CS354P

DR SARAH ABRAHAM

OVERVIEW: AI

ARTIFICIAL INTELLIGENCE (AI)

- Broad category of non-player-controller agents in the game that follow some set of rules
  - Bots in an arena shooter
  - Other cars in a racing game
  - Ally NPCs in an open world game
  - Enemies in an RPG
  - Computer-controlled opponents in board and card games

A FINE LINE...

- The distinction between a scripted event and an AI system is a fine one
  - Very primitive AI can be hand-scripted
  - Limitations in technology make earlier AI much more simplistic than modern AI

Metal Gear

Metal Gear Solid V

THEN WHAT MAKES IT INTELLIGENT?

- Awareness of the state of the world
  - Understands concepts such as terrain, player state, and own current state
- Ability to react in a way that reflects this understanding
  - Reacts in a way that the player can interpret as intelligent

Detroit: Become Human

AI REQUIRES AT LEAST SOME COMPLEXITY OF BEHAVIORS

THREE STAGES OF AI LOOP

- Sensing
  - Taking in information about the world state
- Thinking
  - Determining best course of action based on the world state
  - Note that “best” is not necessarily mathematically optimal even in cases where we can calculate optimal
- Acting
  - Performing the necessary steps to complete the chosen action
  - If action is not completable, agent may have to sense and think to determine new best action

AI SENSING

- Agent retrieves information from the world
  - When to retrieve?
  - What to retrieve?
  - How to retrieve?

AI POLLING AND EVENTS

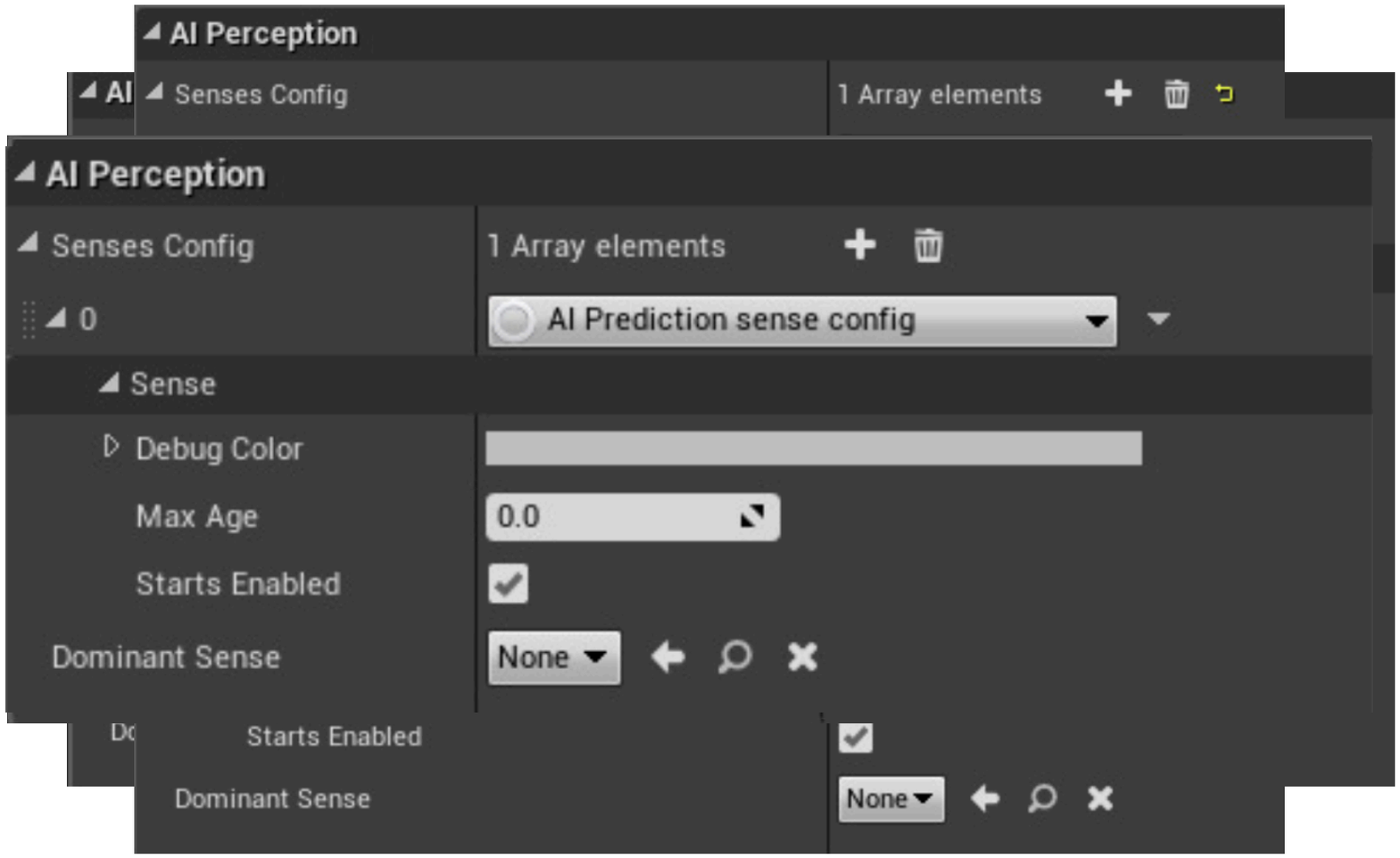
- In polling model, agent examines the world state at a fixed rate
  - Looks for changes in the world and updates working knowledge accordingly
- In event-driven model, agent receives information based on changes in the world state
  - Notified when a change in the world occurs and updates working knowledge accordingly
- Can use a combination of both as necessary

UNREAL: AI PERCEPTION COMPONENT

- Component that can be attached to a Pawn
- Defines:
  - What the sense to listen for
  - Sensor parameterization
  - How to respond
- Response handled through events

UNREAL: AI PERCEPTION COMPONENT SENSES

- Multiple senses can be added to a component
  - Choose subset based on project requirements
- Senses are:
  - Damage
  - Hearing
  - Sight
  - Touch
  - Team
  - Prediction



UNREAL: ENVIRONMENTAL QUERY SYSTEM (EQS)

- Experimental system for collecting data from the environment to inform agent decision-making
  - Generator nodes collect information about the world state
  - Test nodes define how to process that information
  - Results of tests inform agent how to react
- Example: Teammate AI looks for health pick-ups when the player's health is low while remaining out of enemy's line of sight

AI SENSING CONSIDERATIONS

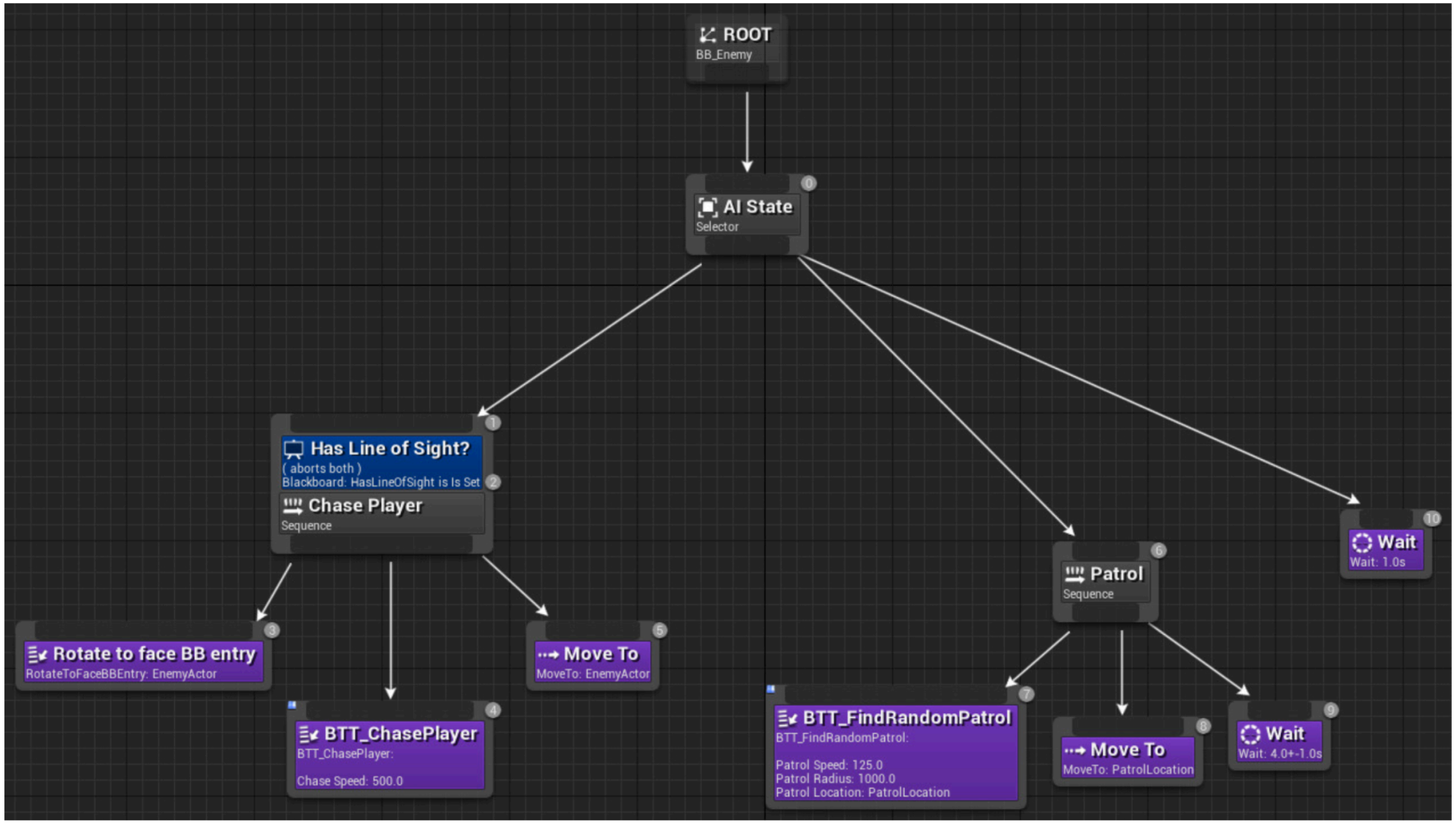
- The world state of even smallish games is very large
- Too expensive to poll frequently or query too much information
- Determine what is important for the agent to know to create a compelling experience
- Try to access this information only when necessary
  - e.g. When the player is on the other side of the map, turn off agents' sensors

AI THINKING

- Agent takes working knowledge about the world state and determines next action
  - What is the action?
  - Does this action require secondary actions?
  - How much time to process?
  - How to process?

BEHAVIOR TREES

- Current industry standard data structure for working with AI
  - Hierarchical tree that encapsulates all possible behaviors based on world and agent state



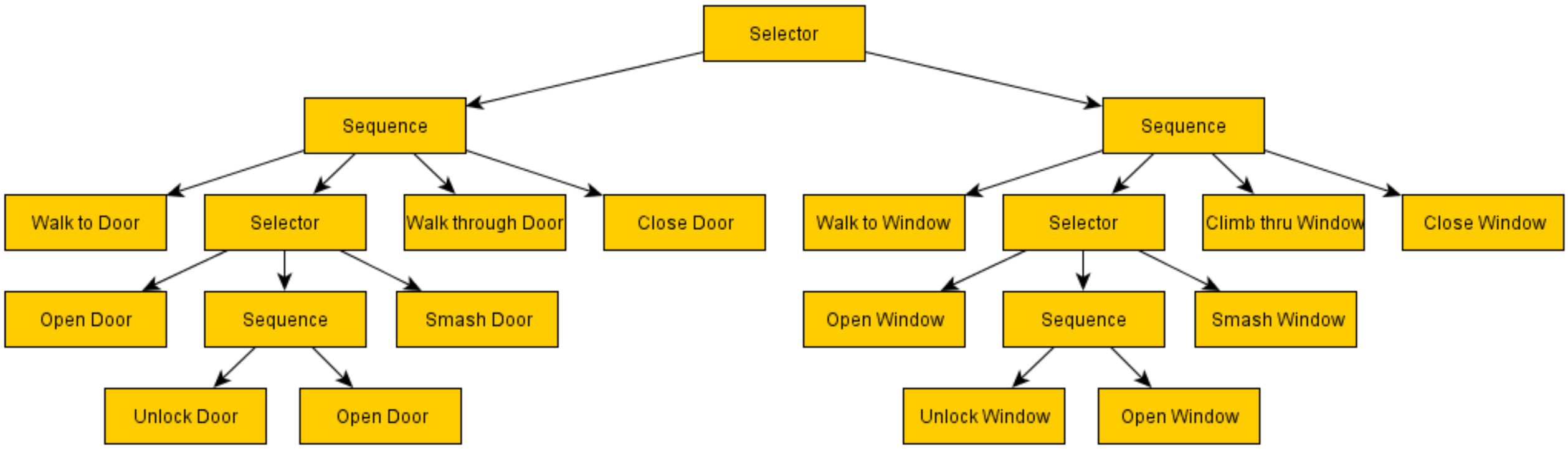
BEHAVIOR TREES NODES

- Nodes can be in the following states:
  - Succeeded, Failed, Running
- Parent node executes child nodes in a given order
  - While a child is being processed, it and its parent are in the Running state
  - When a child fails or succeeds based on conditionals, it passes this information to its parent
- Note that we are **not** completely traversing the tree in a time step
  - Can remain in a running sub-node for as long as it takes to resolve

NODE TASKS

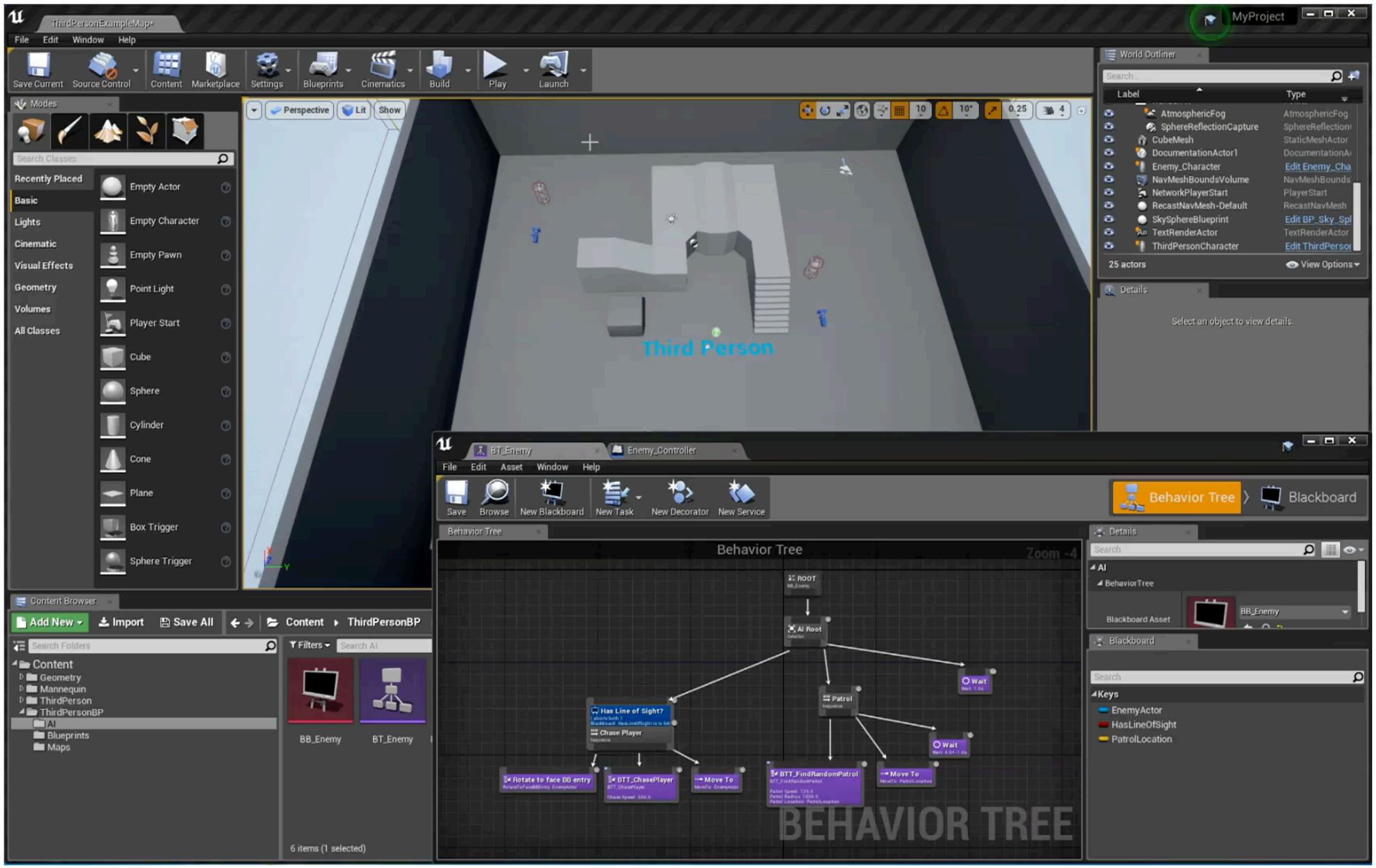
- Each node has a task or tasks, which define its behavior
- **Actions** define what the agent can do, including playing animations, moving, interacting, etc
- **Conditionals** determine if world/agent properties are in a specific state (often determine success or failure state of nodes)
- **Composites** are parent nodes that have 1 or more child nodes
  - Can check conditionals, and determine what order to execute child nodes in, etc
- **Decorators** can have one child
  - Modifies child behavior in some way (executes child multiple times, provides interrupts, continues to run child until it is successful, etc)

BEHAVIOR TREE EXAMPLE



Example from Project Zomboid

UE4 DEMO



https://www.youtube.com/watch?v=I60i4YLwqD8

UNREAL BEHAVIOR TREES

- Nomenclature and structure is a little different from the “standard” behavior trees
- **Composite** nodes are branch nodes
  - Determine basic rules of the tree
- **Task** nodes are leaf nodes
  - Perform actions
- **Decorator** nodes are *attached* to other nodes and determine if branch can be executed (functionally conditional nodes)
- **Service** nodes are *attached* to composite nodes and can update agent’s world knowledge and can execute children in parallel

UNREAL: BLACKBOARD

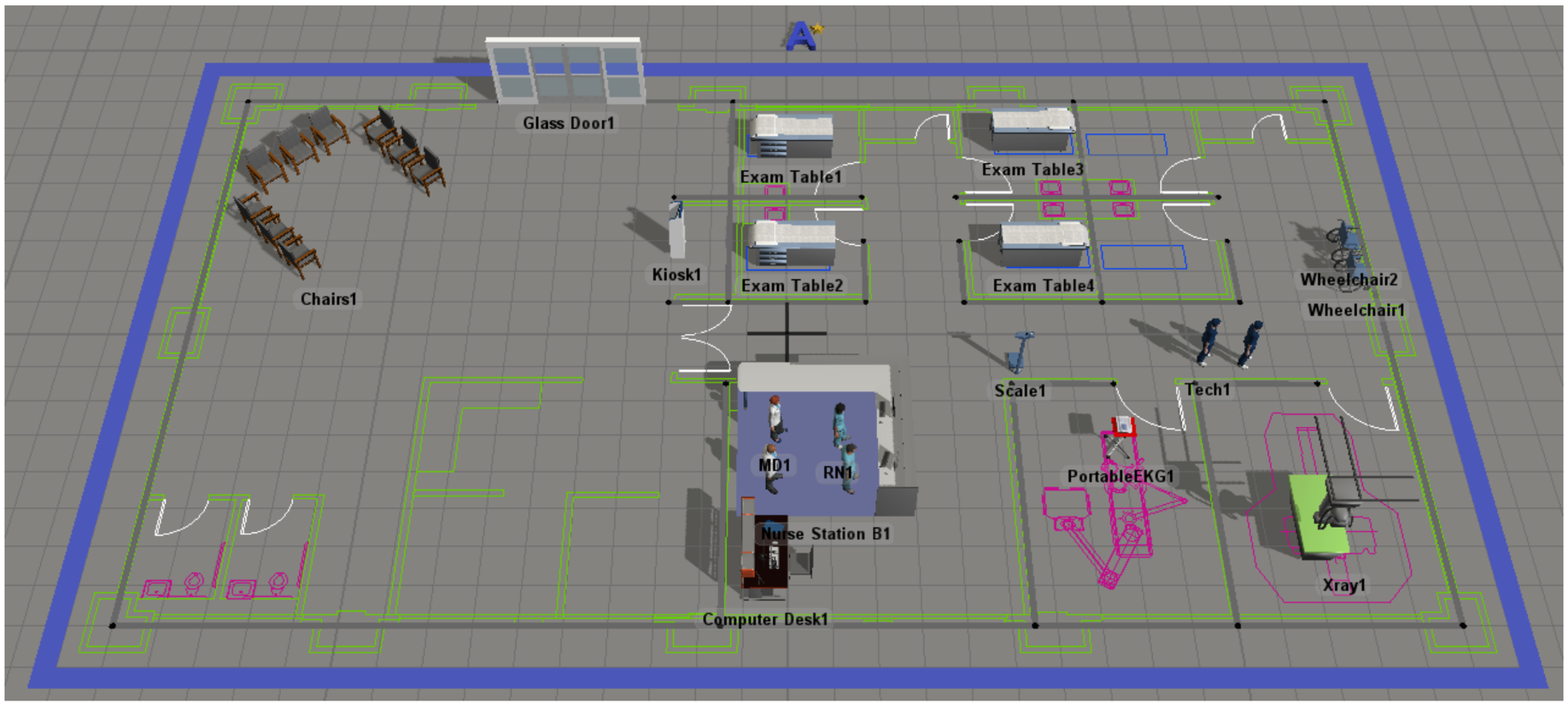
- World knowledge is everything that the agent knows
- Knowledge can be stored in a Blackboard as keys for accessible reading and writing
  - Knowledge can be local to one agent or a squad of agents
- Changes in keys can trigger events
  - Behavior trees in UE5 are event driven rather than tick-based
- Calculations can be cached for better responsiveness
- Centralized location of data results in fewer levels of indirection to access data

AI ACTIONS

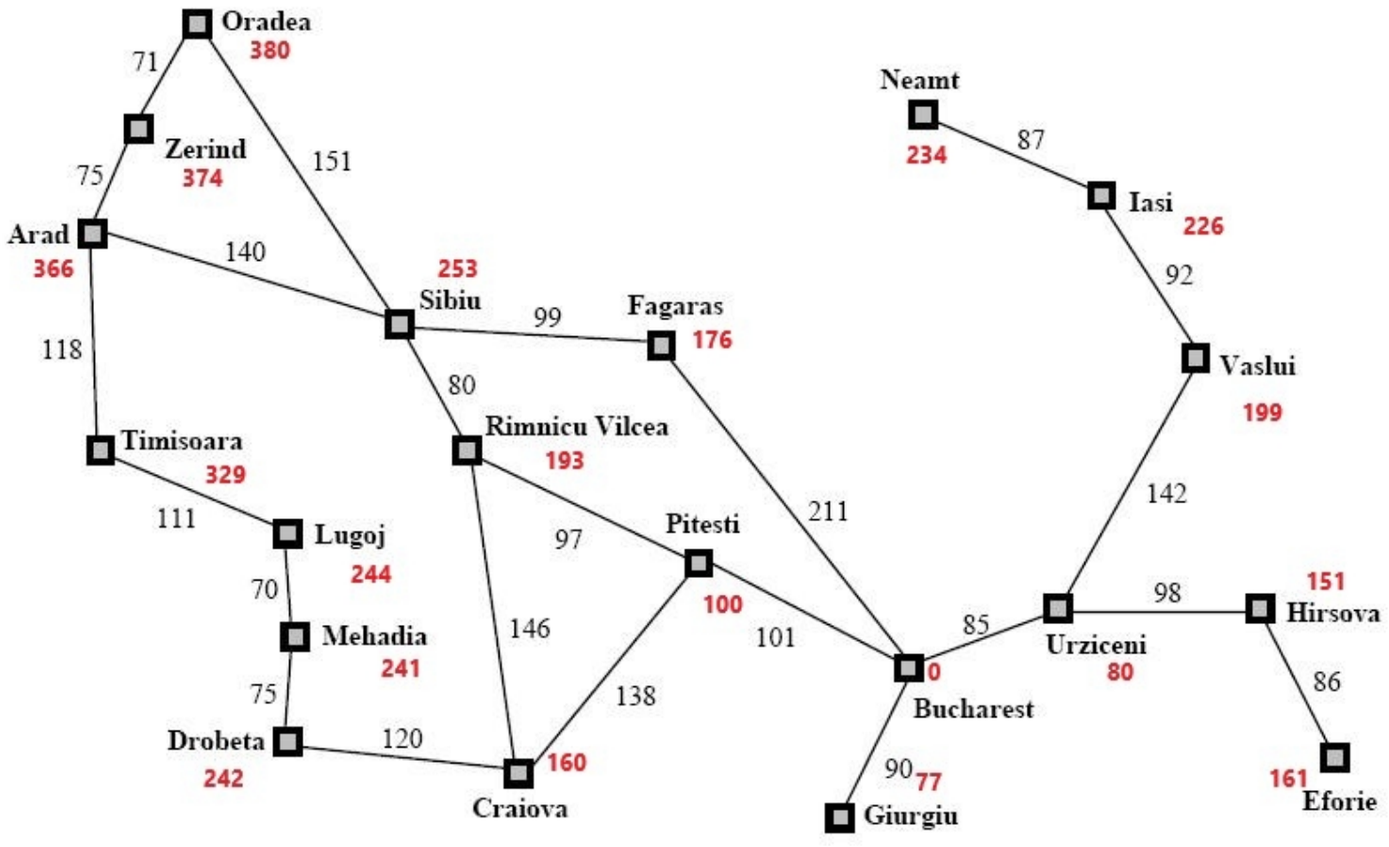
- Once a decision has been reached, the agent must perform the action
  - Update agent state
  - Play associated agent animation
  - Perform associated agent action
- One important action is navigation
  - How do we *convincingly* move the AI throughout the world?

NAVIGATION

- Agents use pathing algorithms to navigate through the world
- Problem: how do I get from point A to point B?
- Constraints: respect obstacles and emulate human choices

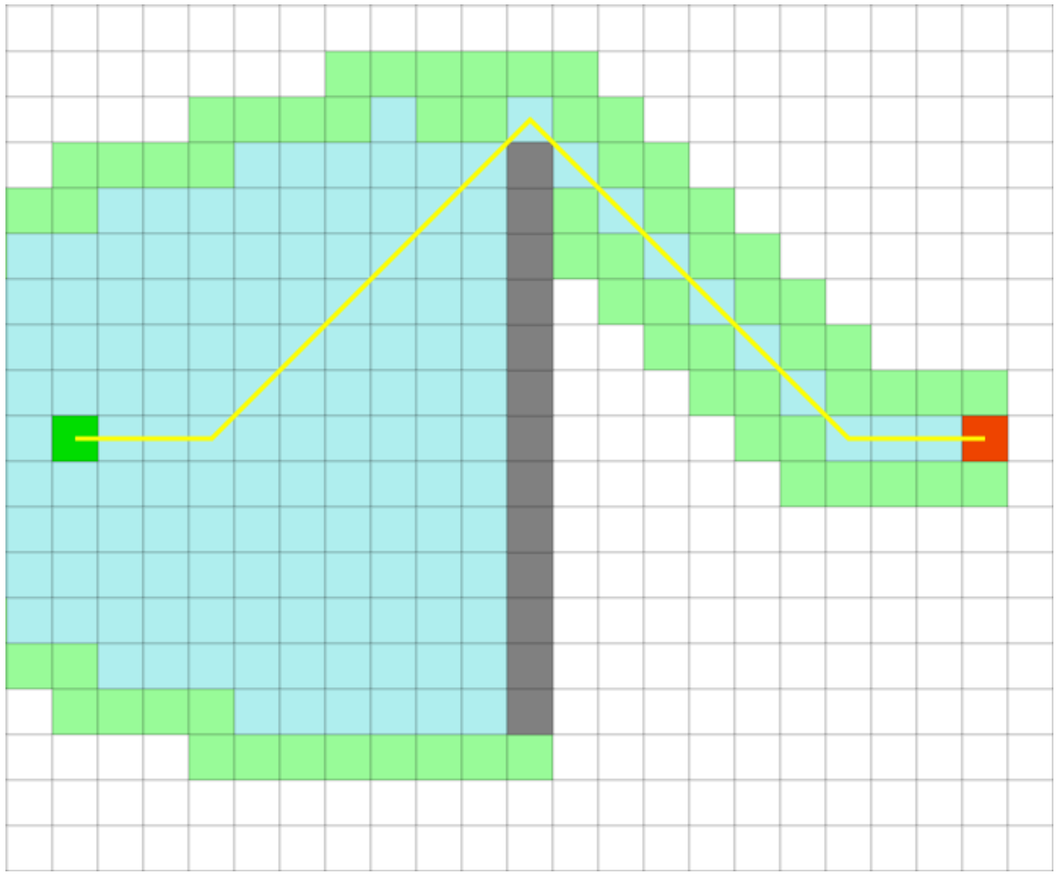


A TYPE OF GRAPH PROBLEM



Node-based

Grid-based



A* PATHFINDING

- De facto algorithm for agent navigation in games
- Considers two functions to optimize:
  - g(n): Current best **cost** for getting **to** a node from the start
  - h(n): Current best **estimate** for how much more it will cost to reach the goal **from** a node
- Heuristic used for h(n) determines optimality and efficiency

A* CHALLENGES

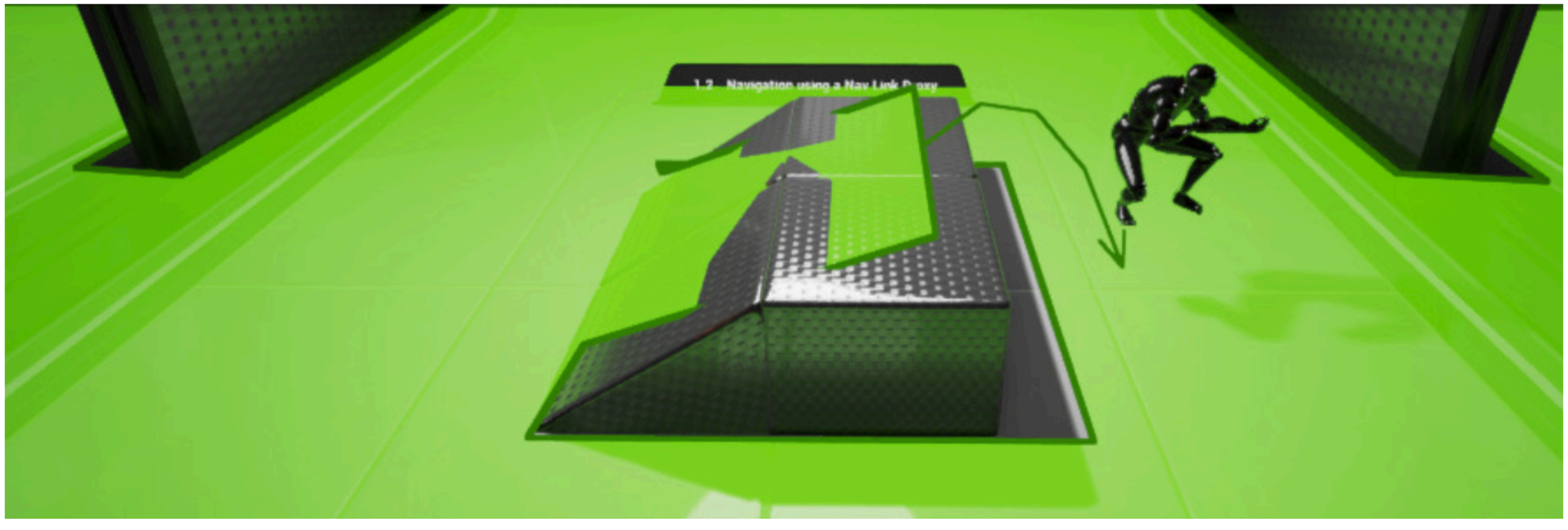
- Expensive to do across large areas
  - Use of waypoints and hierarchical planning to reduce state complexity
- Must consider how to replan if world state changes
  - Player or other agents change position
  - World contains dynamic or destructible obstacles
- Must store data efficiently to handle multiple agents

NAVMESHES

- Navigation meshes (navmeshes) aid in the computation of paths for AI agents
- Create a polygon mesh that defines where agents can walk
  - Polygons provide nodes for calculating A*
  - Can traverse the polygon itself along a linear path
- Simplifies collision detection (if placed properly, ensures agent will not collide with geometry)
- Generated automatically or hand-created by designer

UE5 NAVMESHES

- Provides all basic features for agent traversal
- Can be placed in editor (no need for lower level programming)
- Challenge is for the designer to make it compelling and functional



CROWD SIMULATION

- Large number of agents navigating world, avoiding each other and player, and engaging in different goals
  - Often treated as a “particle system” using rules and forces

Cyberpunk 2077

FURTHER READING

- What is a Behavior Tree? <https://opsive.com/support/documentation/behavior-designer/what-is-a-behavior-tree/>
- Behavior Trees for AI: How they work<https://www.gamasutra.com/blogs/ChrisSimpson/20140717/221339/Behavior_trees_for_AI_How_they_work.php>
- UE4 Behavior Tree Documentation <https://docs.unrealengine.com/en-US/Engine/ArtificialIntelligence/BehaviorTrees/index.html>
- UE4 NavMeshes <https://docs.unrealengine.com/en-US/Resources/ContentExamples/NavMesh/index.html>