

Pathing: Nav Meshes

Matthew Guzdial

guzdial@ualberta.ca



Announcements

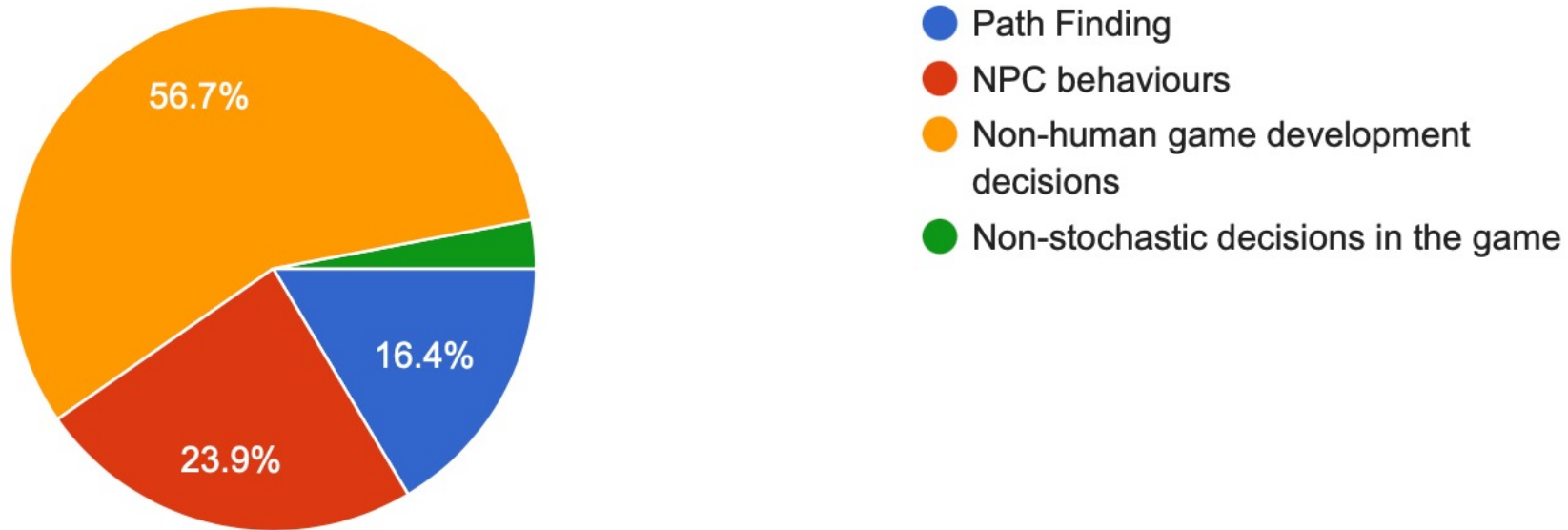
- TA office hours
- HW1 due Friday 11:55pm (grace period for 23 hours).
- Extra helper video on HW1 on eClass tonight.
- Lab on Thursday 5-7:50pm (ETLC E2-002) a
- Another practice quiz Friday, first real quiz **September 24**

Last Class

- Introduction to greedy path finding
- Introduction to path networks
- Pathing on path networks
- Discussion of pros and cons of path networks
- Practice Quiz
- (Only 80% of students took part in the participation questions)

Practice Quiz

(If this was marked it'd have a class average of ~70%)



A 4x4 grid with numbers 1 through 16. Blue circles of varying sizes are placed over the grid cells. The circles are centered on cells 4, 8, 10, 13, and 16. The circle over cell 4 is the largest, followed by cell 8, then cell 16, cell 13, and cell 10 is the smallest.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

- Don't forget to give a why if I ask for why (or no points)
- Some of you still thought 7 would be okay
- Final question about generating a grid with 2 different grid sizes? "change grid size" and similar are not answers.

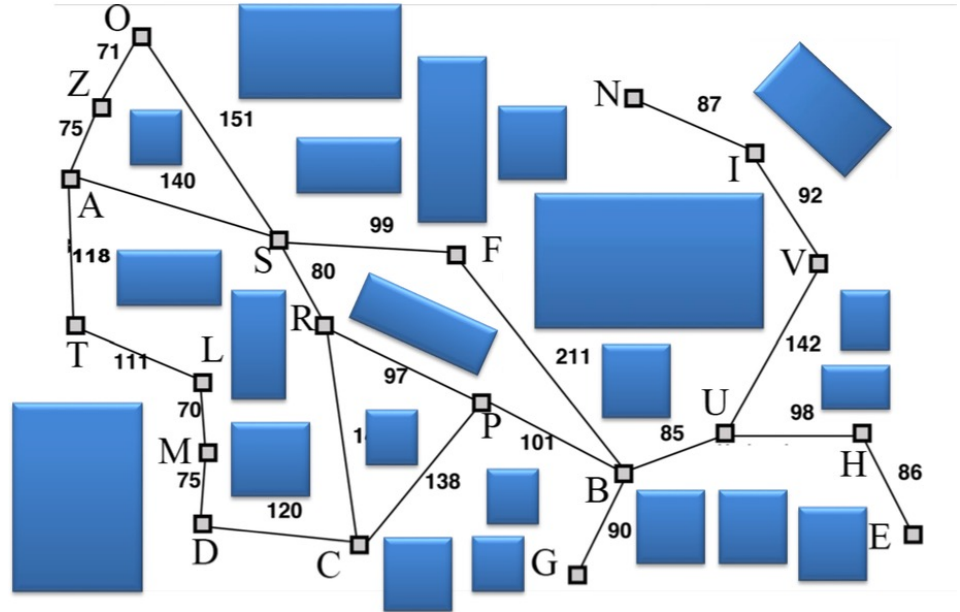
Grid Pros

- Simplified Space
 - Know you are “safe” in a grid point
- Natural fit for pathing algorithms
- Simple general algorithm for any map



Path Network Pros

- Faster pathing time
- More natural movement
- Allows for some continuous movement “off-network”

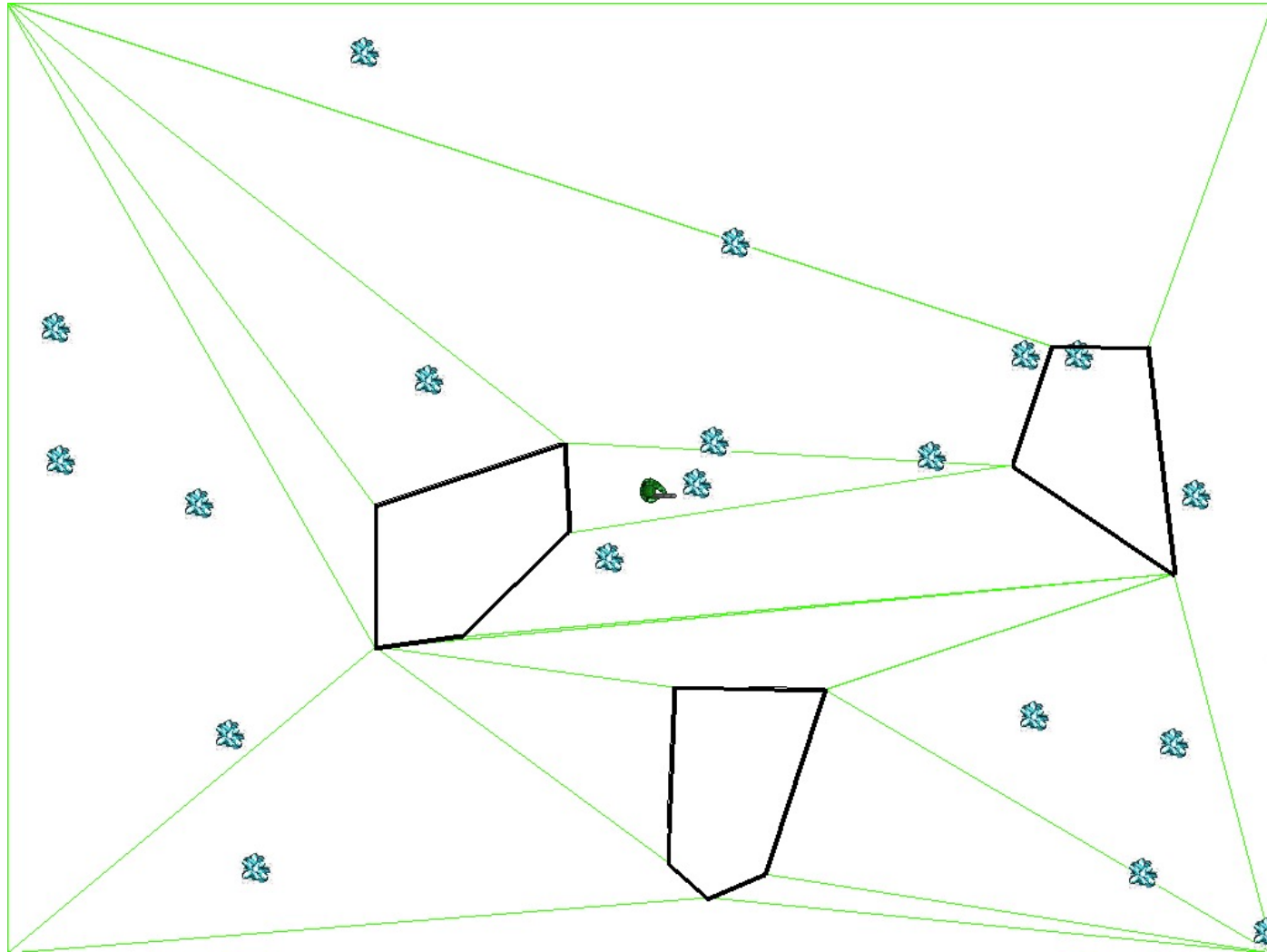


Grid and Path Network Shared Cons

- Movement is restricted
 - Center of grid points
 - Lines of the path network
- Default pathing looks awkward/jagged
 - Humans don't typically stop at a series of waypoints in trying to get somewhere
 - We would prefer to not run path smoothing if we can, since it's an additional time cost (Reminder: low computation % for Game AI).

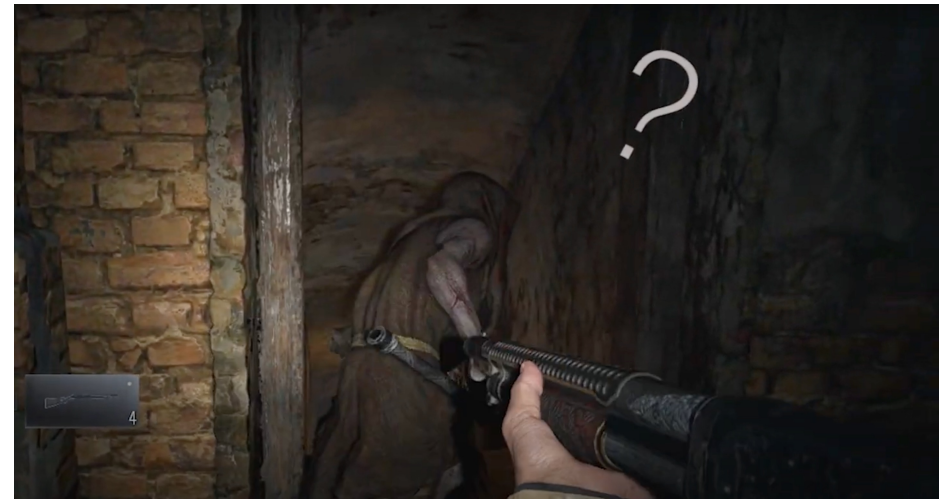
Solution: Nav Mesh

A collection of polygons (mesh) with shared edges that designate areas of (largely) free movement.





Sonic Boom: Rise of Lyric



Resident Evil 8

<https://youtu.be/SC6lZz4LLkl>



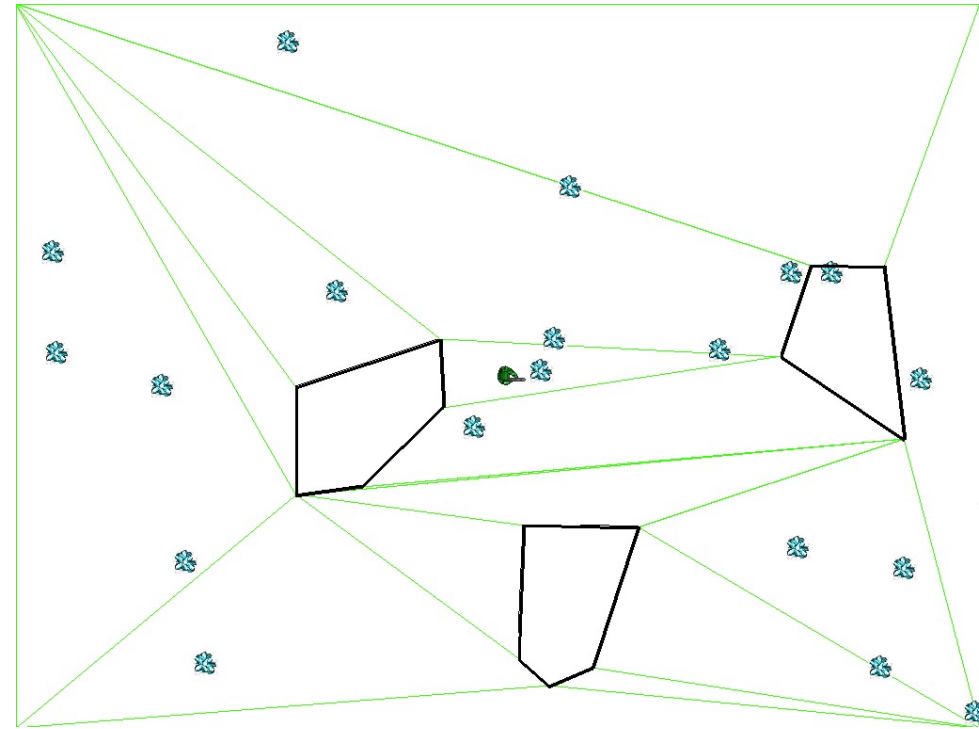
Ratchet & Clank: Rift Apart



Skyrim

NavMesh Pros

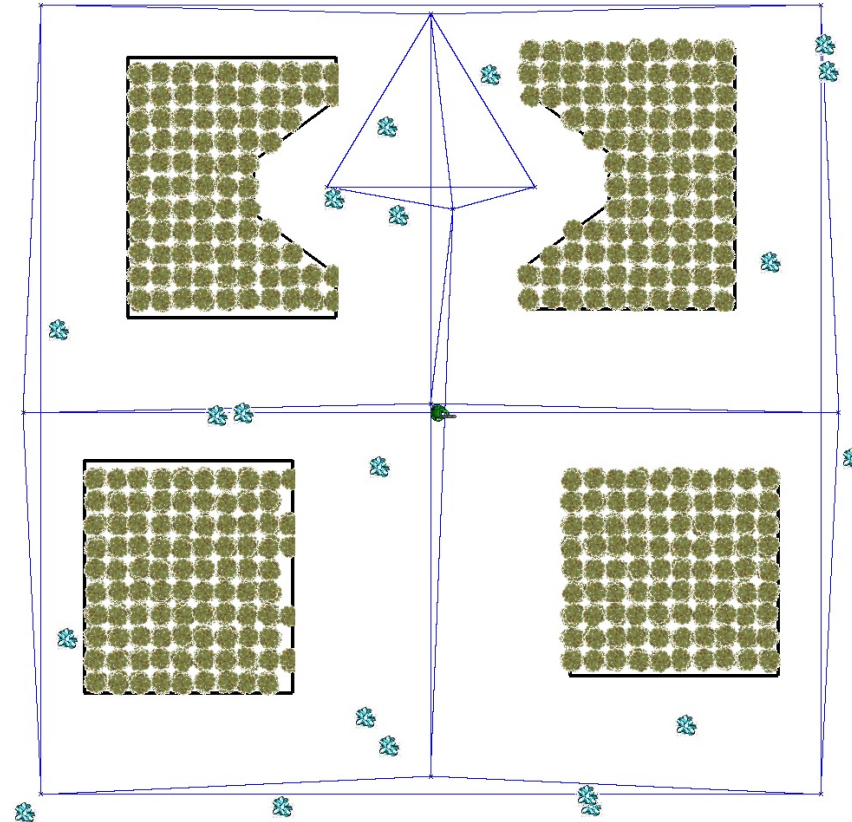
- Movement not restricted within the mesh
- Only need to path in between individual sections of the mesh
- Robust to 3D movement.
- Typically generated than hand-tweaked.



Generating the Mesh

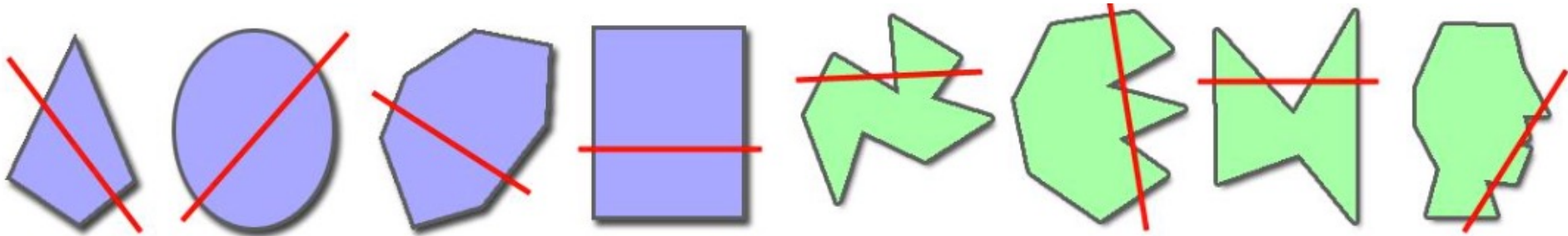
- Goal: Find the minimum number of nodes/polygons/list of edges to cover all free space in the map

Negative Example ->



NavMesh Generation

- Each node of the mesh is a list of edges
 - These edges must define a *convex polygon*
 - Convex Polygon: Any point within the polygon is unobstructed from any other



Generating the Mesh: Greedy/Simple Approach

For point a in world points: <- All obstacle points and the edges of the area to be mapped.

For point b in world points:

For point c in world points:

if (it is a valid triangle) and !exists:

add triangle to mesh

Iterate through triangles to merge to quads

Iterate through quads to merge to 5-sided shapes...

How can we speed up the simple/greedy method to generate a nav mesh?

Speeding up Nav Meshing

Lots of ways!

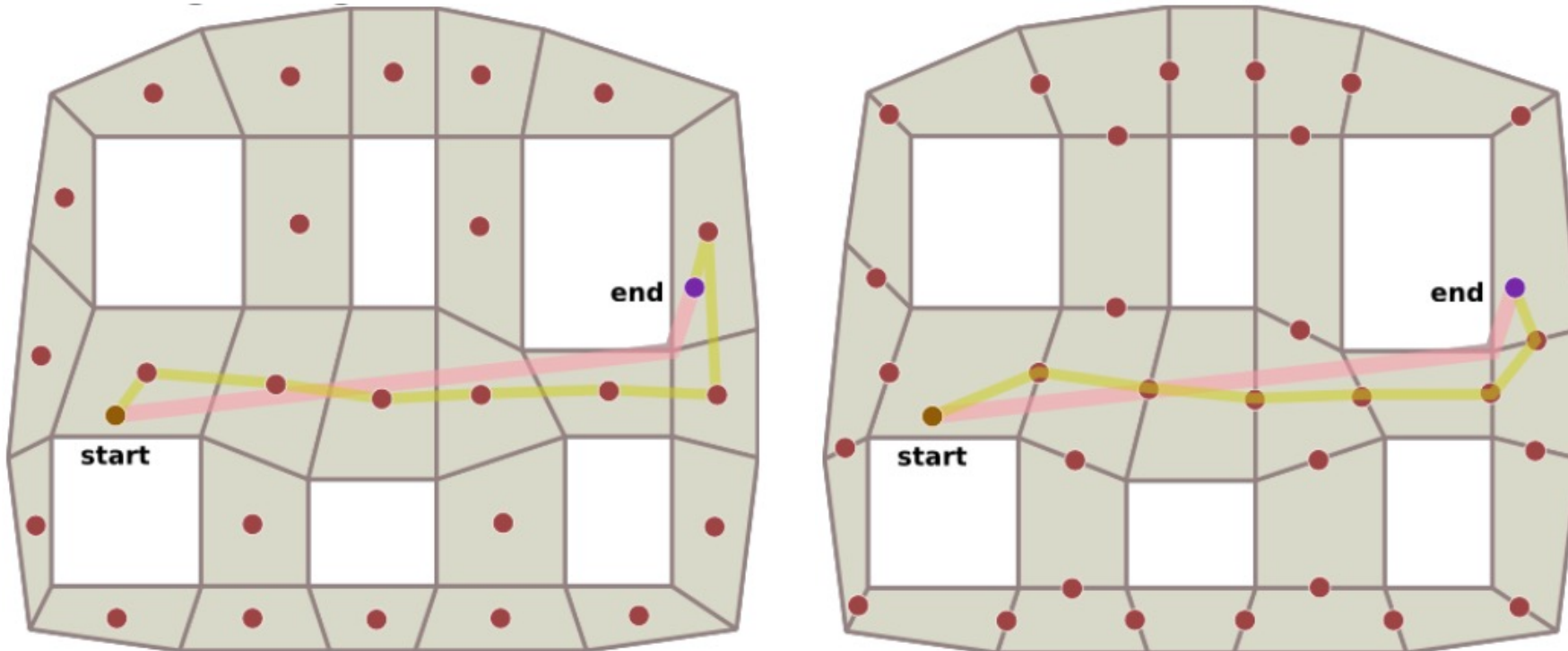
We could start with the whole world as one mesh, then reduce till its valid.

We could use a flood fill style approach, and grow a mesh off some starting point to avoid redundantly iterating over world points.

Etc.

Navigating Between Polygons

To path between polygons of the nav mesh we need to know what point we can safely path between in/on each polygon (**Path Network!**)



Nav Mesh + Path Network

- Nav Mesh: defines a set of continual free-movement spaces, like variably sized grids
- Path Network: Allows us to move between these spaces when needed

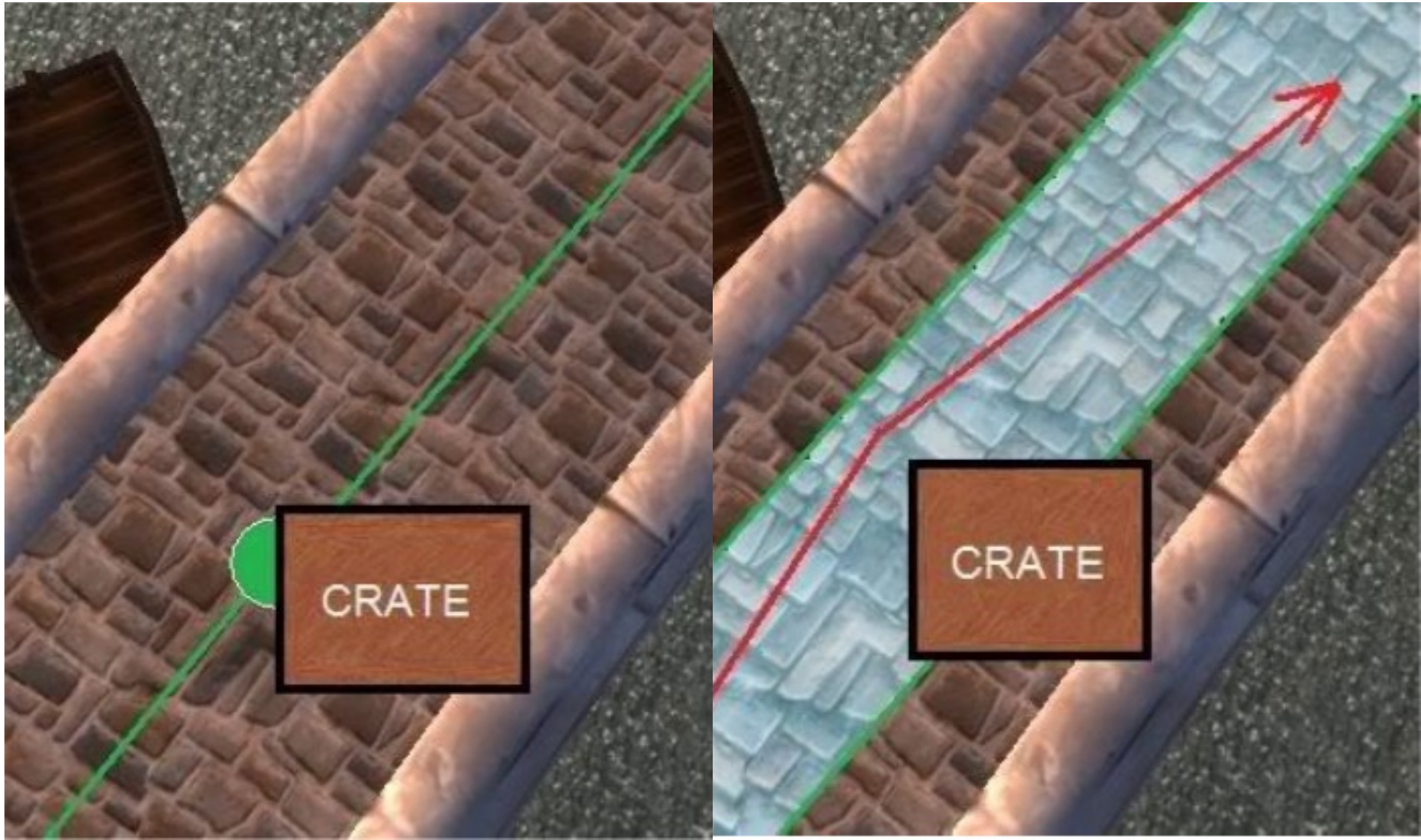
Need significantly more waypoints to match the
freedom of nav mesh



Paths appear less jagged



Allow for path correction: Alterable/Generated Content



Nav Meshes allow for many agents

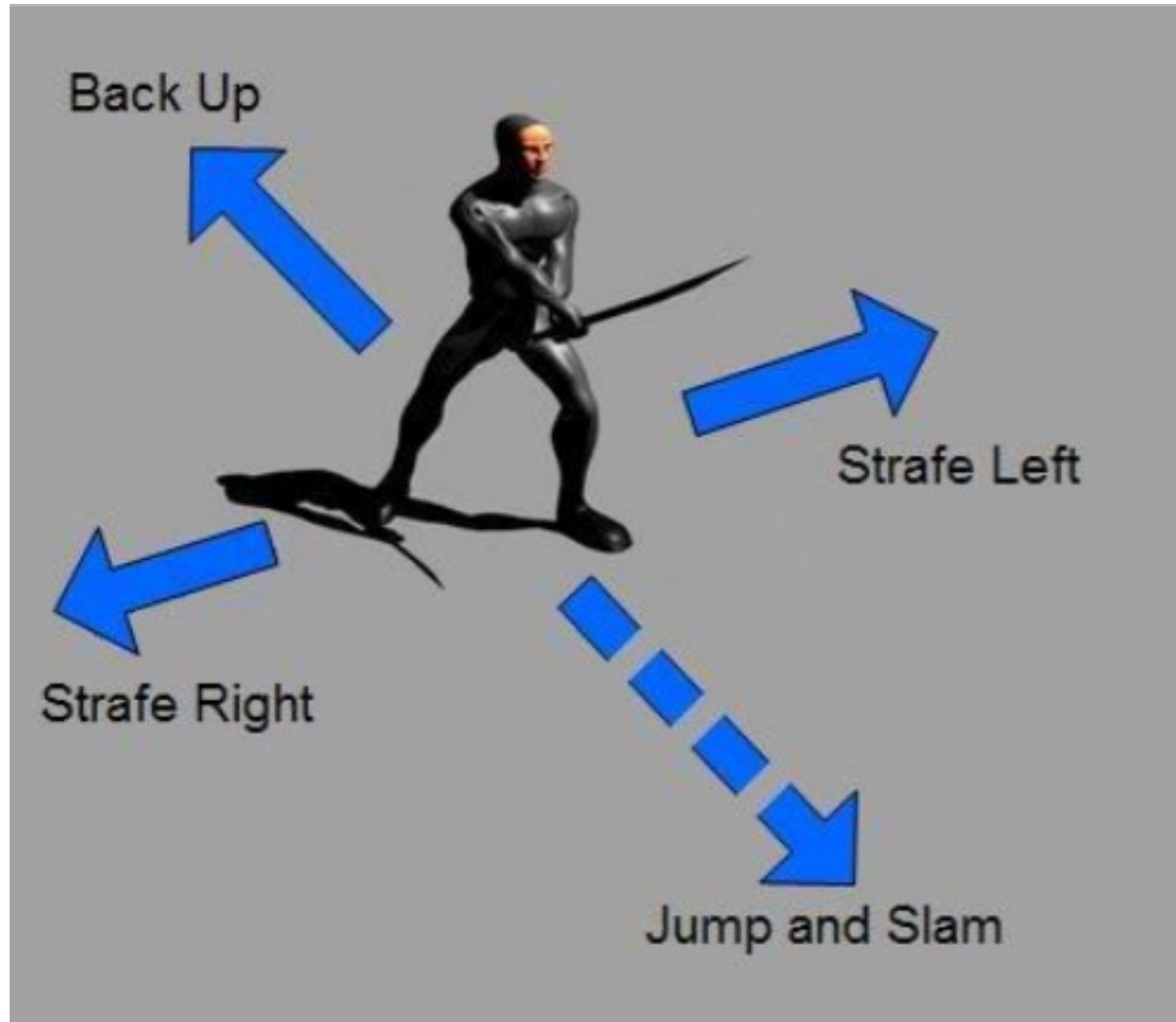


Hyrule Warriors: Age of Calamity



Starcraft 2

Nav Meshes allow for strategic movement



Designers Can Still Add Info



PQ 1: Memory Usage

<https://forms.gle/piH5WFqDmMcT2e827>

<https://tinyurl.com/guz-pq4>

Rank these four space representations according to the on disc and on runtime memory they would use for the same simple scene (empty space and obstacles):

1. Grid
2. Path network (designed)
3. Nav Mesh (generated)
4. Nav Mesh (tweaked)

PQ 1: Answer

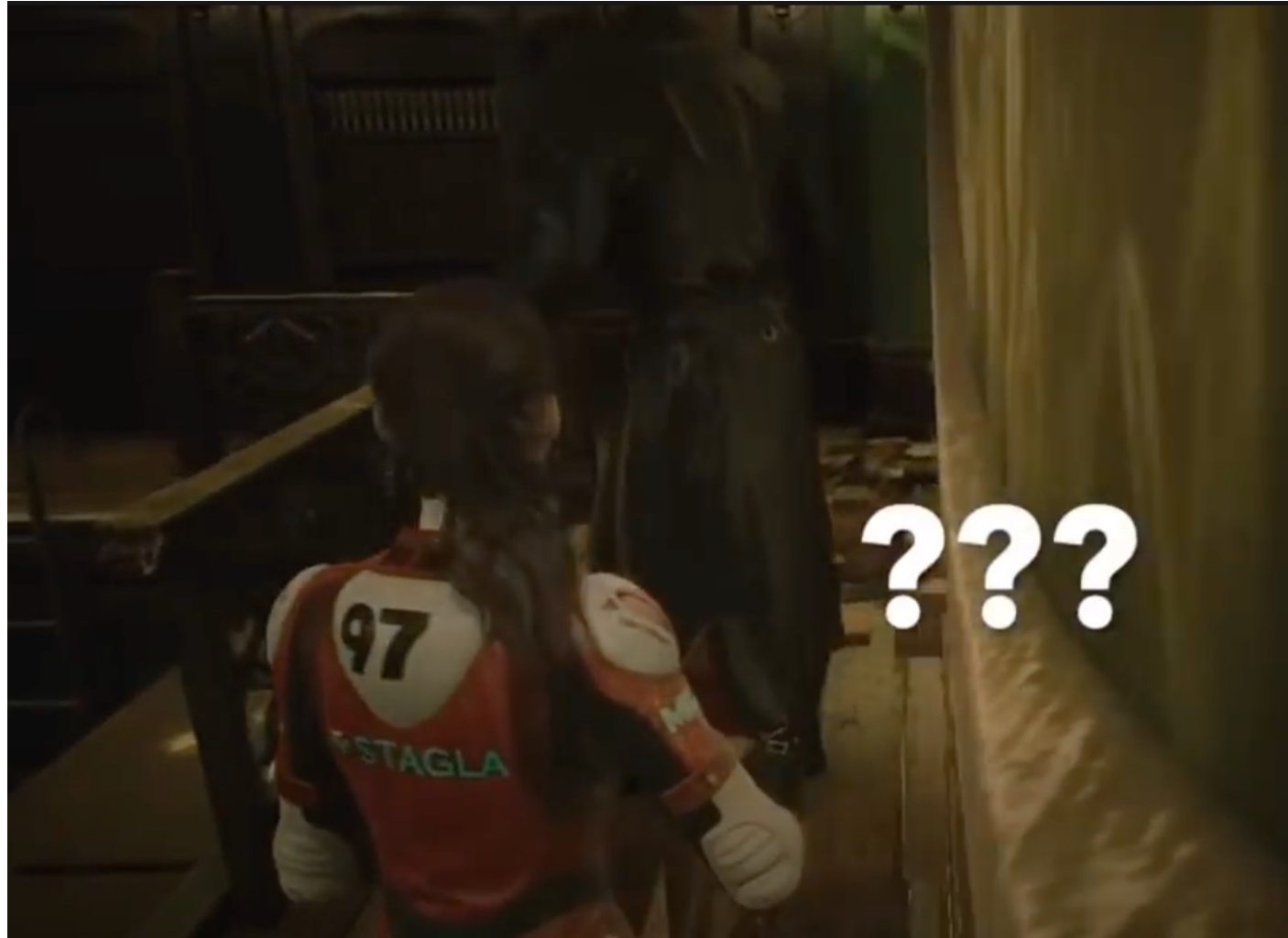
On Disc (3-4 roughly equivalent)

1. Path network (designed)
2. Nav Mesh (tweaked)
3. Nav Mesh (generated)
4. Grid

Runtime (2-4 might switch depending on context and environment)

1. Grid
2. Nav Mesh (generated)
3. Nav Mesh (tweaked)
4. Path Networks (designed)

Even with all of this, Nav Meshes aren't perfect.



<https://www.youtube.com/watch?v=NixiSQitPY8>

Game design can cover for Game AI

- AI agents don't live long enough to let you see flaws in their pathfinding
 - Seek the player to force this
- AI agents don't move much or *mysteriously* teleport
 - <https://youtu.be/13YIEPwOfmk>
https://youtu.be/2exHZm_6lg8
- AI agents can “give up” and return to a safe default
 - <https://www.youtube.com/watch?v=gXjUzHhNjIA>



Halo (Elites)



Breath of the Wild (Old Man)

Summarize what we learned

- Grids and Path Networks
- Greedy Pathing
- Nav Mesh
- Nav Mesh Generation
- Nav Mesh + Path networks

Next time...

- Path smoothing
- Steering
- Moving multiple agents



Any Extra Time Discuss Extra Credit