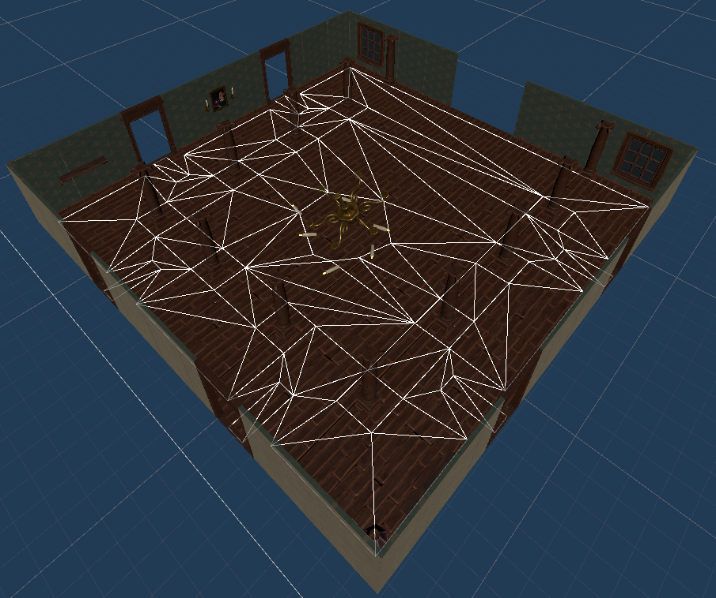
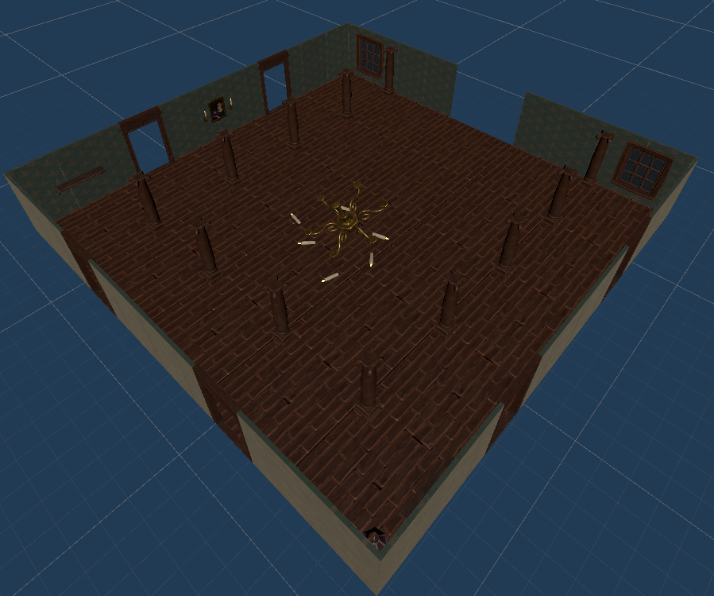
Pathfinding documentation

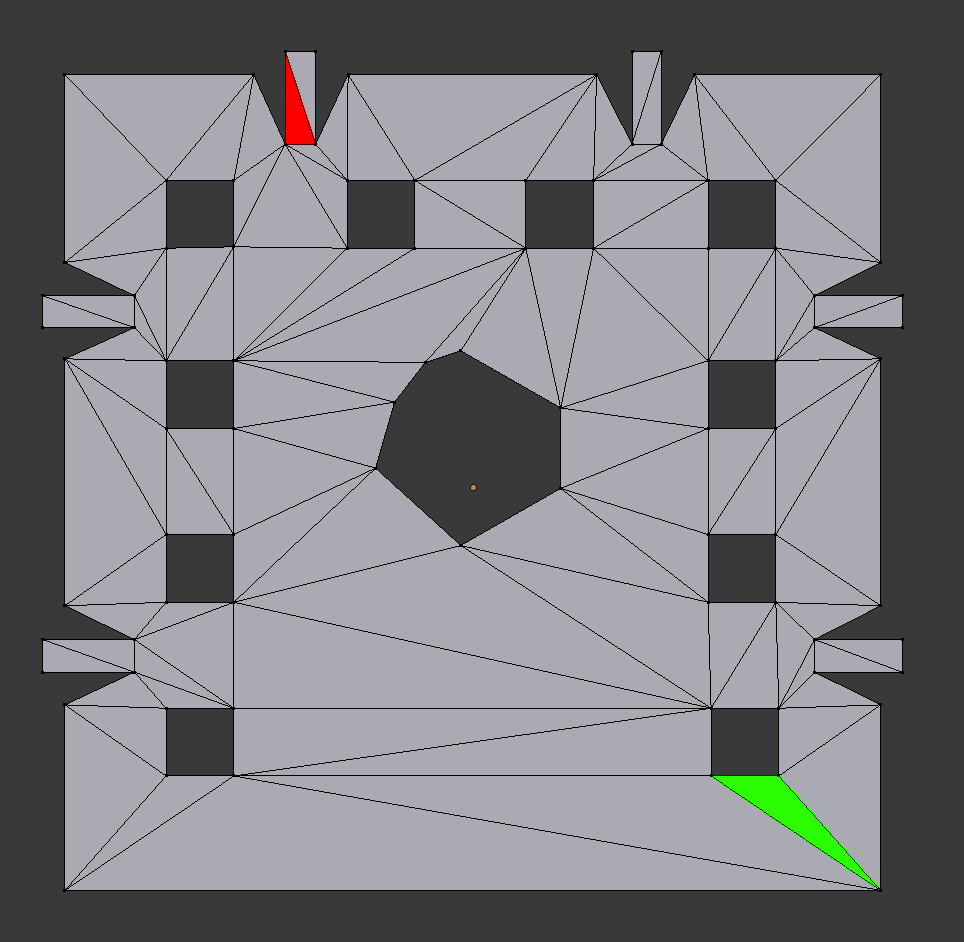
(To be expanded)

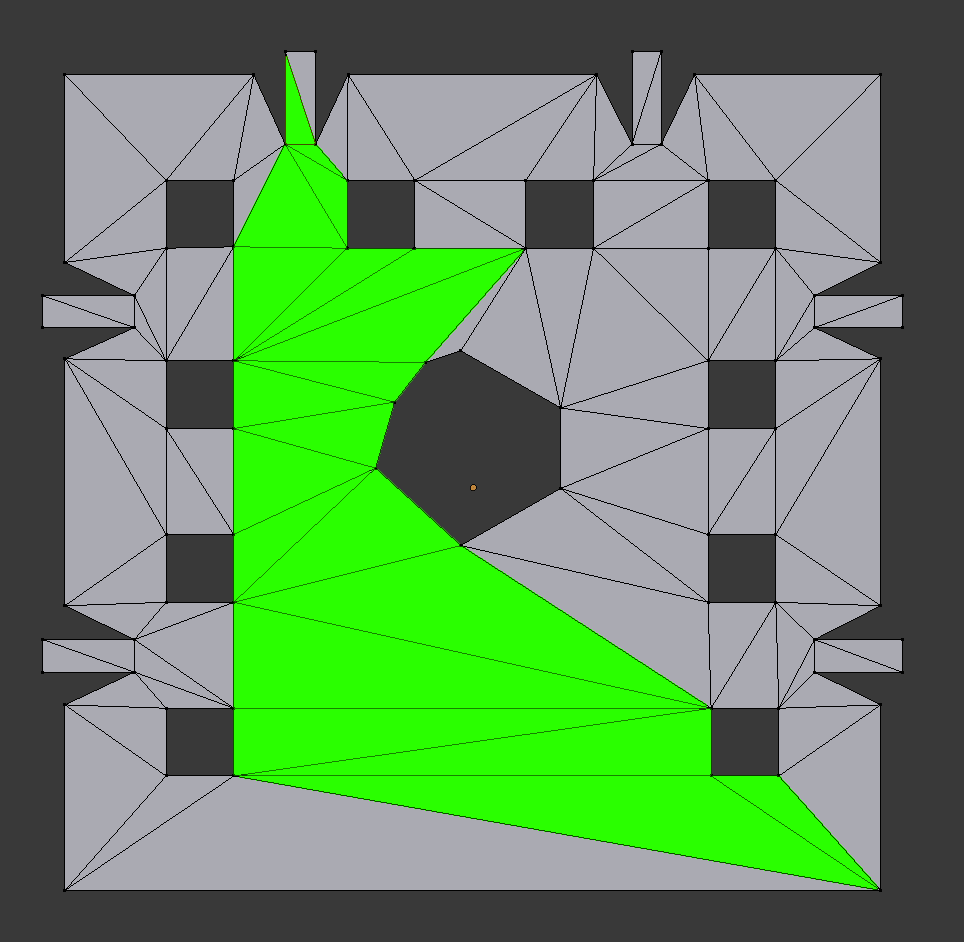
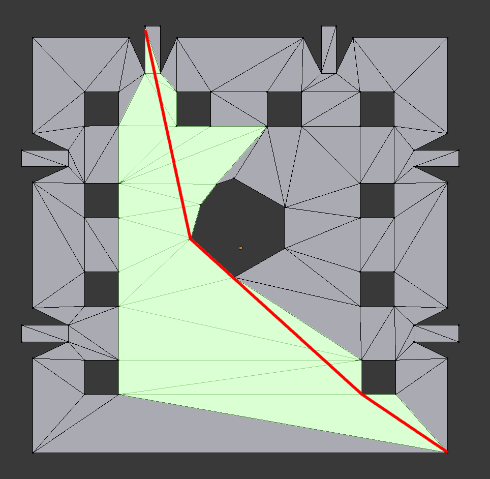
# Theory

The pathfinder uses navmeshes for navigating, meaning a simplified mesh that defines a walkable area. Navmeshes can be created either manually or by any external generator as long as said generator outputs a mesh.

Above is an example of a room with and without

## Pathfinder steps



1. The pathfinder defines the start and end faces
2. The pathfinder builds a path of faces using an A\* algorithm.
3. The pathfinder runs a string-pull algorithm on the selected faces, which returns the shortest path from start to end within the area.

## Obstacle avoidance

The pathfinder has two systems for avoiding obstacles not included in the navmesh.

# Inspector settings

## Pathfinder component

# Movement settings

### Use root motion

Whether the agent should move through root motion from the animator component or if the component is to move the agent itself.

### Movement speed

How fast the agent should move. Does nothing if “*Use root motion”* is enabled.

### Rotation speed

How fast the agent can turn.

### Leg height

The desired distance between the ground and the agent’s origin.

# Pathfinding settings

### Max recursion depth

How many times the stringpull algorithm can recurse before aborting. Mostly a safety feature.

### Transform raycast offset

The offset from the agents origin to where the component raycasts from. Mostly to avoid hitting the floor if the origin is low.

### Node arrival margin

How close the agent has to be to a point to consider it self at the point.

### Pathfind debug

When enabled, draws lines to show the current path the agent is following, magenta being the global path (path between rooms) and cyan being the local path (path within a room).

# Obstacle avoidance settings

### Raycast avoid distance

How far away the obstacle avoid rays should reach.

### Raycast avoid angle

The angle between the front and the left/right rays.

### Raycast avoid weight

How strongly the agent is to avoid objects hit by the raycasts.

### Minion avoid distance

How close another agent has to be for the agent to start steering away from it.

### Minion avoid weight

How strongly the agent is to avoid other nearby minions.

### Obstacle avoid debug

When enabled, draws the different forces acting on the agent.

Green: Sum of all forces

Yellow: Path following force

Red: Agent avoid force

Cyan: Raycast avoid force

# Unstuck settings

### Min velocity threshold

How slow the agent has to move to be considered stationary (stuck).

### Stuck time threshold

How long the agent has to be “stationary” to start trying to unstuck itself.

### Unstuck radius

How far the agent can move to unstuck itself.

# Miscellaneous settings

### Door interact range

How far in front of the agent to start looking for doors to open.

### Debug

Enables/disables debug info.