Using AI in Unreal

New Concepts (High-Level Overview)

- Navigation Mesh Volume:
  - Lets you define bounds for each AI element in your game
  - Deals with obstacles, etc automatically

- AI Components:
  - AIController
    - Analogous to PlayerController, except now handled by AI
  - Blackboard
    - A list of variables for use by the AI called blackboard keys
  - Behavior Tree
    - Definition of behavior of your AI sits here
  - Behavior Tree Tasks
    - Sub-tasks run by the behavior tree to set the state of the AI

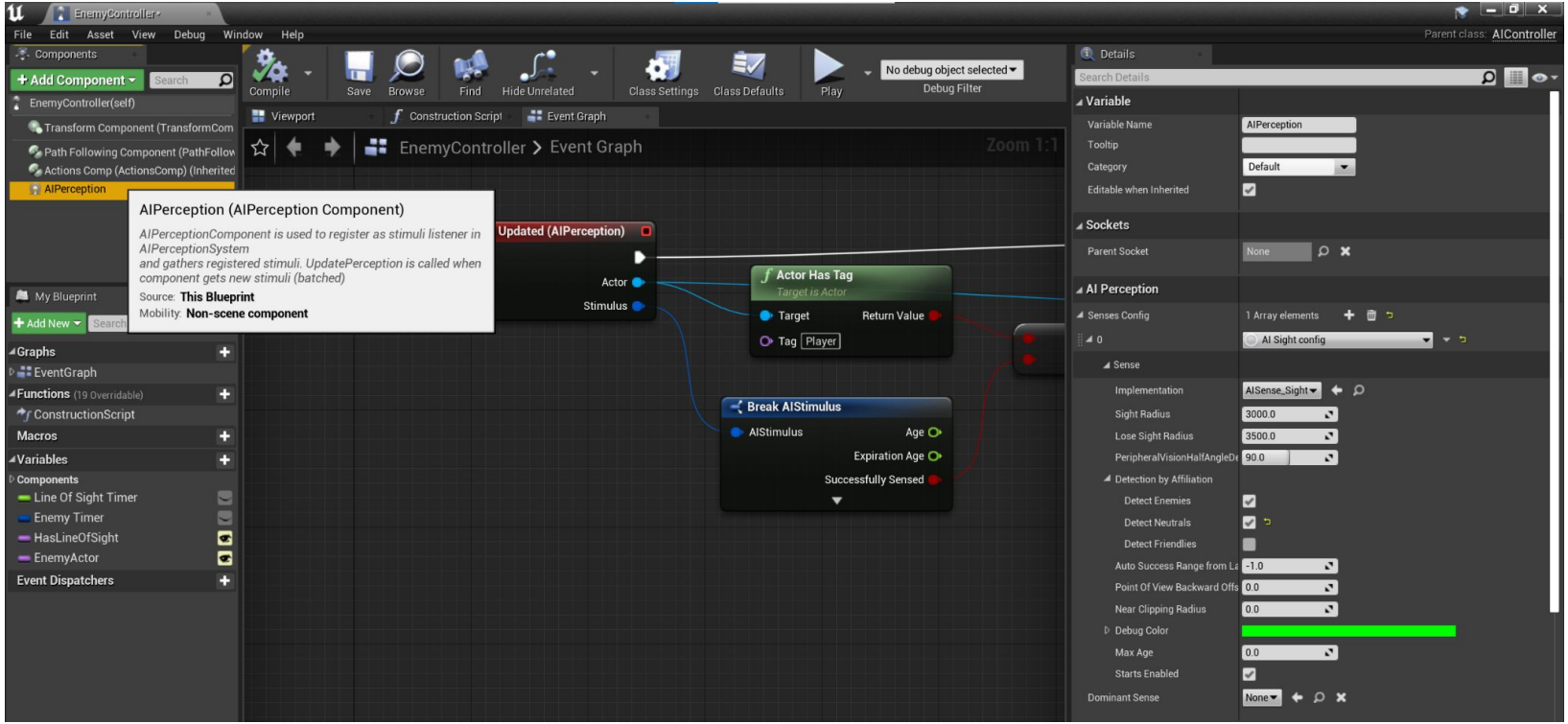
AI Controller

- Observes surroundings to make decisions (without player input)

- Example: BP_Enemy instances are controlled by EnemyController (derived from the AIController class)

- EnemyController is set to operate according to the BT_Enemy behavior tree when it possesses a pawn

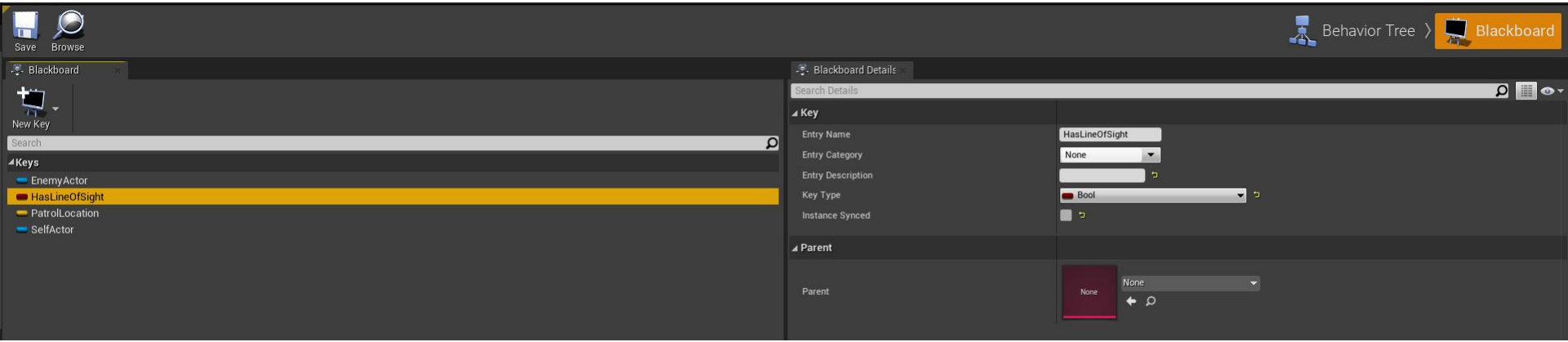
- AIControllers can have AIPerception components to perceive the world

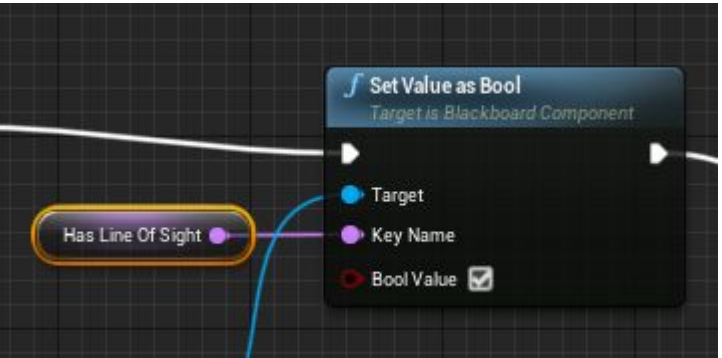


- Example: an AI Sight config can be used to detect if the BP_Player is seen

Blackboard

- Blackboard variables keep track of the state of the AI
  - Values can be accessed and modified by querying the name of the keys

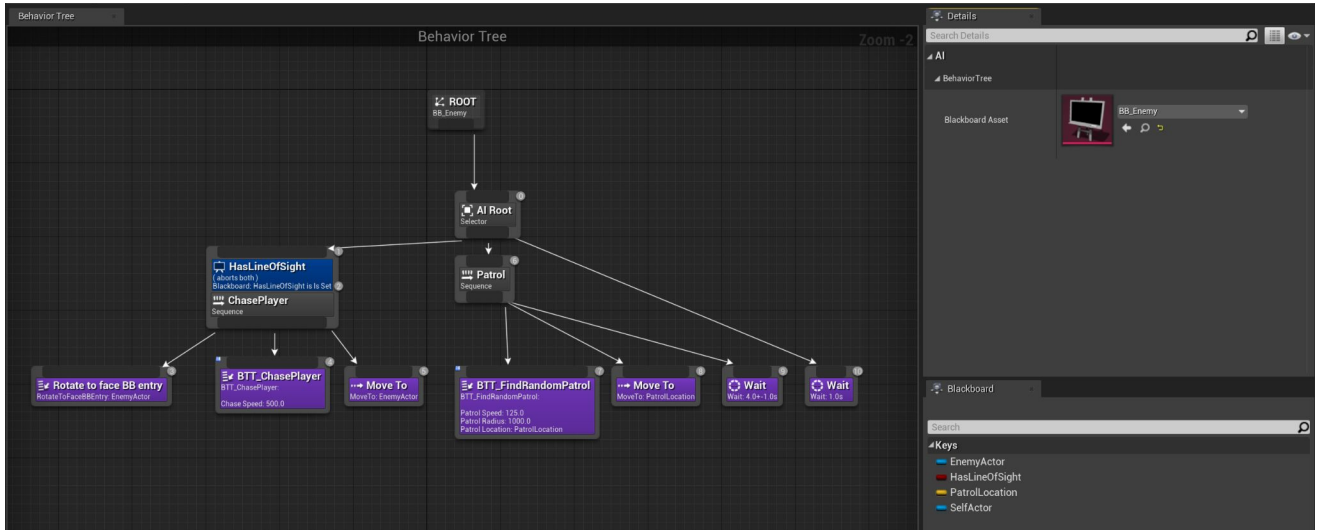




- Example: HasLineOfSight keeps track of whether or not the AI has sight of the player
  - Toggled by the AIController

Behavior Tree

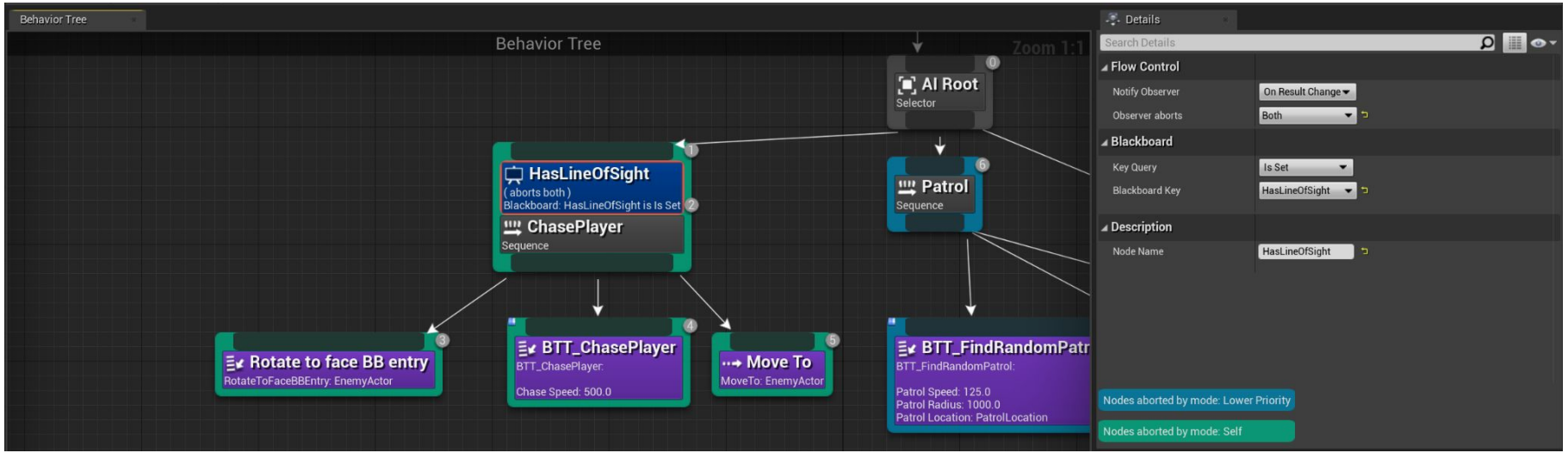
- Behavior tree handles logic for how to react based on blackboard values
  - Linked to a blackboard asset to access its keys



- Executes from top-down and then left-right
  - Purple nodes are tasks that are run by certain flow control nodes

Decorators

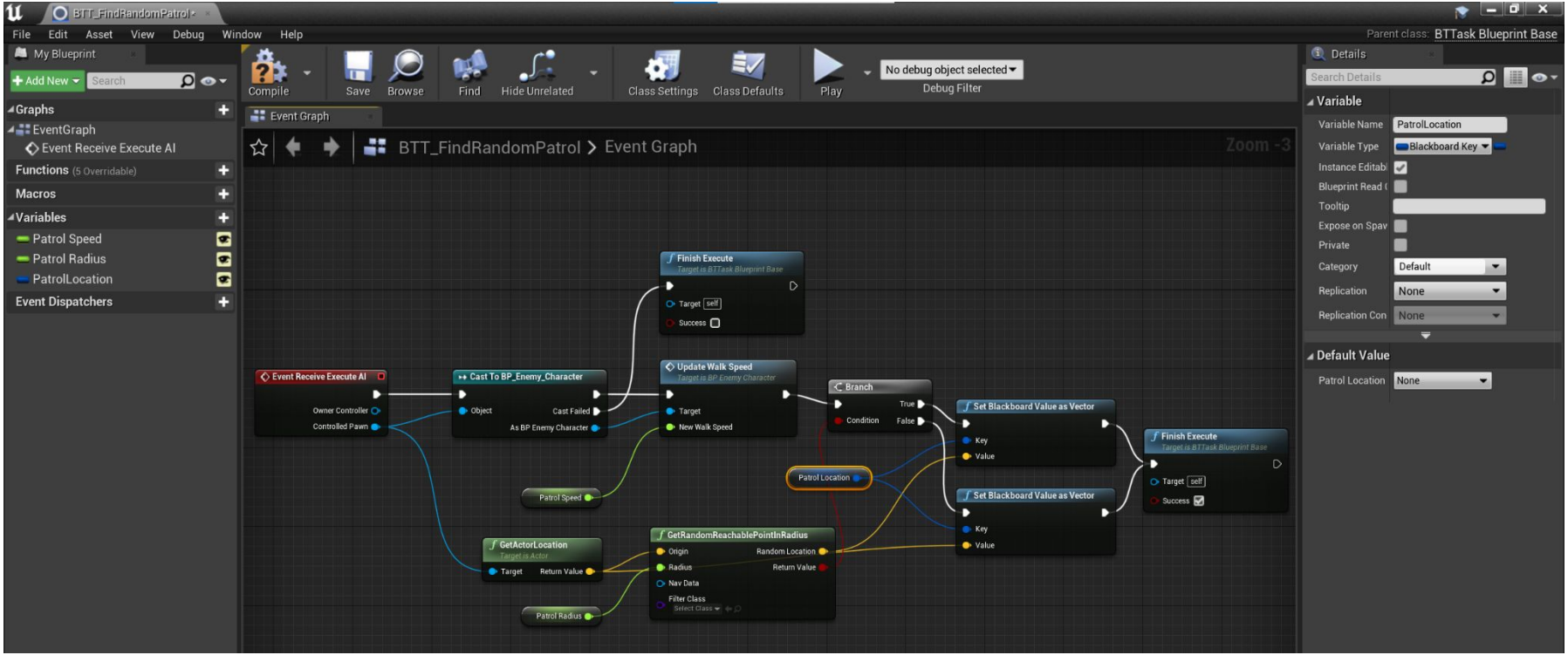
- Decorators help manage decisions made in the behavior tree



- Example: whenever HasLineOfSight is updated (Is Set or not):
  - If HasLineOfSight becomes false while chasing, stop chasing
  - If HasLineOfSight becomes true while patrolling, start chasing

Behavior Tree Tasks

- UE4 comes with default behavior tree tasks, but custom ones can be defined as well



- Example: the enemy finds a new location (based on the generated NavMesh) to patrol to, and queries the blackboard with key “PatrolLocation” to update the value

Full Example:

1. BP_Enemy instance spawns
    a. Set to be controlled by EnemyController (derived from AIController) on spawn
2. If BP_Player is in sight of BP_Enemy, the BP_Enemy is set to chase player
    a. EnemyController perceives world to check for BP_Player
        i. If BP_Player is found, set blackboard key “EnemyActor” to player location and “HasLineOfSight” to true
        ii. Move to “EnemyActor” when “HasLineOfSight” is true
3. If not, BP_Enemy chooses a random location to walk to
    a. Use the generated NavMesh to choose a random reachable point in given radius
    b. Set blackboard key “PatrolLocation” to the output point
    c. Move to “PatrolLocation”

Key Points:

- Pawns can be controlled by AIControllers (instead of PlayerControllers)
- The BehaviorTree controls the decisions of the AI
  - AIController runs the BehaviorTree
  - Blackboard asset is attached to the BehaviorTree
    - Keeps track of AI state
- Both the AIController and BehaviorTreeTasks can modify Blackboard keys
  - BehaviorTree runs based on the Blackboard values

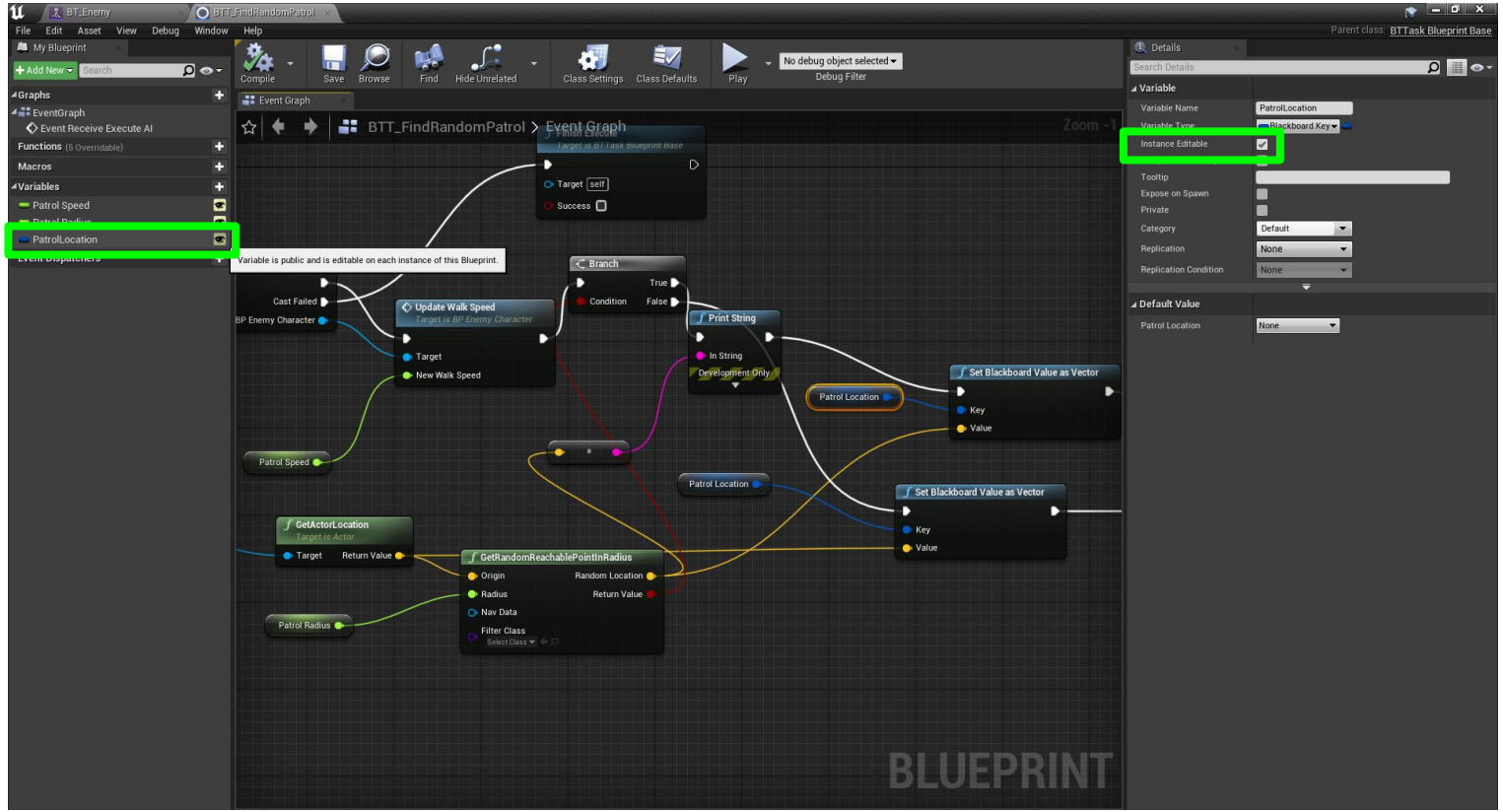
Links to Tutorials (Implementation Details):

- Setting up basic navigation using a NavMesh: https://docs.unrealengine.com/4.27/en-US/InteractiveExperiences/ArtificialIntelligence/NavigationSystem/

- Setting up a “guard” character that randomly patrols the map and follows the player when player is visible: https://docs.unrealengine.com/4.27/en-US/InteractiveExperiences/ArtificialIntelligence/BehaviorTrees/BehaviorTreeQuickStart/

Troubleshooting (Tutorial Obscurities):

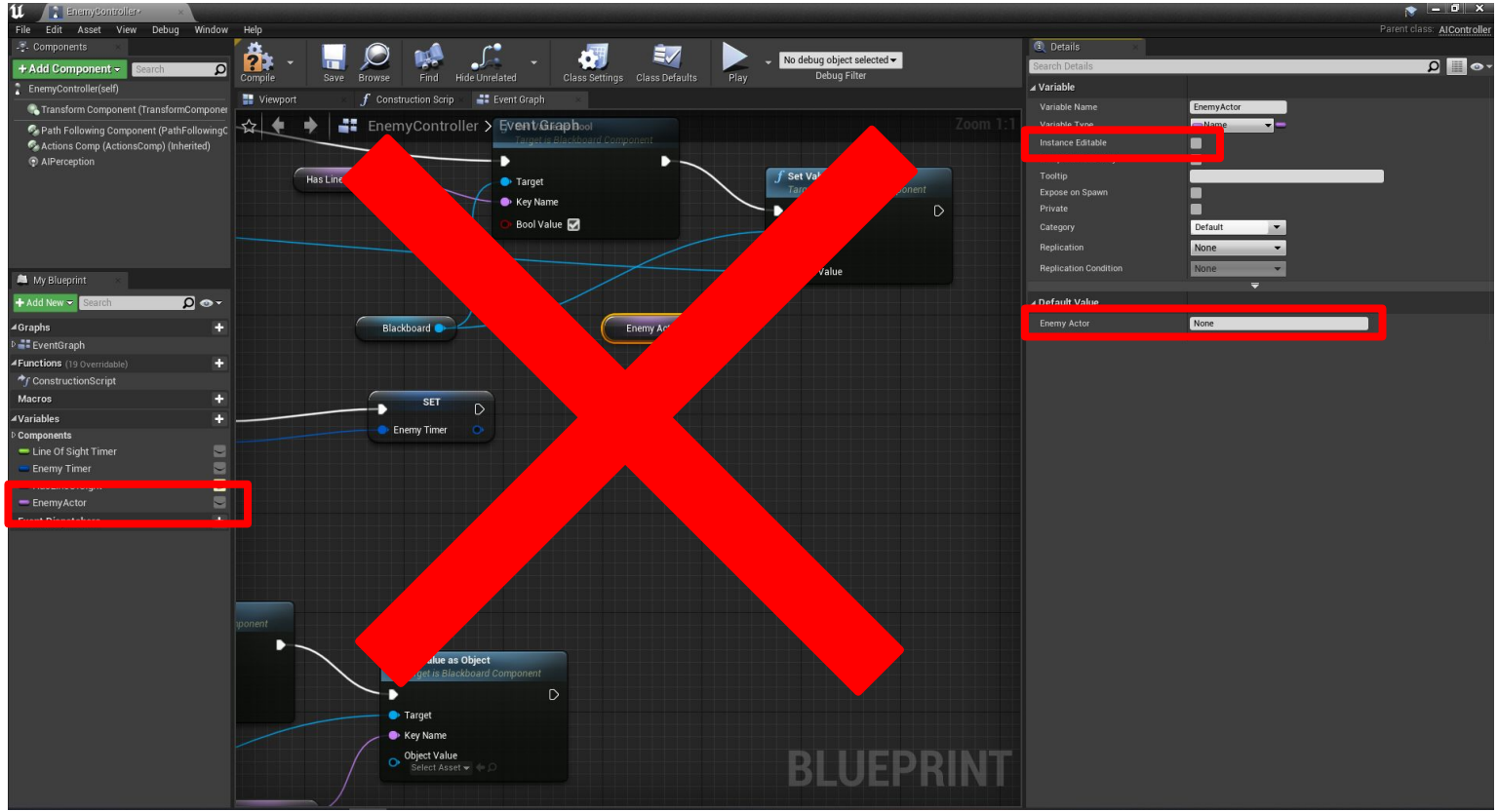
- Unable to set blackboard variables through tasks



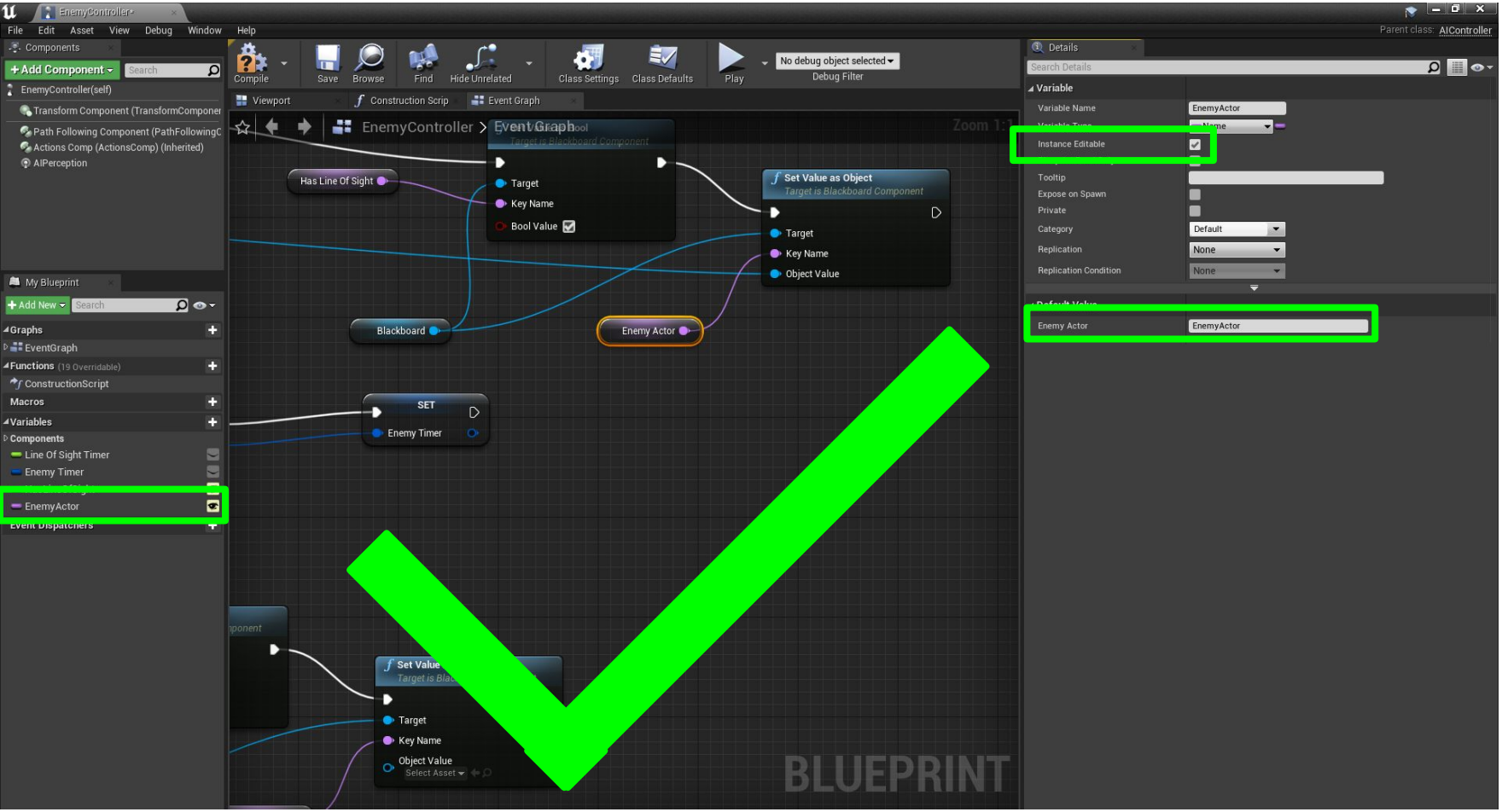
Make sure the promoted variable is public and instance editable!

Troubleshooting (cont.)

- Unable to set blackboard variables through AIController



- Here the variable isn’t instance editable, and is actually querying the key “None”



- The variable is instance editable and querying the correct key “EnemyActor”