

# ARTIFICIAL INTELLIGENCE

## Enemy AI

We decided that each type of enemy should have a different behavior of it's own.

**Grunts** follow one of the set paths on the level. They choose a path at the beginning of the wave and stick to it.

**Archers** keep their distance. They like to be an annoyance and cause the player to be distracted from the other enemies. They will flee from the player if pursued.

**Mages** also keep their distance. They also like to distract the player by shooting fireballs at the player, dealing a lot of damage. They stand their ground when pursued.

**Warriors** will chase and attack the player on sight. They will fight until they are killed.

**The King** will also attack the player on sight, but he doesn't show up until the last wave. He will not stray from his castle though.



## Tower AI

The towers in our game have the ability to track and shoot enemies based on a few different parameters. They can shoot:

The first enemy

The last enemy

The strongest enemy

The weakest enemy

A specific type of enemy



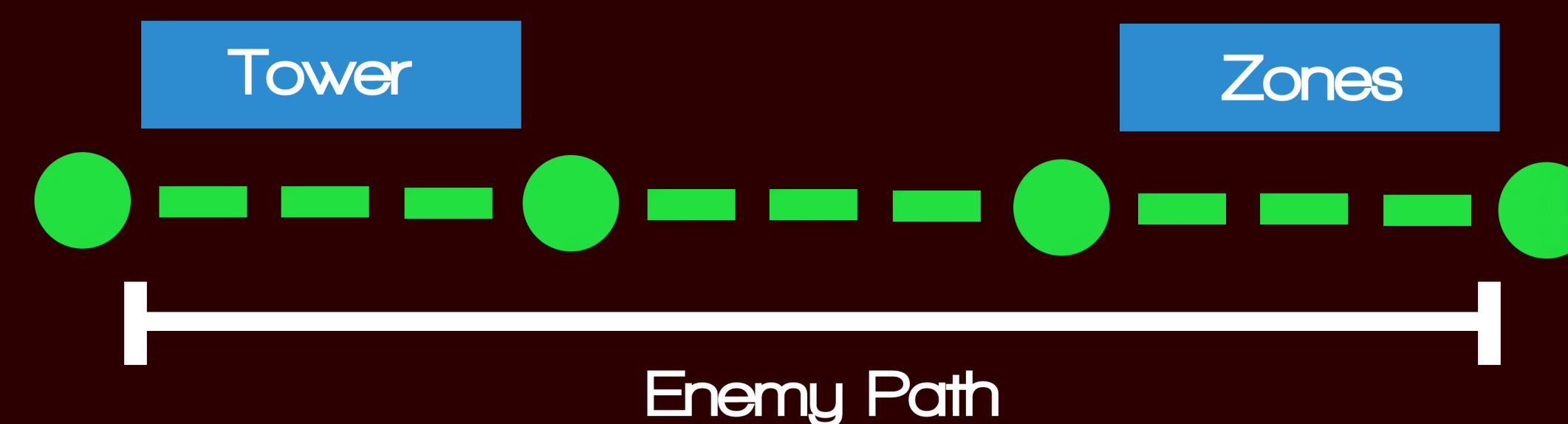
# INFINITE LEGIONS

## TECHNICAL

## LEVEL DESIGN

Every one of the levels in Infinite Legions was hand crafted by one of the developers. We did this because paths needed to be strategically placed throughout the level, and that was something that procedural generation could not handle. A few other factors also led to us hand crafting the levels.

The towers in our game have to be built in placement zones. These zones had to be strategically placed around the map as to not hinder the player with not enough of them, but not too many to give the player too much of an advantage.



Another reason that we chose to hand craft the levels was to add in hidden paths to trick the player. It is another level of strategy to keep the player on their toes.

The last reason we chose to hand craft the levels was so that we could add in fun scenery and other object that wouldn't be added by using procedurally generated content.



## DEVELOPERS

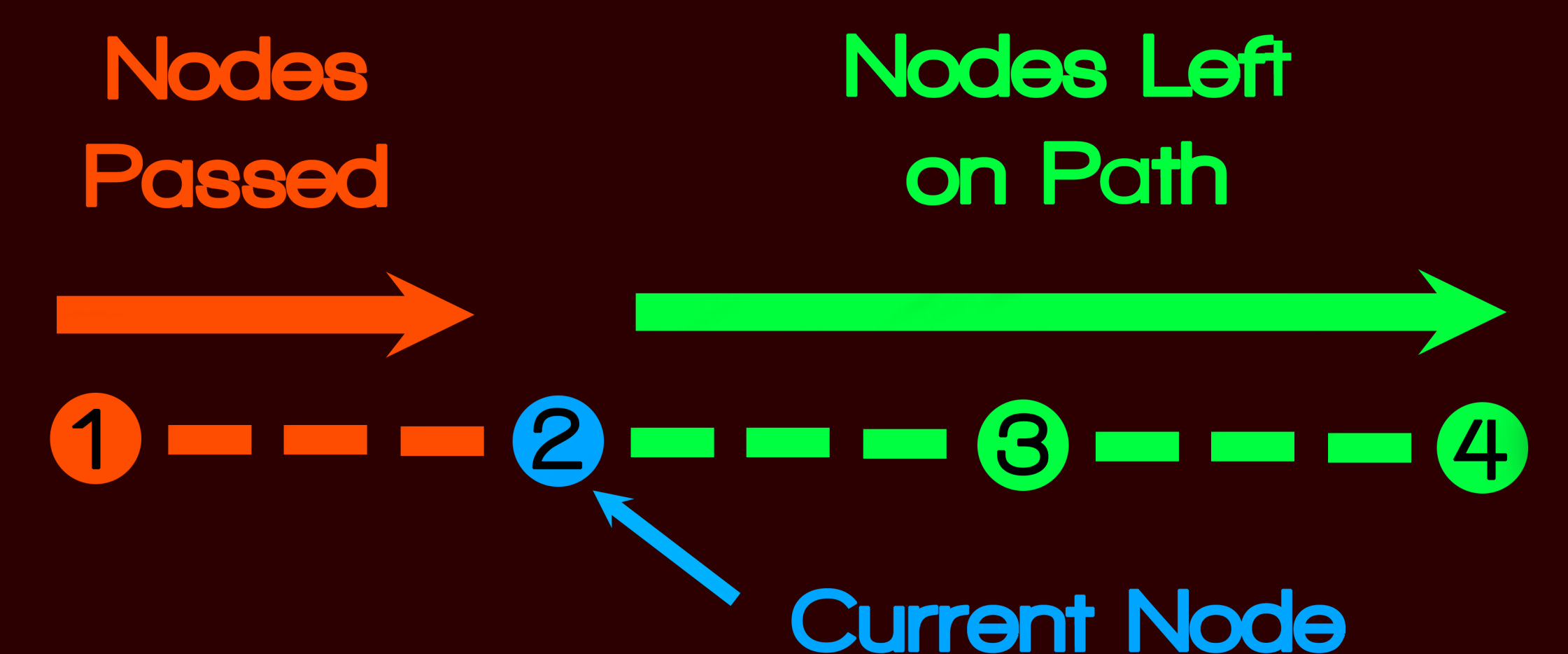
- Ben Bricker - Bricker.96@osu.edu
- Mitch Cooley - Cooley.198@osu.edu
- Garrett Davis - Davis.2570@osu.edu
- Rocky Forehand - Forehand.9@osu.edu
- Tom Wallace - Wallace.545@osu.edu

# NAVIGATION

To get our minions and players to correctly navigate the environment, we used a feature built into Unity called a NavMesh.

## Enemy Navigation

The enemies in our game walk along a path as they traverse the map. In order to navigate this path, they go from path node to path node until they reach the end of the path. We tell them which path node to go to and Unity's NavMesh tells them how to get to that node.



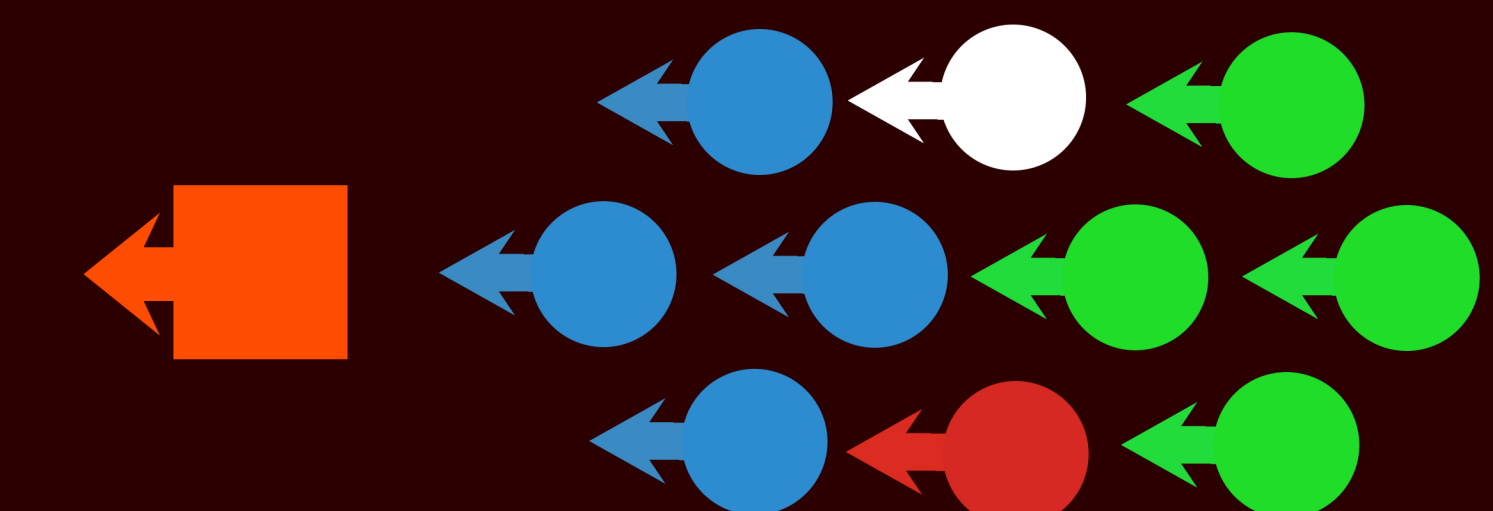
## Minion Navigation

The minions in our game also follow points on the map to decide where they are going. The difference with them is that they follow points that are around the player. These points can be changed around to put the minions into different formations. As with the enemies, the points tell the minions where to go while Unity's NavMesh tells the minions how to get there.

Archer - Knight - Player - Priest - Mage



## Follow Formation



THE OHIO STATE UNIVERSITY

INFINITE LOOP GAMES