models based on region

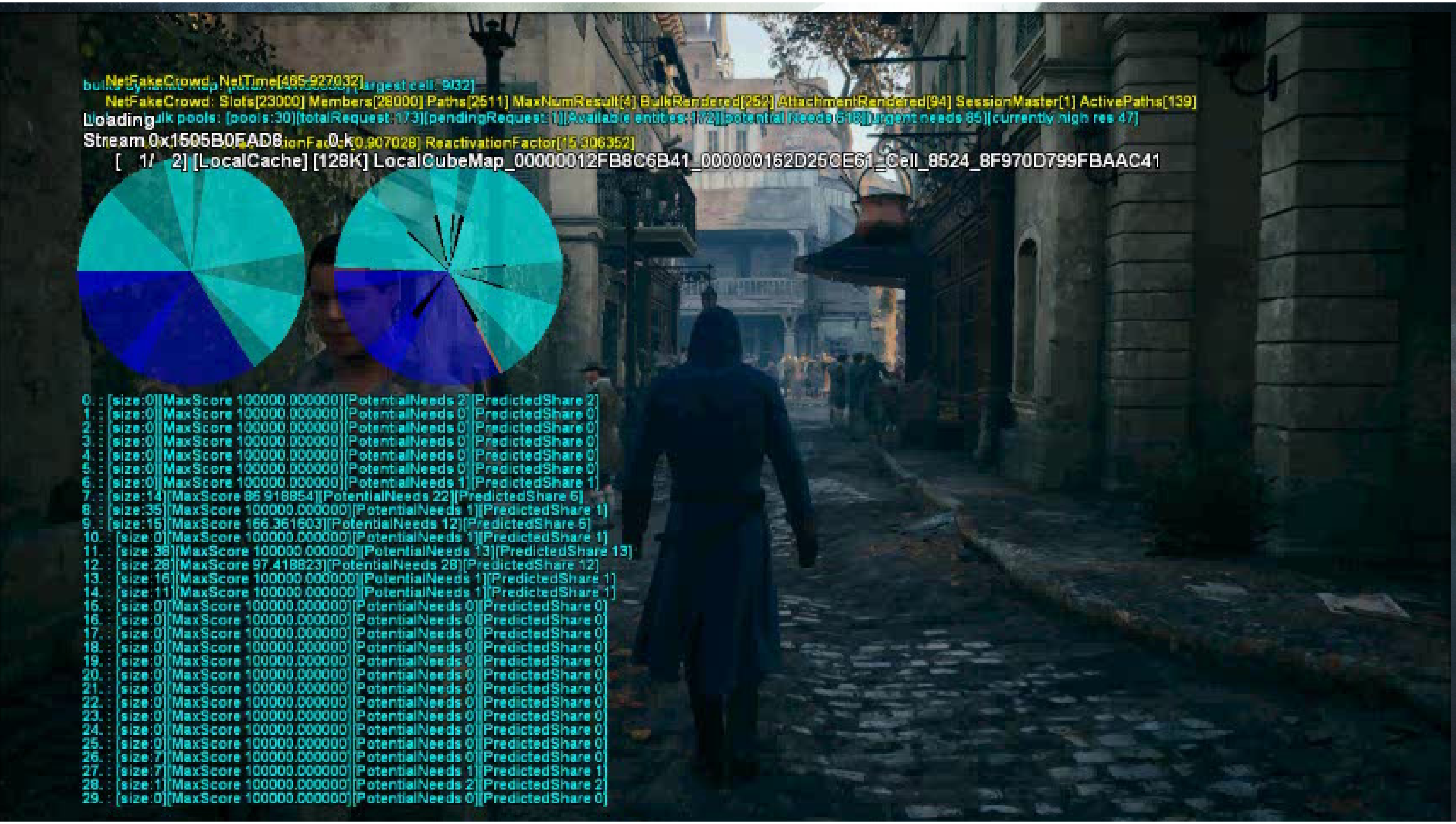

Z
Y
X

The pool

- Tag to match real NPC mesh
- Statistically based on the region composition
- Match low res visual densities
- The pool is constantly adjusting
- Less models means better matching



```
NetFakeCrowd: NetTime[40.818001]
bulks dynamic map: [total: [illegible]] [largest cell: 9/32]
    NetFakeCrowd: Slots[23000] Members[28000] Paths[2435] MaxNumResult[6] BulkRendered[391] At
high res bulk pools: [pools:31][totalRequest:169][pendingRequest:0][Available entities:169][potential N

    NetFakeCrowd: DeactivationFactor[10.883554] ReactivationFactor[0.000000]

0. Bulk_C_FR_HighClass_MA_A(0x12B3CECB01) : [size:1][MaxScore 100000.000000][PotentialNeeds
1. Bulk_C_FR_HighClass_MA_B(0x12B3CECB05) : [size:0][MaxScore 100000.000000][PotentialNeeds
2. Bulk_C_FR_HighClass_MA_C(0x12B3CECB09) : [size:0][MaxScore 100000.000000][PotentialNeeds
3. Bulk_C_FR_HighClass_MF_A(0x12B3CECB0D) : [size:0][MaxScore 100000.000000][PotentialNeeds
4. Bulk_C_FR_HighClass_MF_B(0x12B3CECB11) : [size:0][MaxScore 100000.000000][PotentialNeeds
5. Bulk_C_FR_HighClass_MT_A(0x12B3CECB15) : [size:0][MaxScore 100000.000000][PotentialNeeds
6. Bulk_C_FR_HighClass_FA_A(0x12717374F2) : [size:1][MaxScore 100000.000000][PotentialNeeds 0][
7. Bulk_C_FR_MiddleClass_MA_A(0x12717378E0) : [size:17][MaxScore 15.903541][PotentialNeeds 64][
8. Bulk_C_FR_MiddleClass_MA_B(0x12717378E4) : [size:24][MaxScore 85.095001][PotentialNeeds 47][
9. Bulk_C_FR_MiddleClass_FA_A(0x127173765E) : [size:19][MaxScore 63.925365][PotentialNeeds 49][
10. Bulk_C_FR_LawyerStudentPriest_MA_A(0x1271737608) : [size:0][MaxScore 100000.000000][Poten
11. Bulk_C_FR_Peasant_FA_A(0x1271737AA2) : [size:52][MaxScore 50.618587][PotentialNeeds 130][P
12. Bulk_C_FR_PoorClass_MA_A(0x1271737B48) : [size:24][MaxScore 32.598831][PotentialNeeds 90][
13. Bulk_C_FR_PoorClass_MA_C(0x1271737B50) : [size:10][MaxScore 80.079788][PotentialNeeds 17][
14. Bulk_C_FR_PoorClass_MT_A(0x1271737B54) : [size:12][MaxScore 202.659012][PotentialNeeds 7][
15. Bulk_C_FR_Workers_MA_B(0x12B3CEC134) : [size:0][MaxScore 100000.000000][PotentialNeeds 0
16. Bulk_C_FR_Tanners_MT_C(0x1271737C88) : [size:0][MaxScore 100000.000000][PotentialNeeds 0][
17. Bulk_C_FR_Blacksmith_MA_A(0x12717372F3) : [size:0][MaxScore 100000.000000][PotentialNeeds
18. Bulk_C_FR_Workers_MF_A(0x12B3CEC138) : [size:0][MaxScore 100000.000000][PotentialNeeds 0]
```



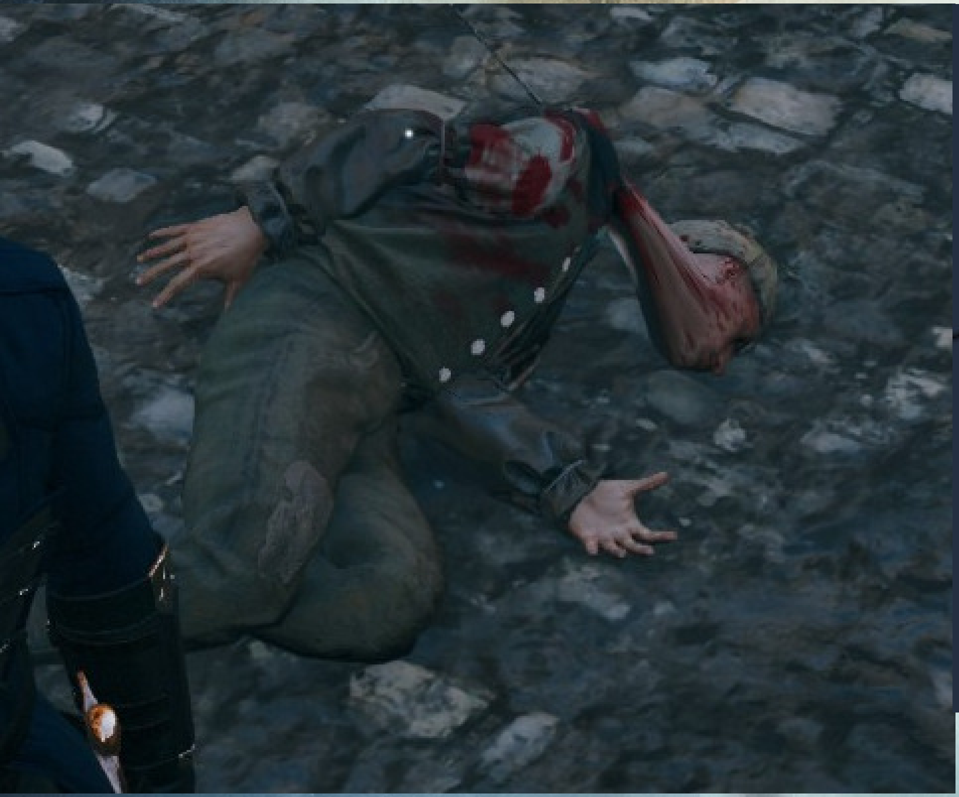
```
      NetFakeCrowd: NetTime[485.927032]
bulk dynamic map: [total: [illegible]] [largest cell: 9032]
      NetFakeCrowd: Slots[23000] Members[28000] Paths[2511] MaxNumResult[4] BulkRendered[252] AttachmentRendered[94] SessionMaster[1] ActivePaths[139]
Loading
bulk pools: [pools:30][totalRequest: 173][pendingRequest: 1][Available entities: 172][potential Needs 518][urgent needs 85][currently high res 47]
Stream 0x1505B0EAD8   0.k
      [illegible]ionFactor[0.907028] ReactivationFactor[15.306352]
      [   1/   2] [LocalCache] [128K] LocalCubeMap_00000012FB8C6B41_000000162D25CE61_Cell_8524_8F970D799FBAAC41

0. : [size:0][MaxScore 100000.000000][PotentialNeeds 2][PredictedShare 2]
1. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
2. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
3. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
4. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
5. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
6. : [size:0][MaxScore 100000.000000][PotentialNeeds 1][PredictedShare 1]
7. : [size:14][MaxScore 86.918854][PotentialNeeds 22][PredictedShare 6]
8. : [size:35][MaxScore 100000.000000][PotentialNeeds 1][PredictedShare 1]
9. : [size:15][MaxScore 166.361603][PotentialNeeds 12][PredictedShare 5]
10. : [size:0][MaxScore 100000.000000][PotentialNeeds 1][PredictedShare 1]
11. : [size:38][MaxScore 100000.000000][PotentialNeeds 13][PredictedShare 13]
12. : [size:28][MaxScore 97.418823][PotentialNeeds 28][PredictedShare 12]
13. : [size:16][MaxScore 100000.000000][PotentialNeeds 1][PredictedShare 1]
14. : [size:11][MaxScore 100000.000000][PotentialNeeds 1][PredictedShare 1]
15. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
16. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
17. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
18. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
19. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
20. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
21. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
22. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
23. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
24. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
25. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
26. : [size:7][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
27. : [size:7][MaxScore 100000.000000][PotentialNeeds 1][PredictedShare 1]
28. : [size:1][MaxScore 100000.000000][PotentialNeeds 2][PredictedShare 2]
29. : [size:0][MaxScore 100000.000000][PotentialNeeds 0][PredictedShare 0]
```

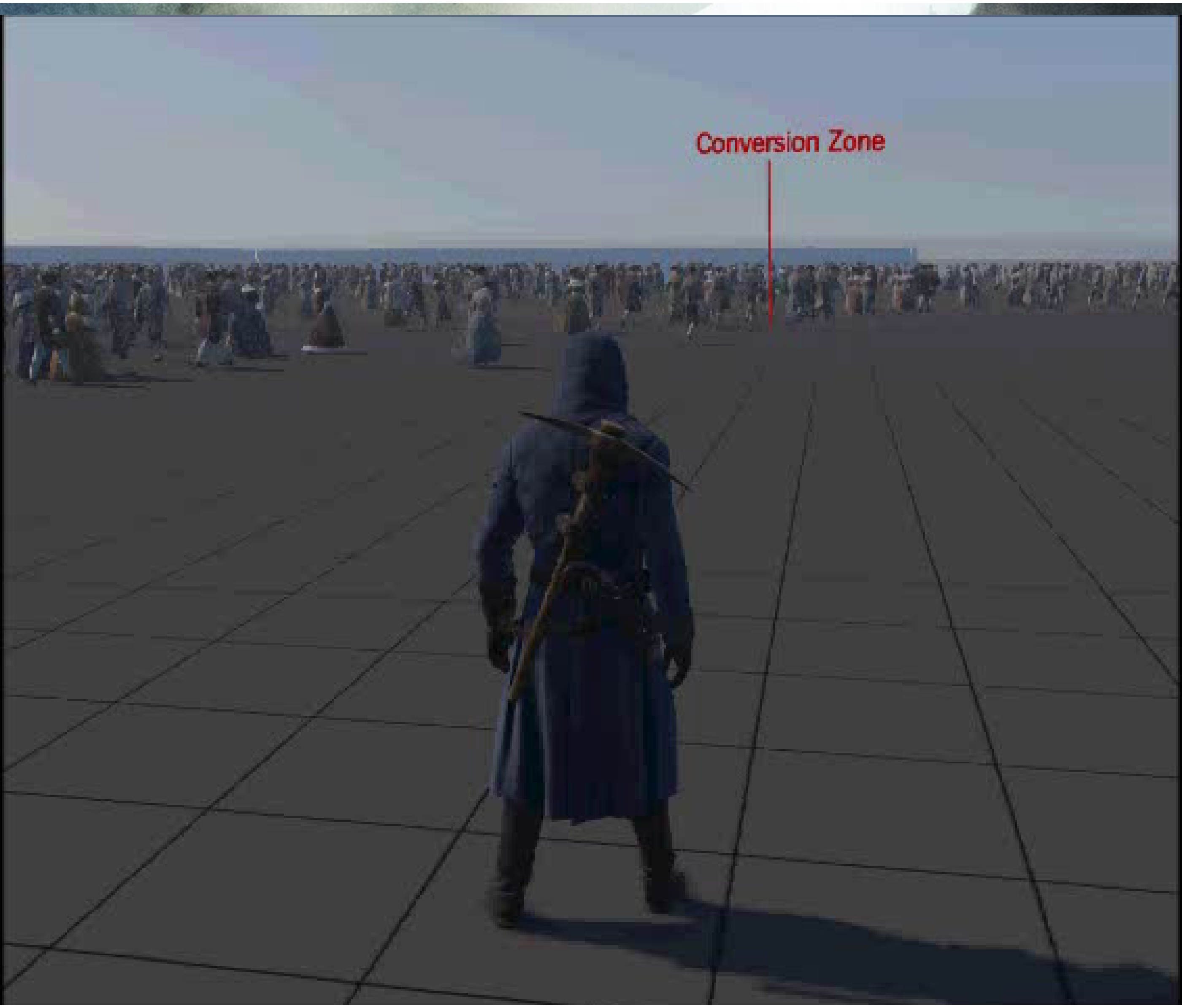
Bulk puppet

- High quality visual
- All components are turned off
- Behaves as a low res bulk

Swapping

- Find the best matching entity
- Reapply color
- Teleport to low res position
- Match the hats and props
- Remapping of the torso bones




Conversion Zone

Matching colors

- Difference in shaders
- Color matching tool
- Matched by character modelers

Hats

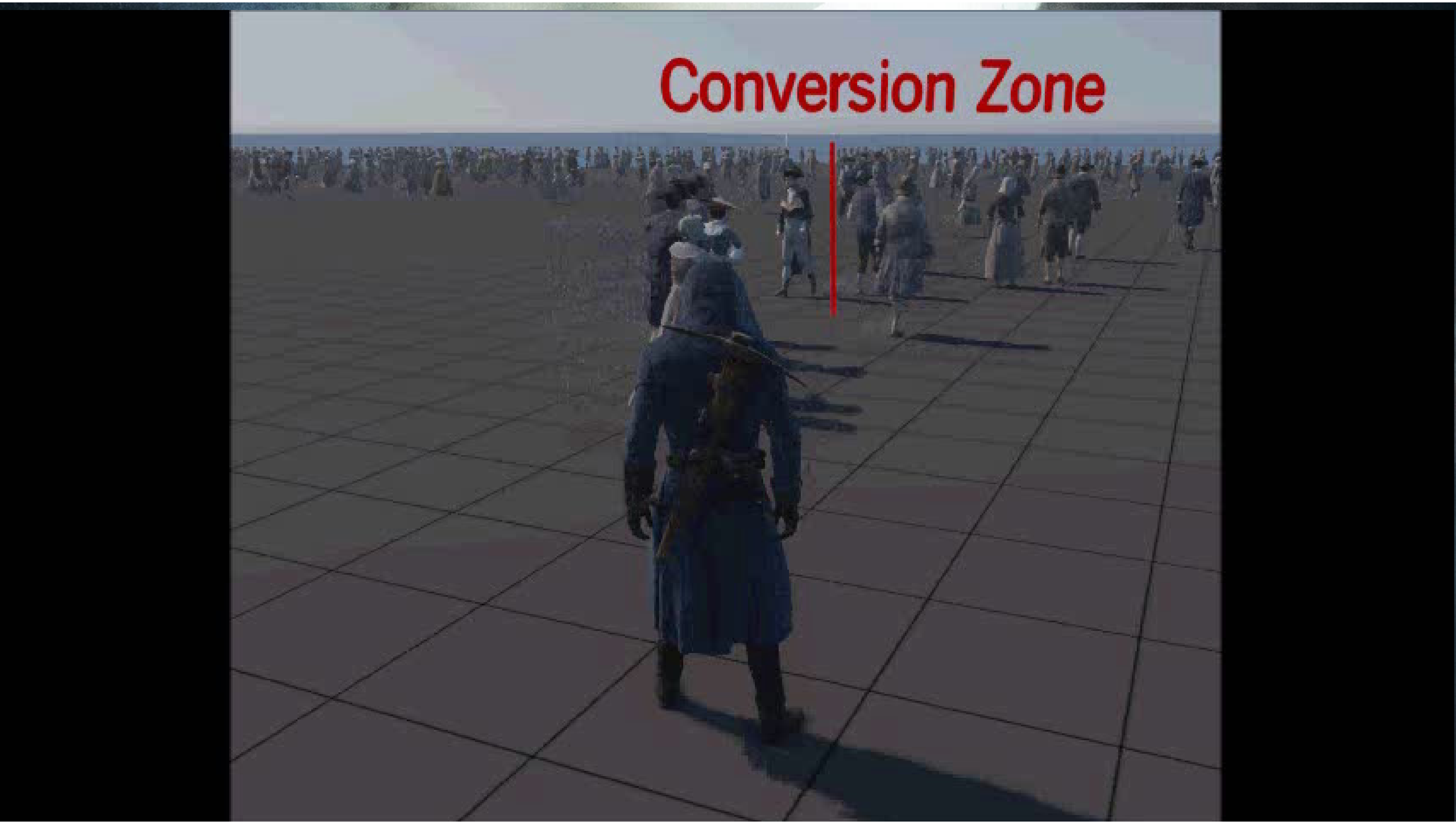
- Creates variety
- Can’t respawn entities
- All hats spawned on real
- Select correct hat on transition

Bulk autonomous

- All components are turned on
- Low res logic is turned off
- Full Havok physics
- Still part of the shepherd

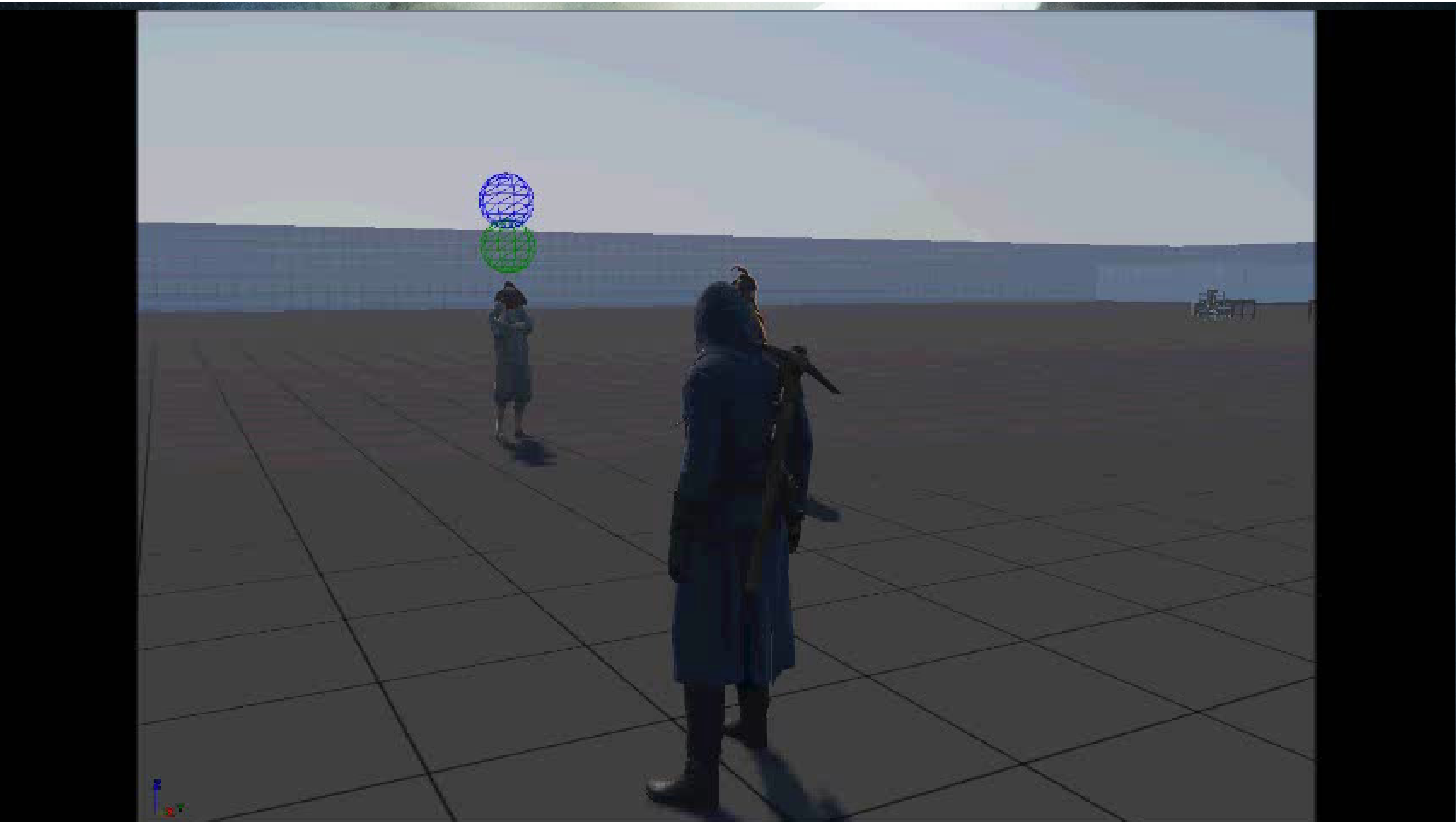
Autonomous conversion

- Reset all modified variables
- Must transition in the correct AI state
- Must be fast


Conversion Zone

Interaction with autonomous

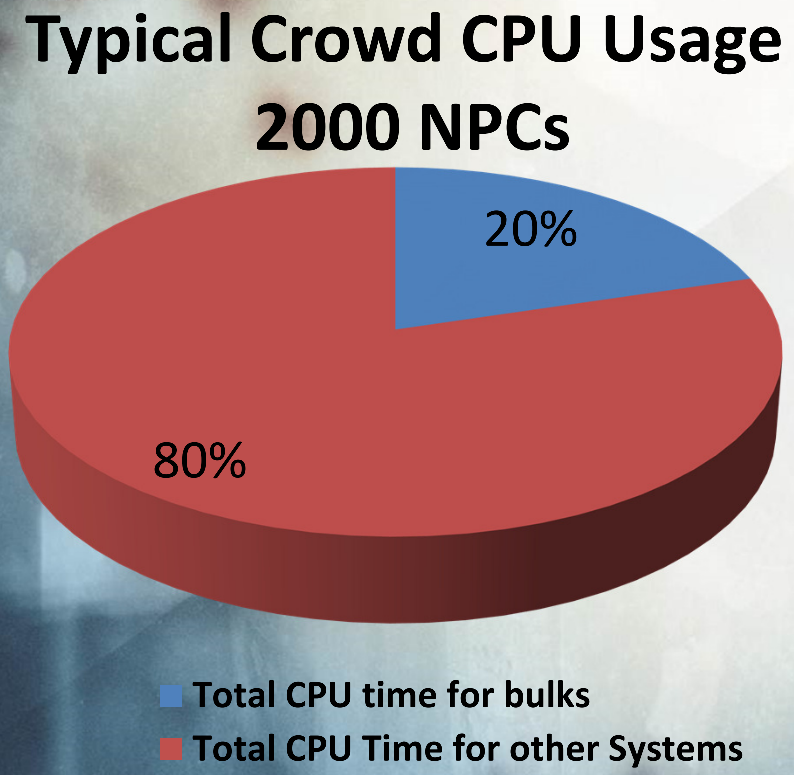
- Interaction entity always changing
- Smart pointers with bulk IDs
- Fallback : Swap with new real NPC

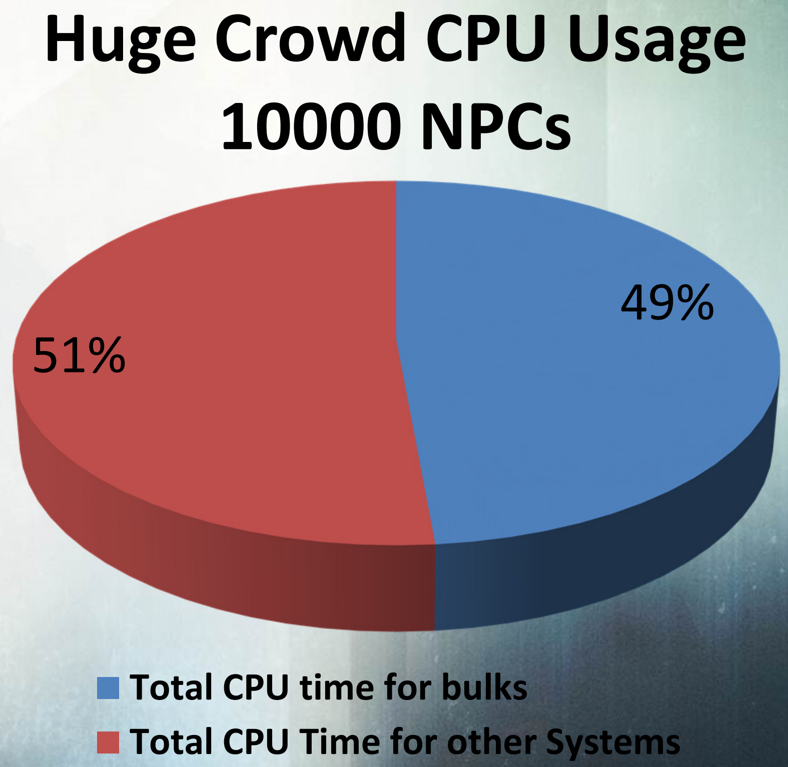


Networking

- Deterministic positions
- Pregenerated paths
- Replicated reactions
- Convert to real NPC

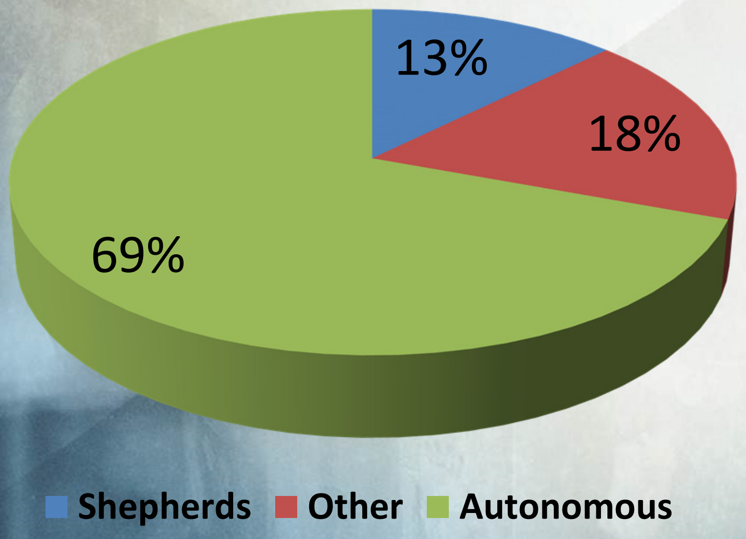
Performance overview


Typical Crowd CPU Usage
2000 NPCs

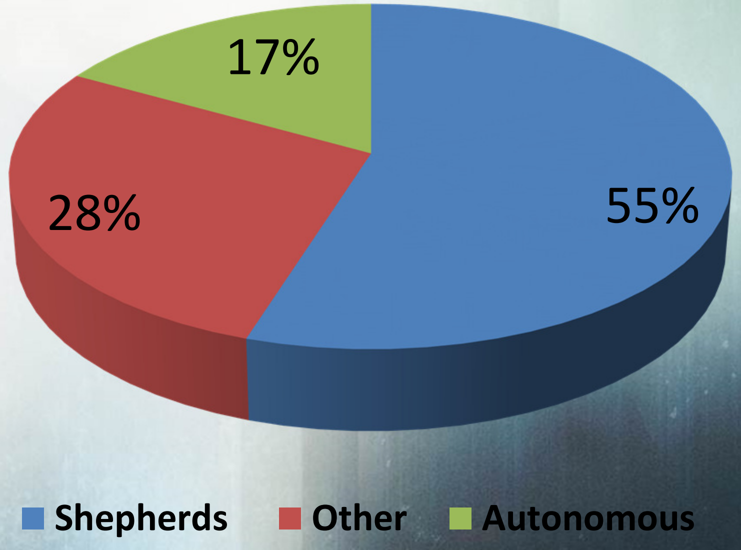

Huge Crowd CPU Usage
10000 NPCs

Performance overview

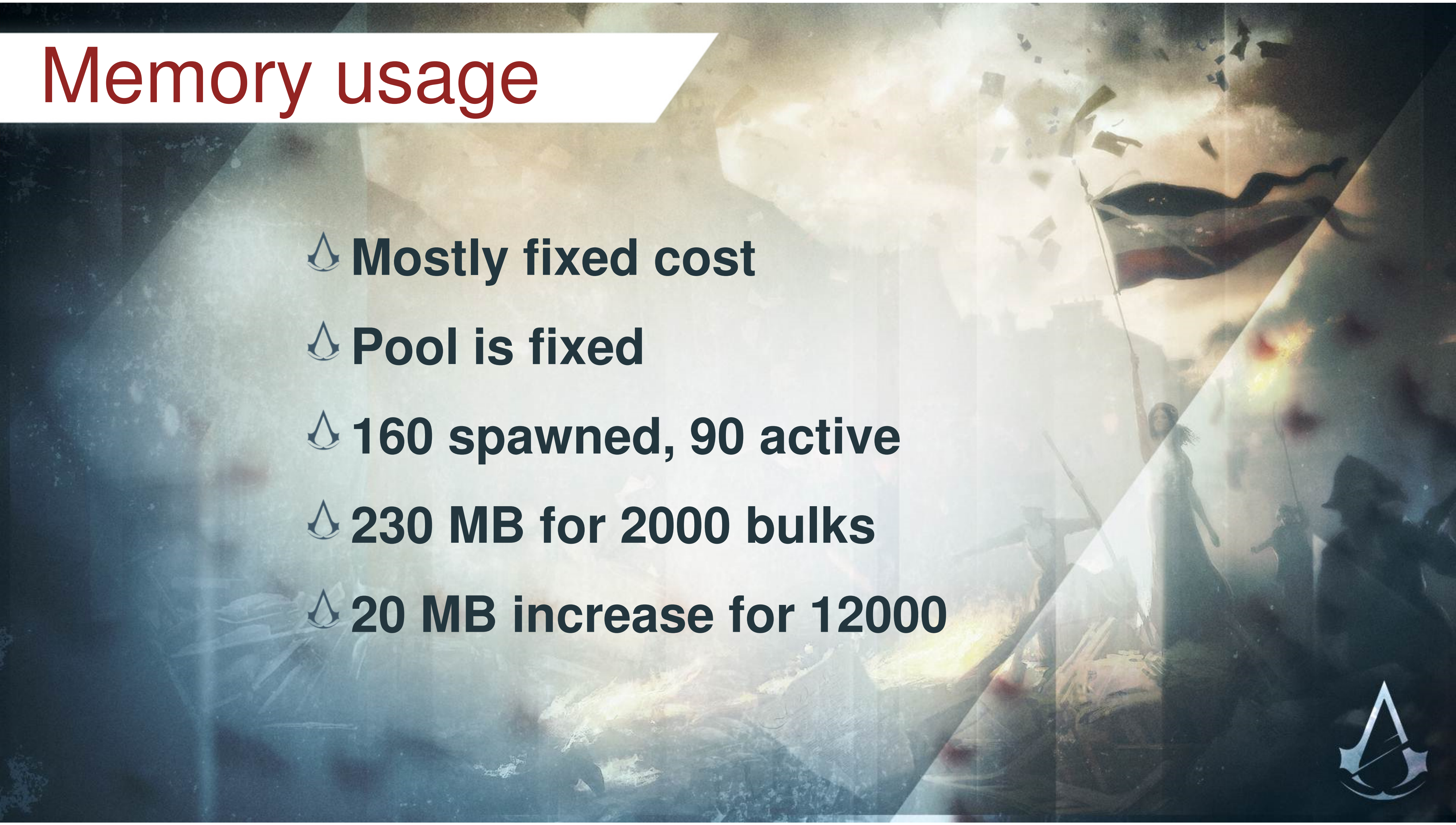
Typical Crowd CPU Distribution : 2000 NPCs



Huge Crowd CPU Distribution : 10000 NPCs

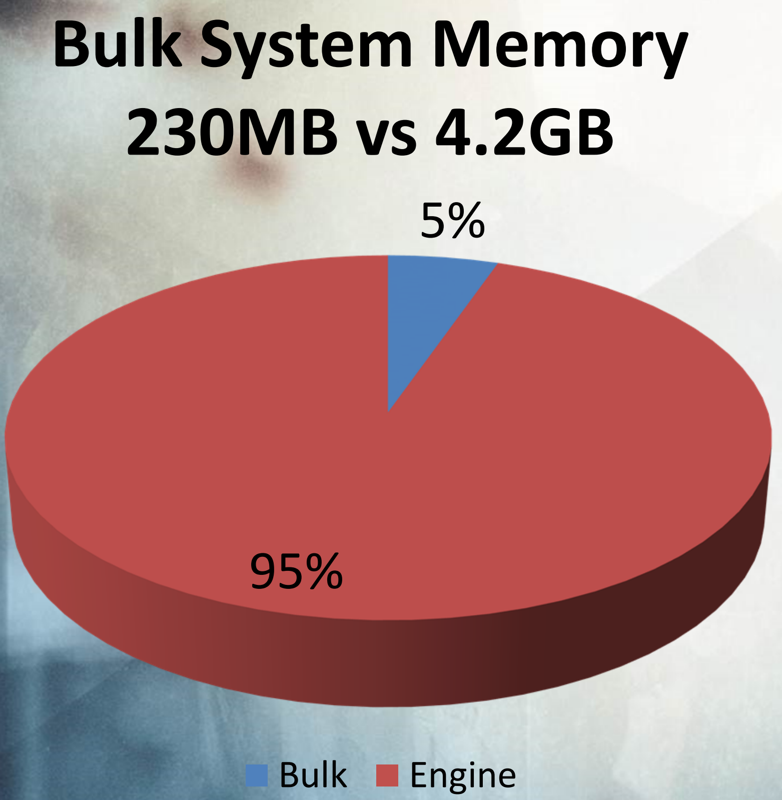


Memory usage

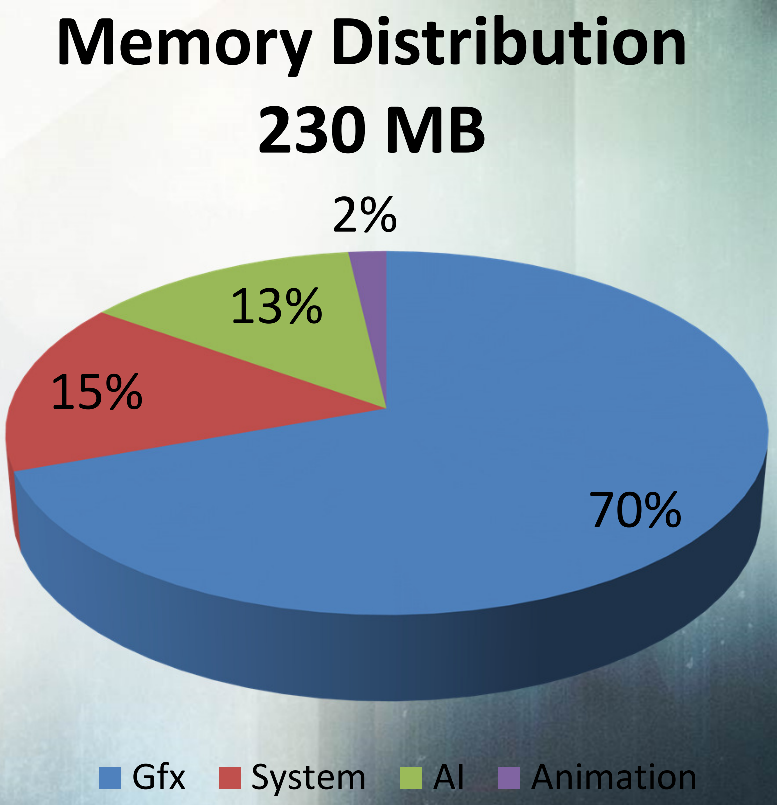


- Mostly fixed cost
- Pool is fixed
- 160 spawned, 90 active
- 230 MB for 2000 bulks
- 20 MB increase for 12000

Memory overview



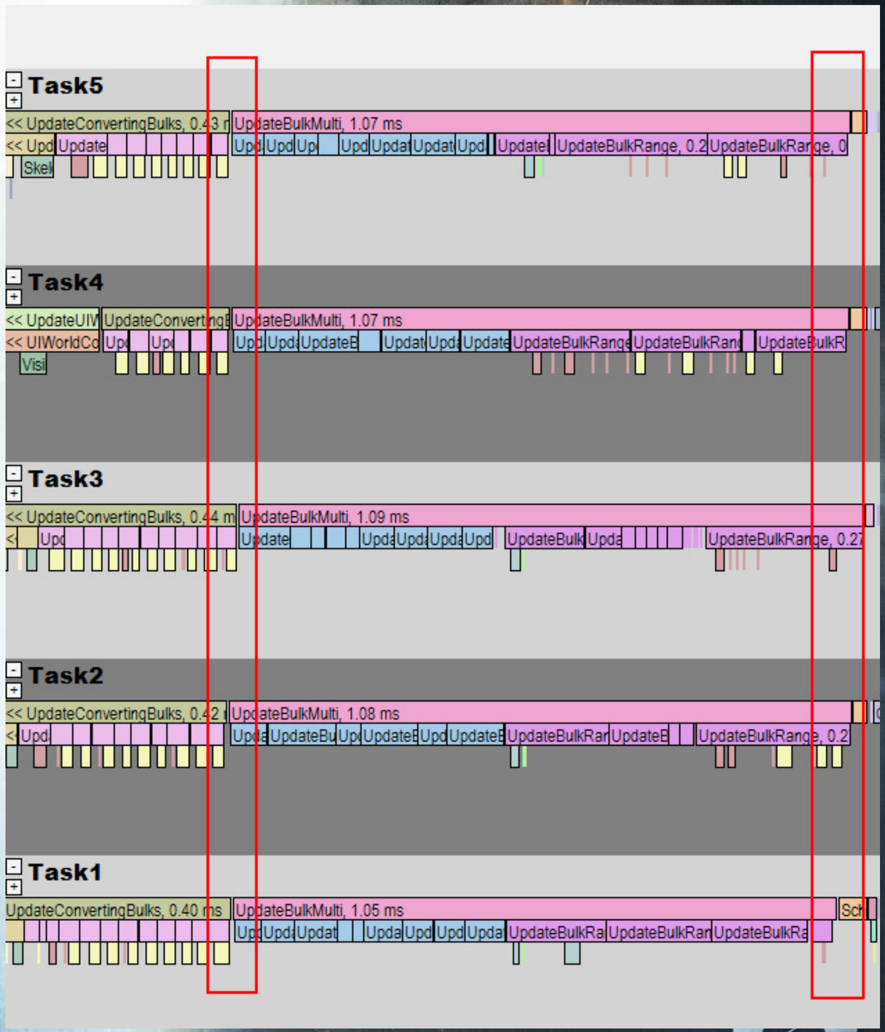
Bulk System Memory
230MB vs 4.2GB



Memory Distribution
230 MB

Multithreading

- Good profiling tool
- Good task scheduling
- Remove all CPU idle
- Lockless coding
- Limit lock times

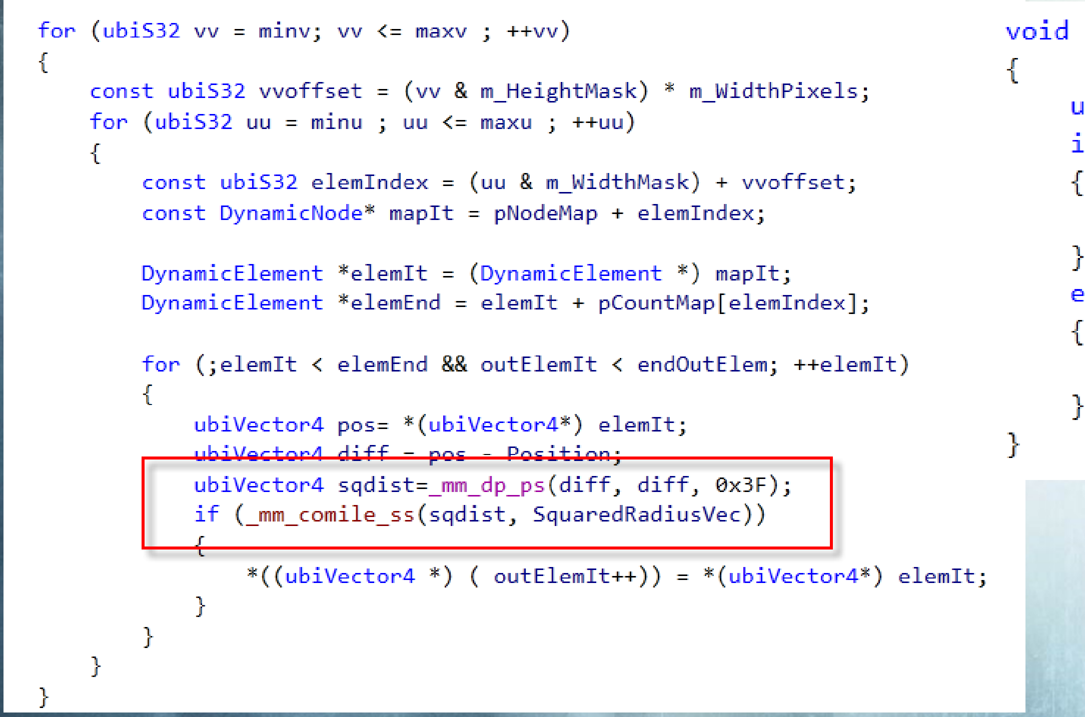


Dynamic Map 2d

- Spatially repeating
- Keep two maps, double buffered
- Lockless insertion
- No remove
- 1.5 us for a query of 1.5m radius

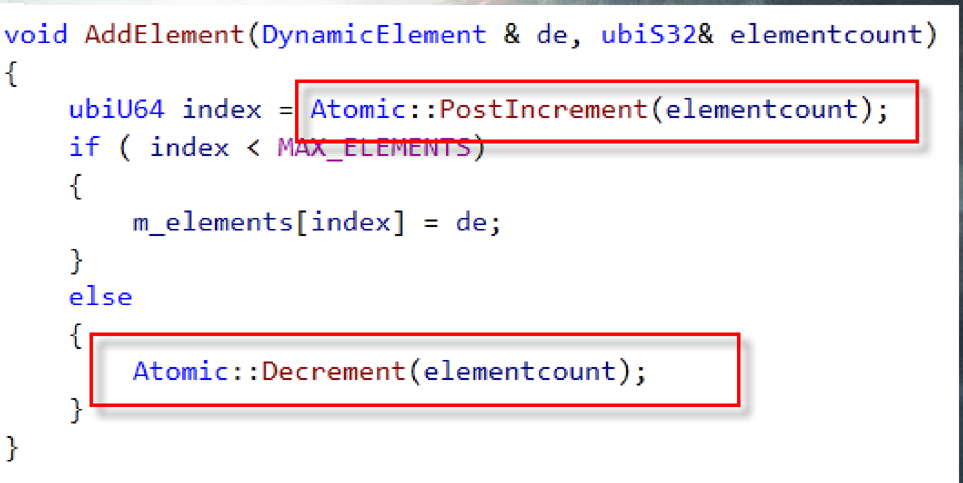
Code snippets

Fast query tight loop



```
for (ubiS32 vv = minv; vv <= maxv ; ++vv)
{
    const ubiS32 vvoffset = (vv & m_HeightMask) * m_WidthPixels;
    for (ubiS32 uu = minu ; uu <= maxu ; ++uu)
    {
        const ubiS32 elemIndex = (uu & m_WidthMask) + vvoffset;
        const DynamicNode* mapIt = pNodeMap + elemIndex;

        DynamicElement *elemIt = (DynamicElement *) mapIt;
        DynamicElement *elemEnd = elemIt + pCountMap[elemIndex];

        for (;elemIt < elemEnd && outElemIt < endOutElem; ++elemIt)
        {
            ubiVector4 pos= *(ubiVector4*) elemIt;
            ubiVector4 diff = pos - Position;
            ubiVector4 sqdist=_mm_dp_ps(diff, diff, 0x3F);
            if (_mm_comile_ss(sqdist, SquaredRadiusVec))
            {
                *((ubiVector4 *) ( outElemIt++)) = *(ubiVector4*) elemIt;
            }
        }
    }
}
```

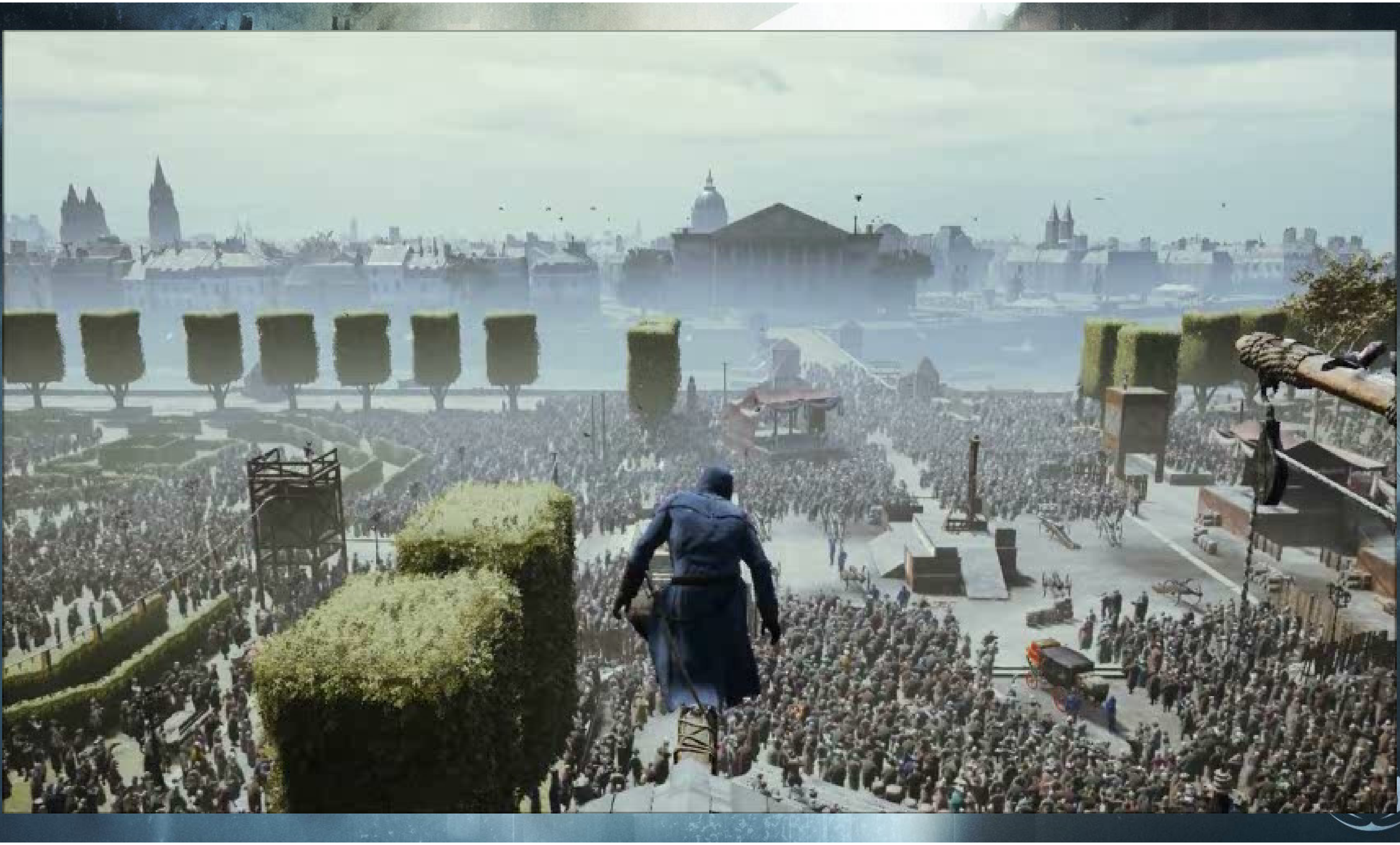
Add element



```
void AddElement(DynamicElement & de, ubiS32& elementcount)
{
    ubiU64 index = Atomic::PostIncrement(elementcount);
    if ( index < MAX_ELEMENTS)
    {
        m_elements[index] = de;
    }
    else
    {
        Atomic::Decrement(elementcount);
    }
}
```

Future work

- Dithering
- Unify reaction systems
- Deterministic reactions
- Better low res reactions
- Armies ?



Questions ?

Special thanks

Marcal Argudo Grau, Anthony Biancardi, Oliver Bodwell, Félix Duchesneau, Richard Dumas, Paul Ivorra, Amit Karim, Dominique Lachance, Aleissia Laidacker, Charles Lefebvre, Maxime Mercier, Cyril Meynier, Jeff Preshing, Tiago Rodrigues, Jose-Manuel Salgado, Ian Morai, Yoan St-Amant, Jean-François St-Amour, Frederic St-Onge, James Therien