Alex J. Champandard AiGameDev.com

Michael Dawe Big Huge Games

David Hernandez-Cerpa LucasArts

Behavior Trees: Three Ways of Cultivating Game AI

AI Summit

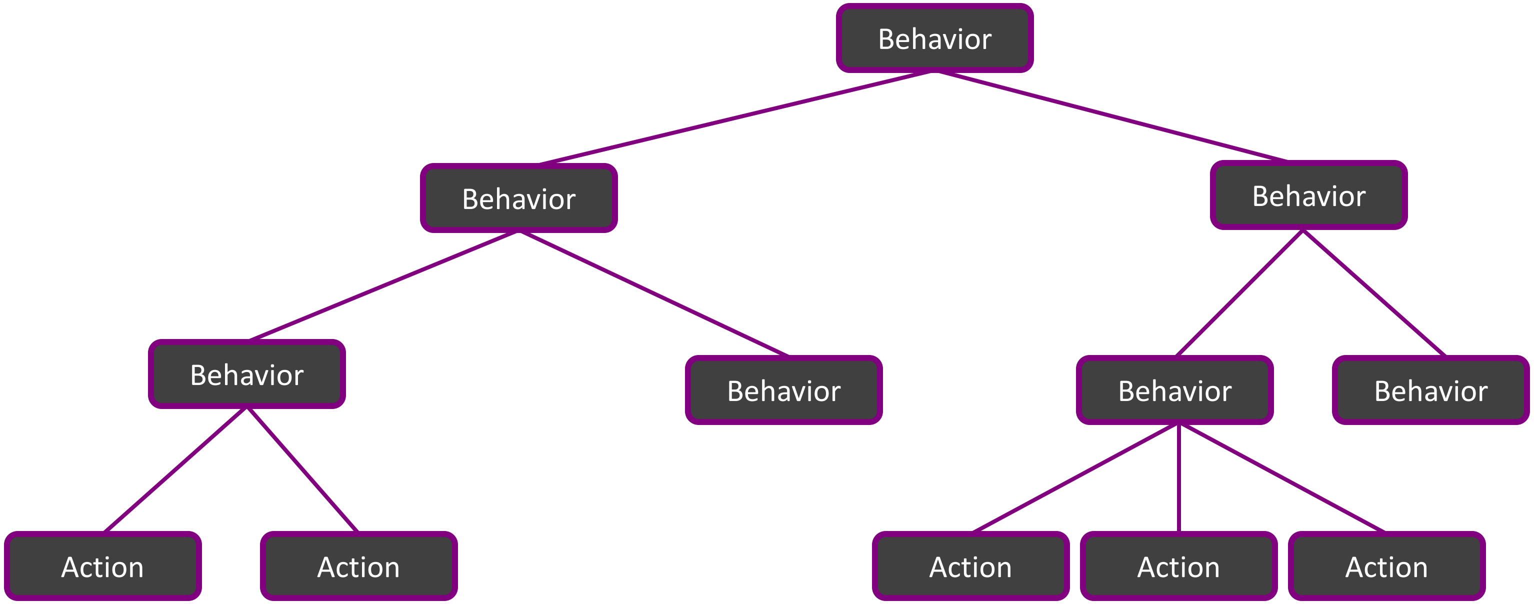
BEHAVIOR TREES APPLIED!

- Halo 3 & ODST
- [PROTOTYPE]
- Spore
- GTA: Chinatown Wars
- The Bourne Conspiracy
- SWAT 4, Bioshock
- Dark Sector
- …

FEATURES

- BTs are a framework for game AI.
- BTs model character behaviors extremely well.
- BTs are simple, yet extremely customizable.

OVERVIEW


Behavior
Behavior
Behavior
Behavior
Behavior
Behavior
Behavior
Action
Action
Action
Action
Action

AGENDA

1) Building Blocks
2) Design Patterns
3) Script Integration
4) Debugging
5) Discussion

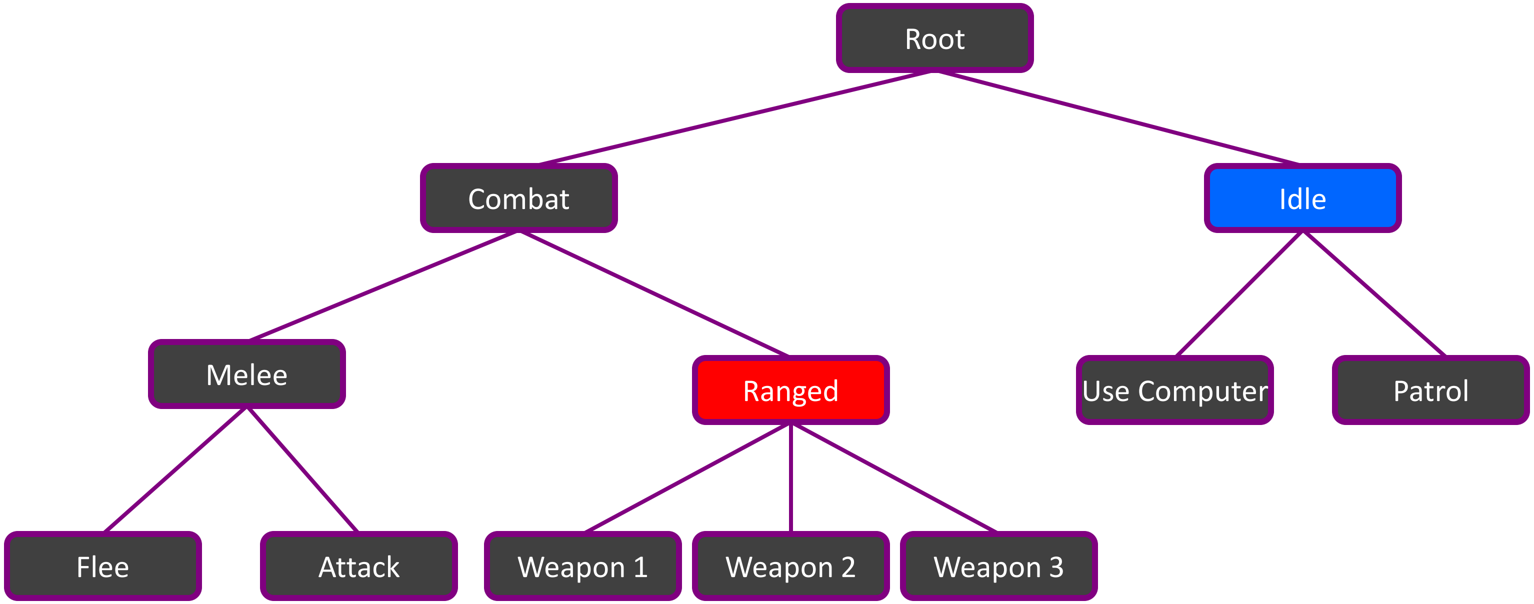
Behavior Trees Part 1, David

BUILDING BLOCKS

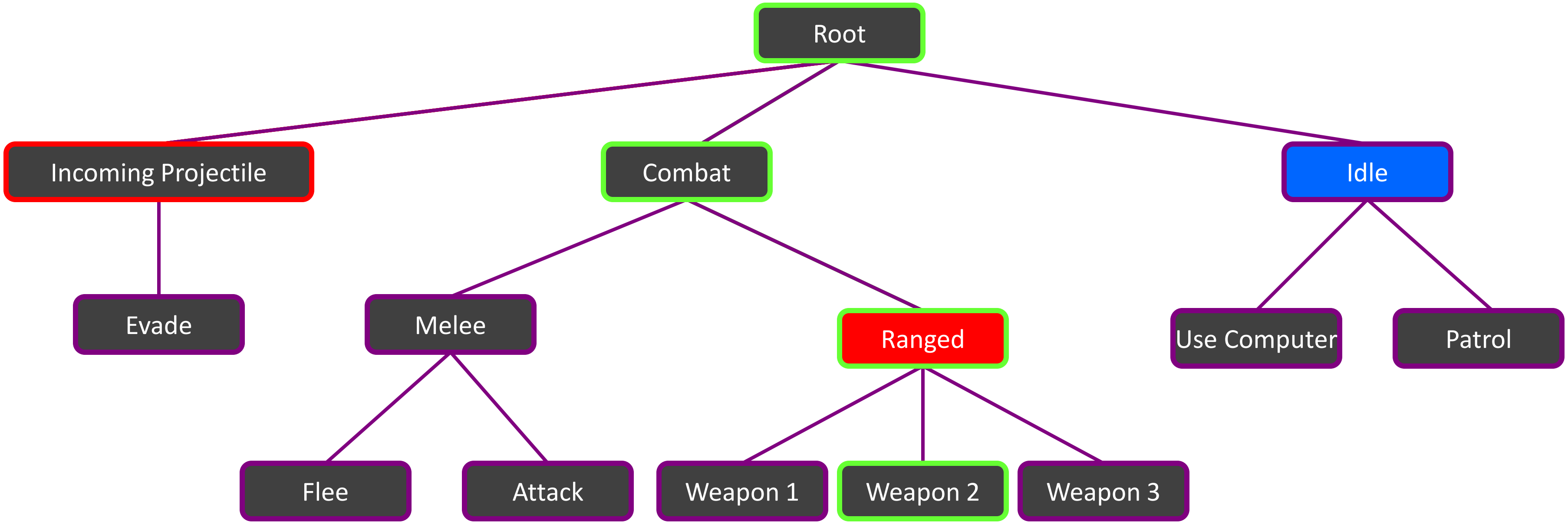
NODE TYPES

- Priority
  - Child nodes are evaluated in order until one validates
- Sequential
  - First child is validated and executed
  - When it is finished, the next one is validated
- Stochastic
  - All children are validated
  - A random node is selected among the valid ones

BEHAVIOR TREE UPDATE



EVENT-DRIVEN BEHAVIORS

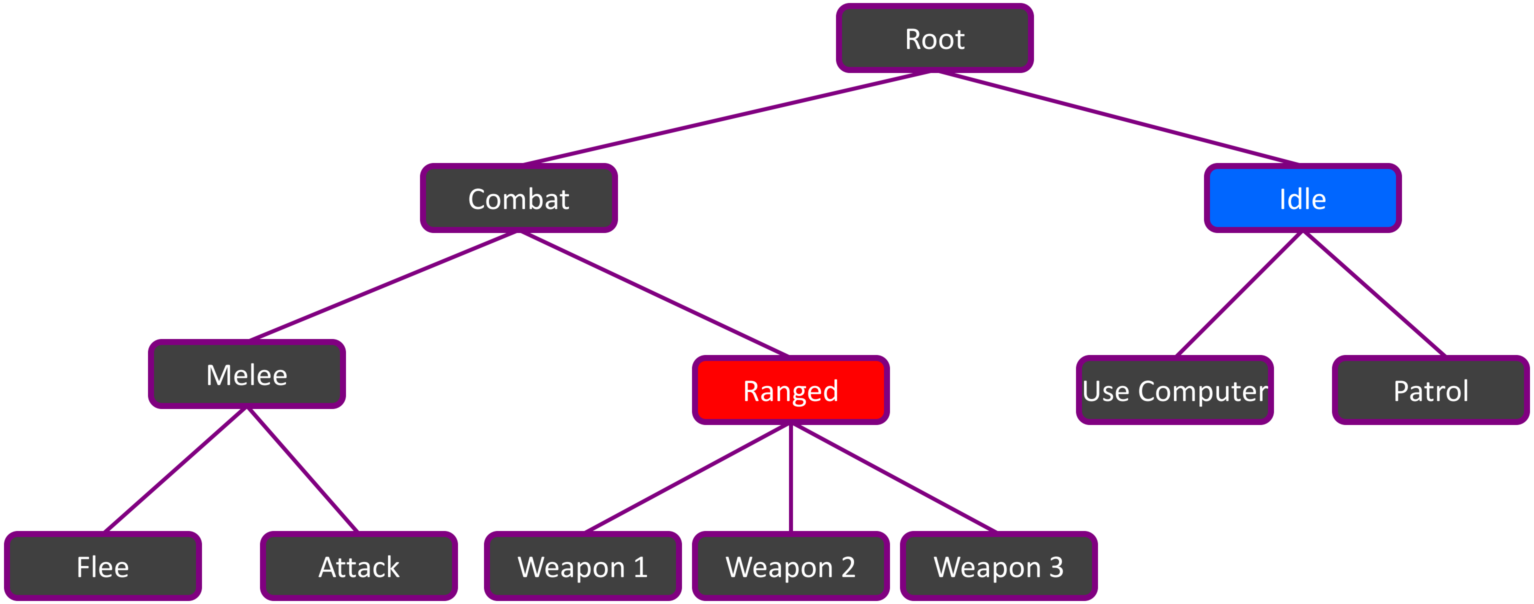


- Stimulus types
  - Disabled by event
  - Autodisabled

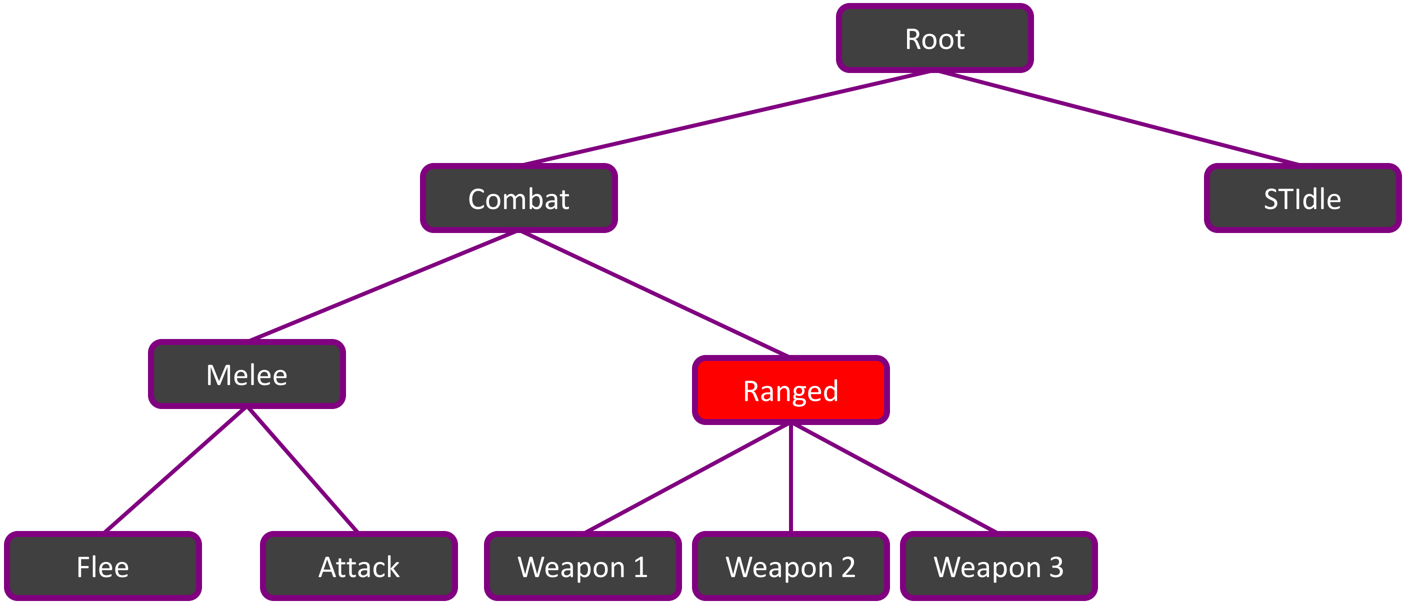
DYNAMIC BEHAVIORS

- Dynamic behaviors support
  - Level specific content
    - Patrols
    - Initial setups
    - Story driven events
  - DLC
- Behaviors are added to actors in the level (enticers)
  - When a NPC uses the actor, it attaches the behavior to the tree

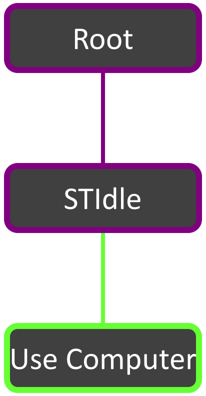
DYNAMIC BEHAVIORS



DYNAMIC BEHAVIORS



DYNAMIC BEHAVIORS



- Validate
  - Look for enticers
- Update
  1. Move to enticer
  2. Wait for other NPCs
  3. Subscribe
     - Attach new behavior to the tree
  4. Wait for behavior to finish
  5. Unsubscribe
     - Remove behavior from the tree

Behavior Trees Part 2, Alex

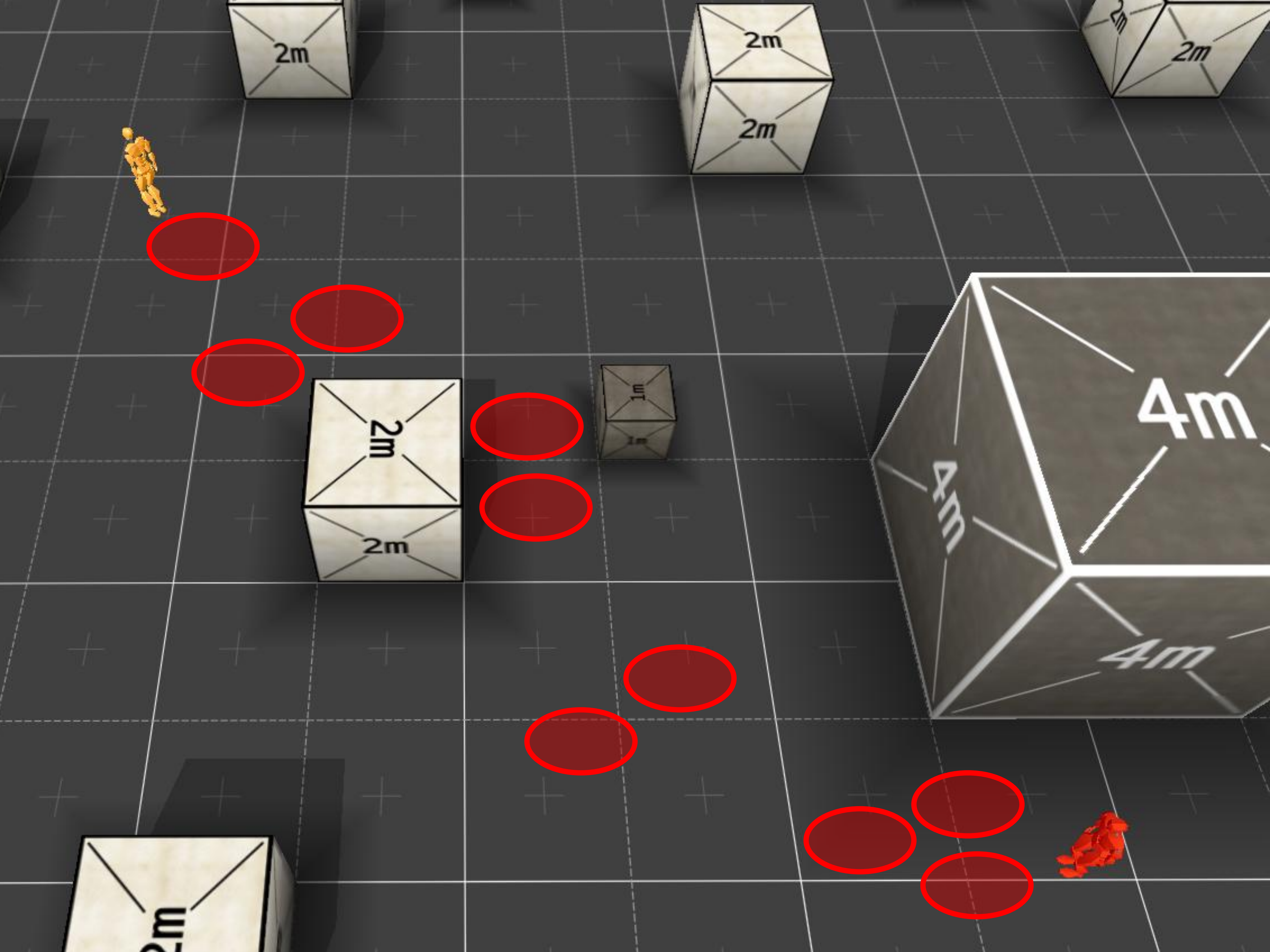
DESIGN PATTERNS

IN THE NEXT 10 MINUTES, FIND OUT...

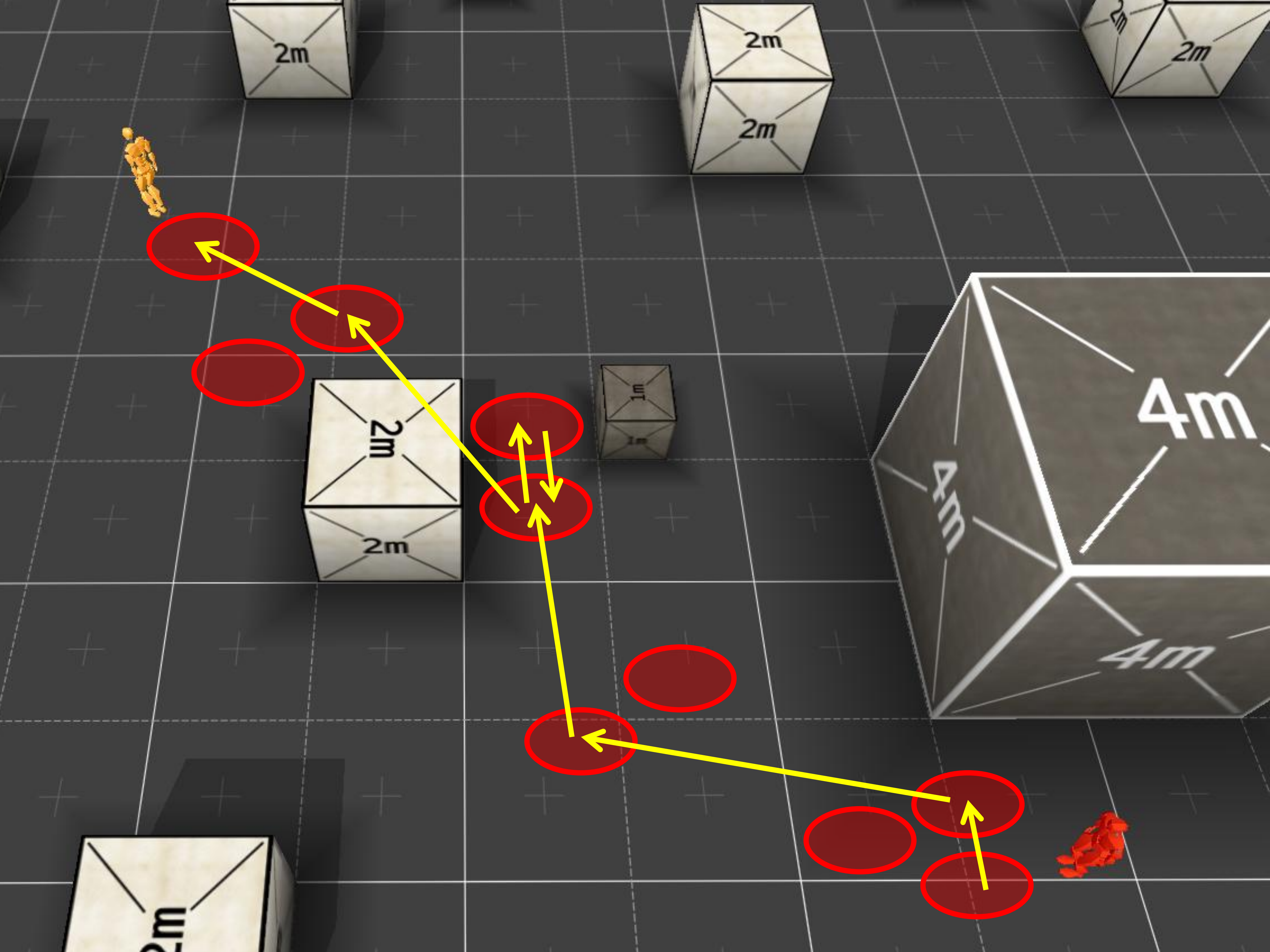
What’s the biggest problem developers face working with behavior trees and scaling up?

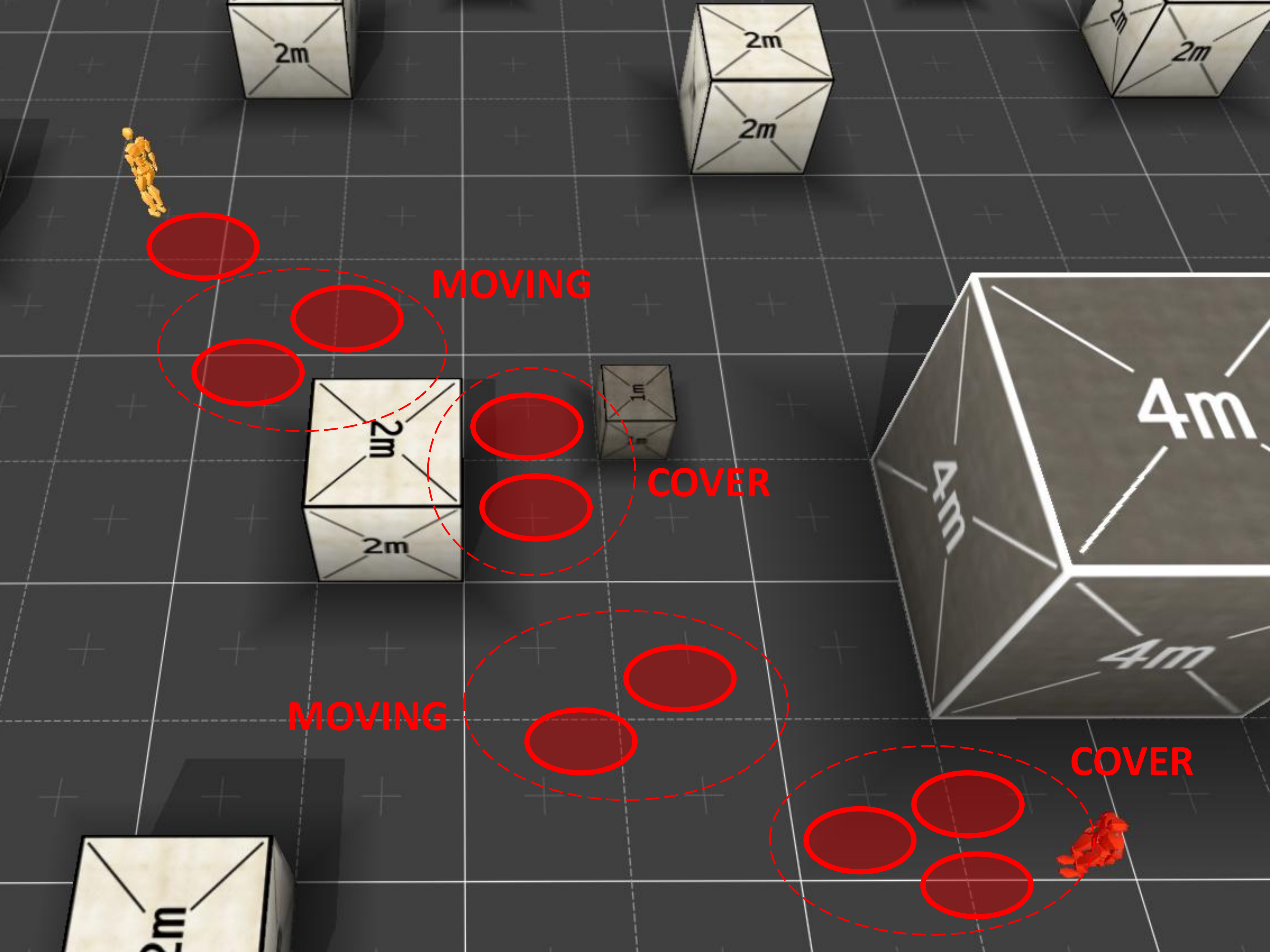
IN THE NEXT 10 MINUTES, FIND OUT…

When should you build your BT like a HFSM, and what happens if you do?

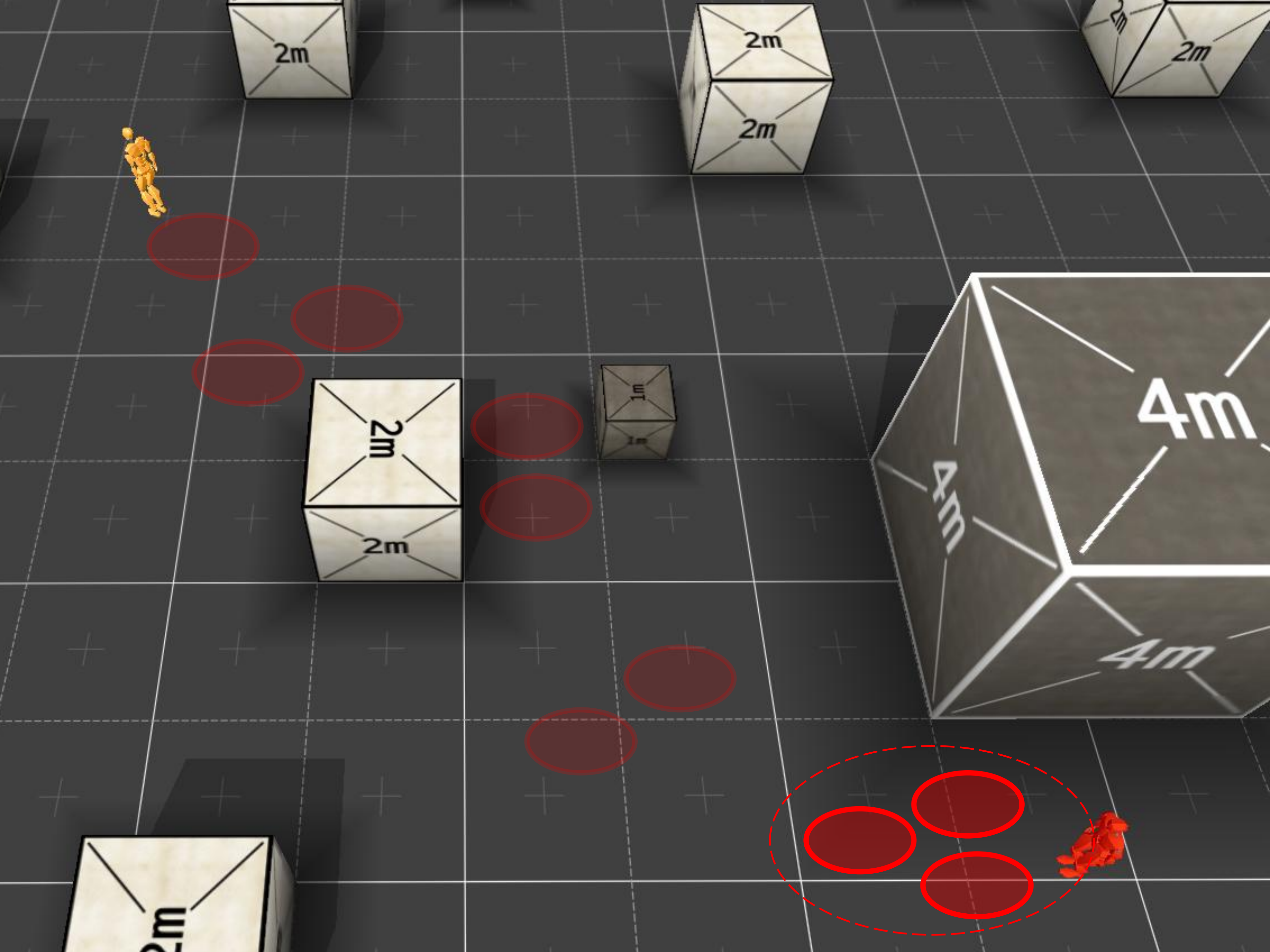


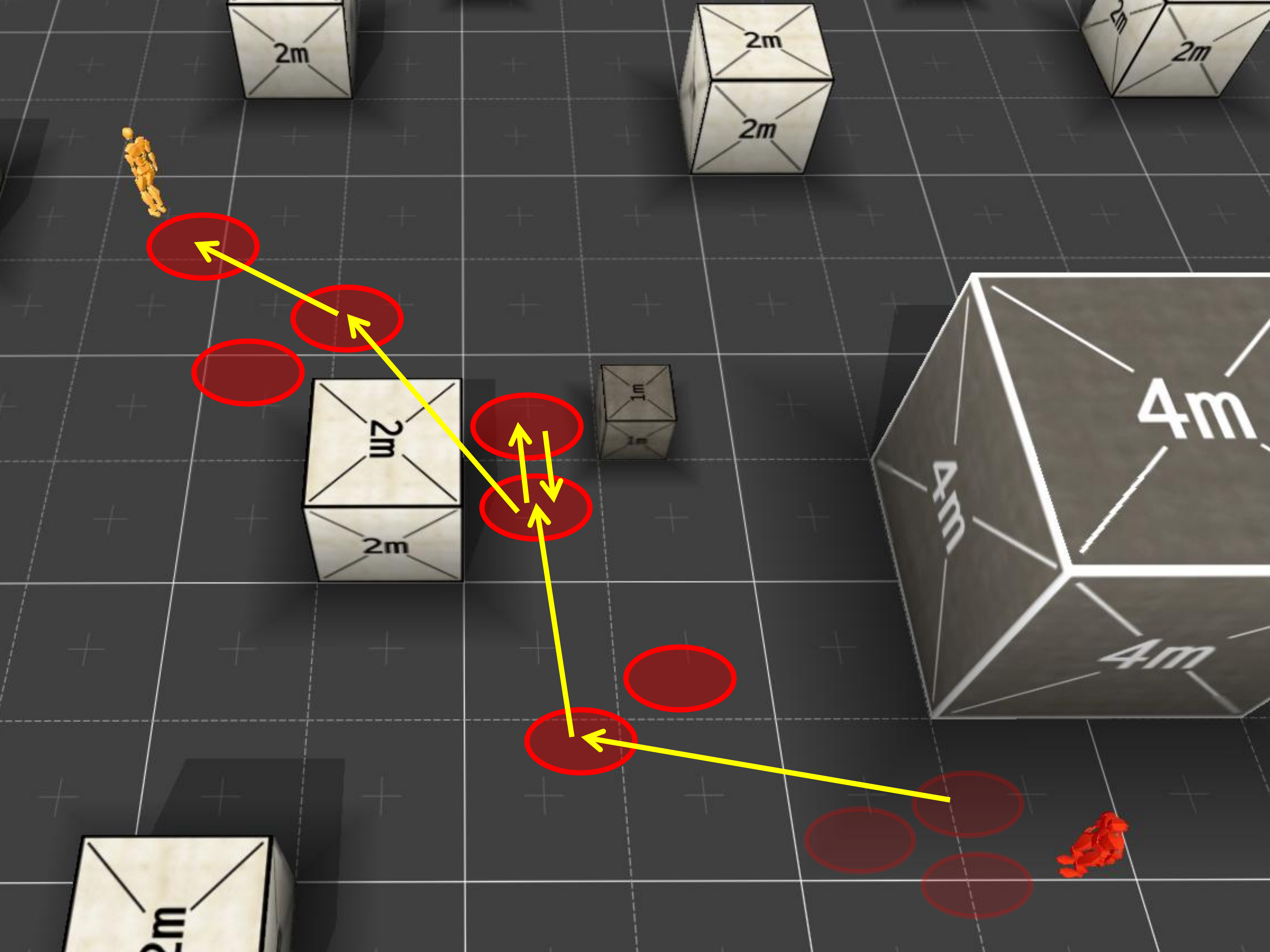
2m
2m
2m
2m
2m
2m
1m
1m
4m
4m
4m
2m
2m
2m

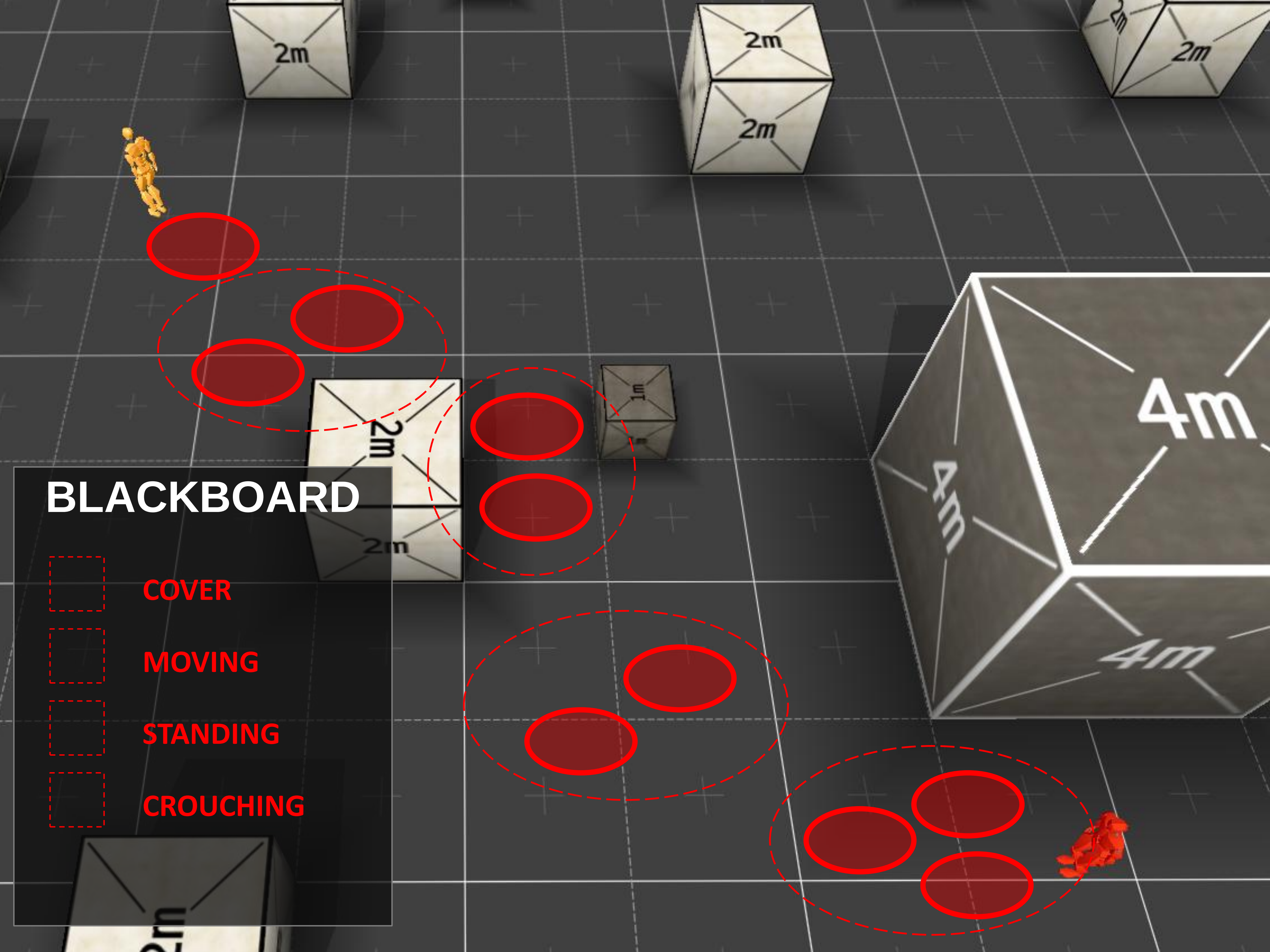

2m
2m
2m
2m
2m
2m
1m
1m
2m
2m
4m
4m
4m
2m



2m
2m
2m
2m
MOVING
1m
2m
COVER
2m
4m
4m
4m
MOVING
COVER
2m

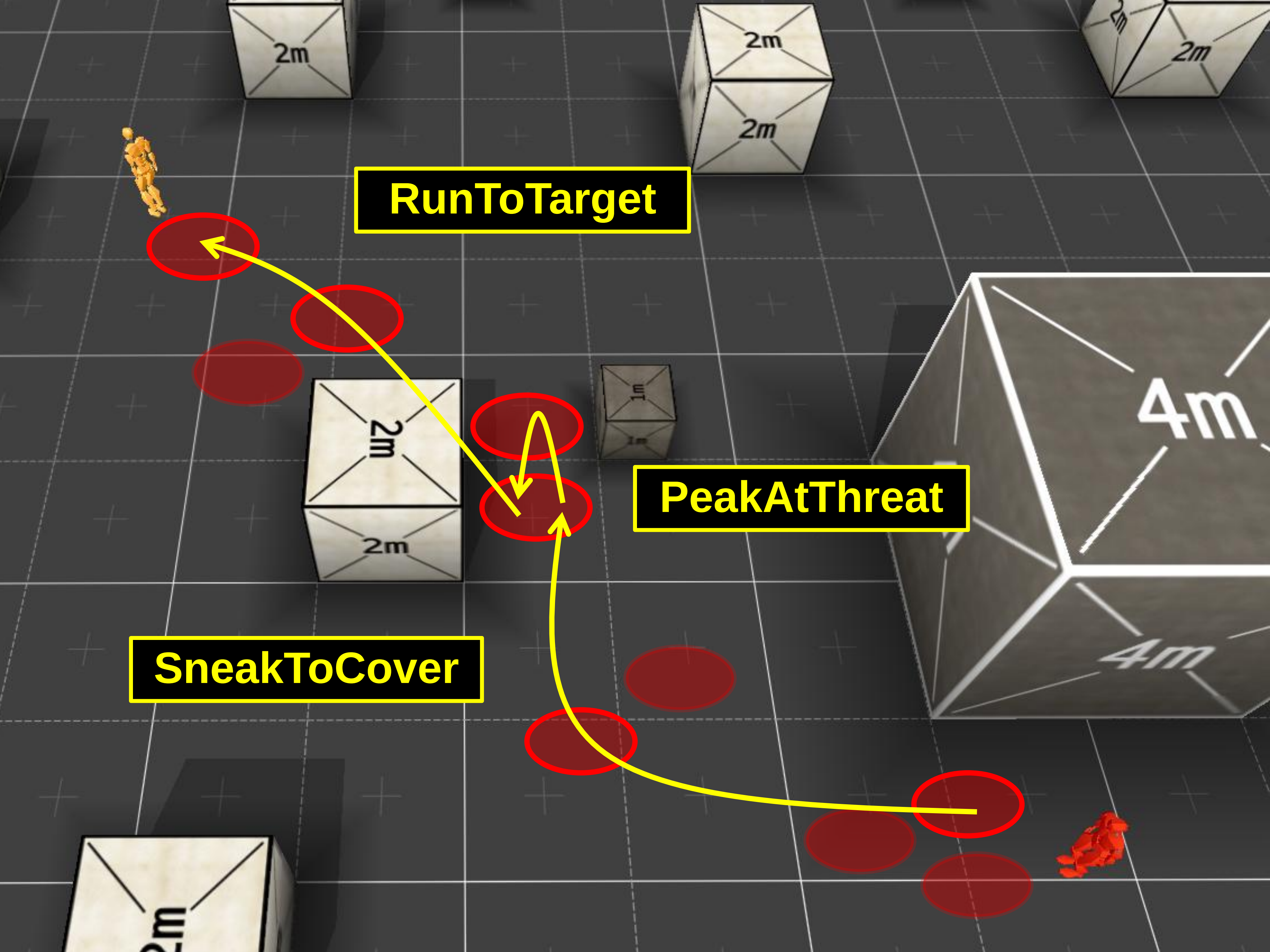

2m
2m
2m
2m
2m
2m
1m
1m
4m
4m
4m
2m
2m
2m


2m
2m
2m
2m
2m
1m
1m
2m
2m
4m
4m
4m
2m



BLACKBOARD

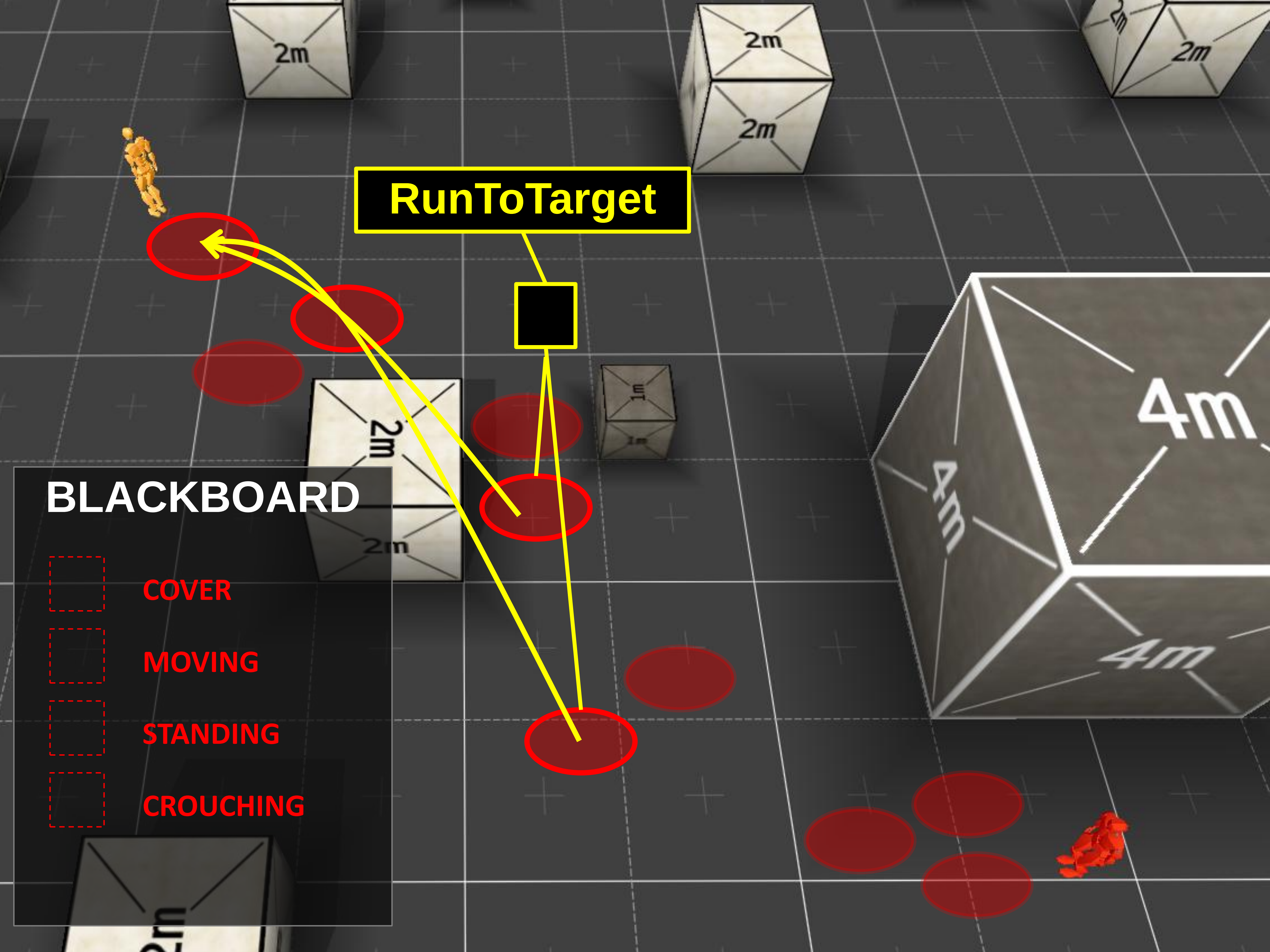
- □ COVER
- □ MOVING
- □ STANDING
- □ CROUCHING



RunToTarget

PeakAtThreat

SneakToCover



RunToTarget

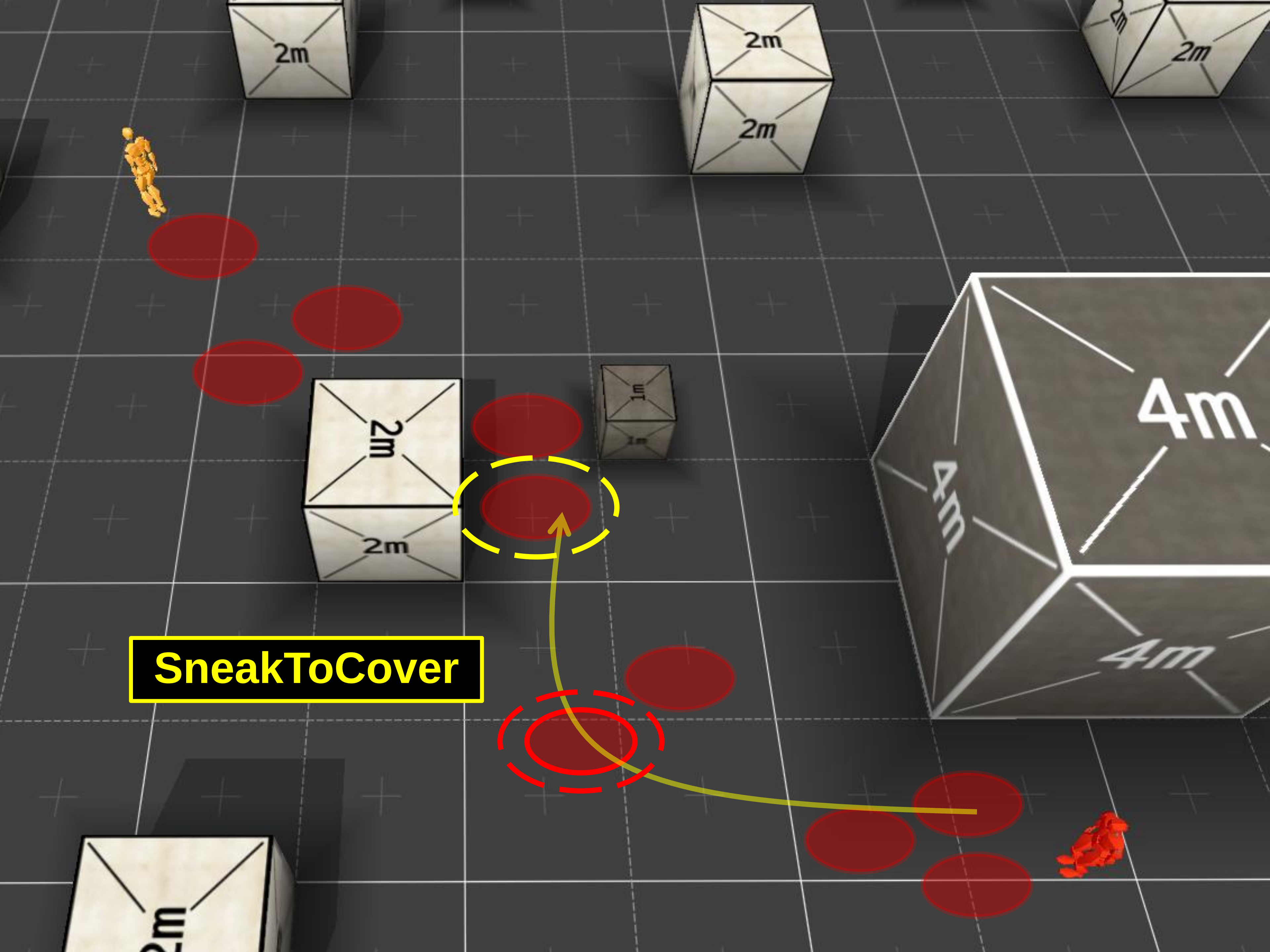
BLACKBOARD

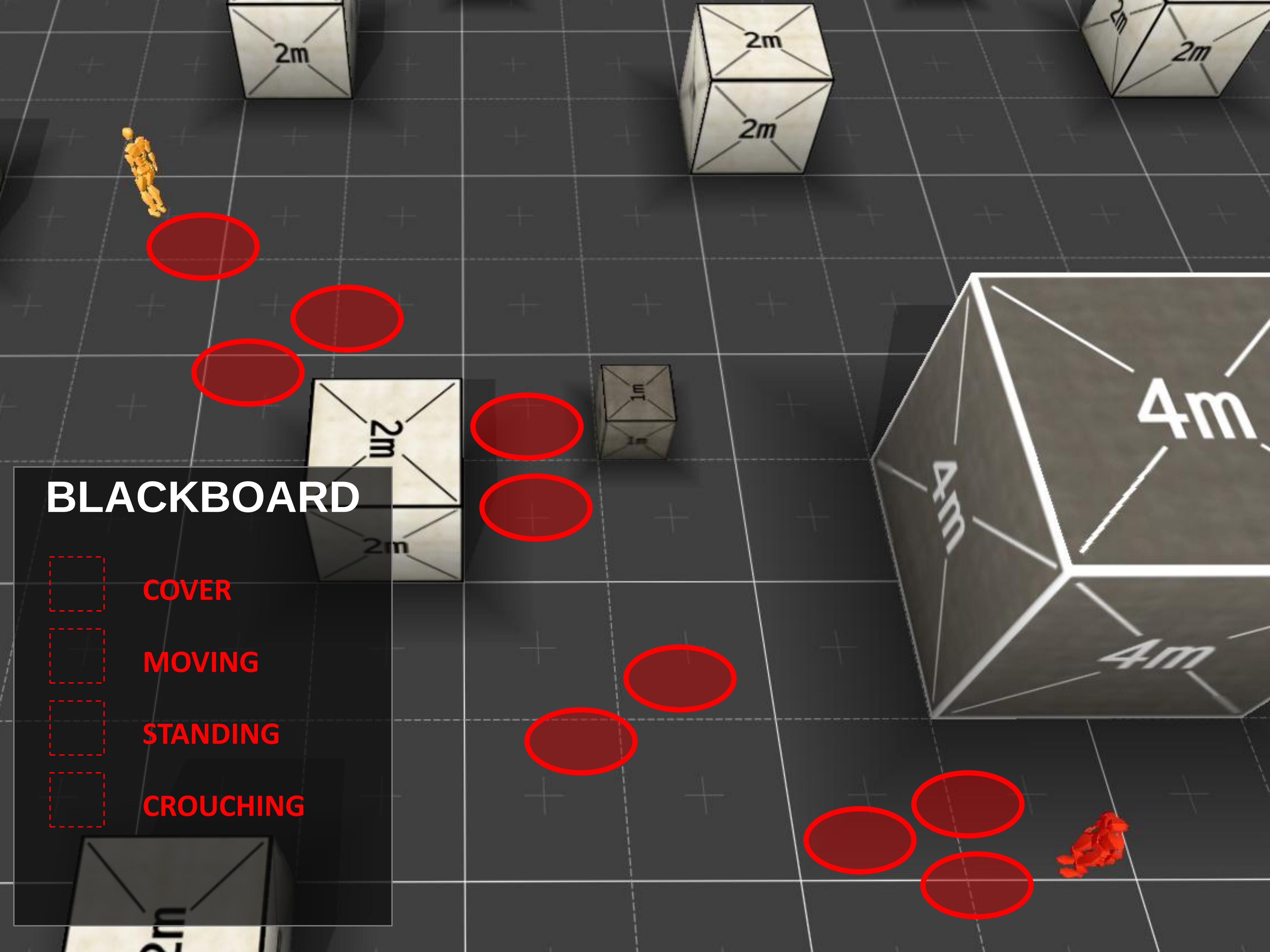
□ COVER

□ MOVING

□ STANDING

□ CROUCHING


2m
2m
2m
2m
2m
1m
1m
2m
2m
4m
4m
4m
SneakToCover
2m



BLACKBOARD

☐ COVER

☐ MOVING

☐ STANDING

☐ CROUCHING

BT DESIGN TAKE AWAY

- Decouple your BT from the problem at hand, for example using a blackboard.

- Build purposeful behaviors as sequences of short goal-directed actions.

- Be careful with “state-like” behaviors that keep running.

- Leverage the power of the tree search!

Behavior Trees Part 3, Michael

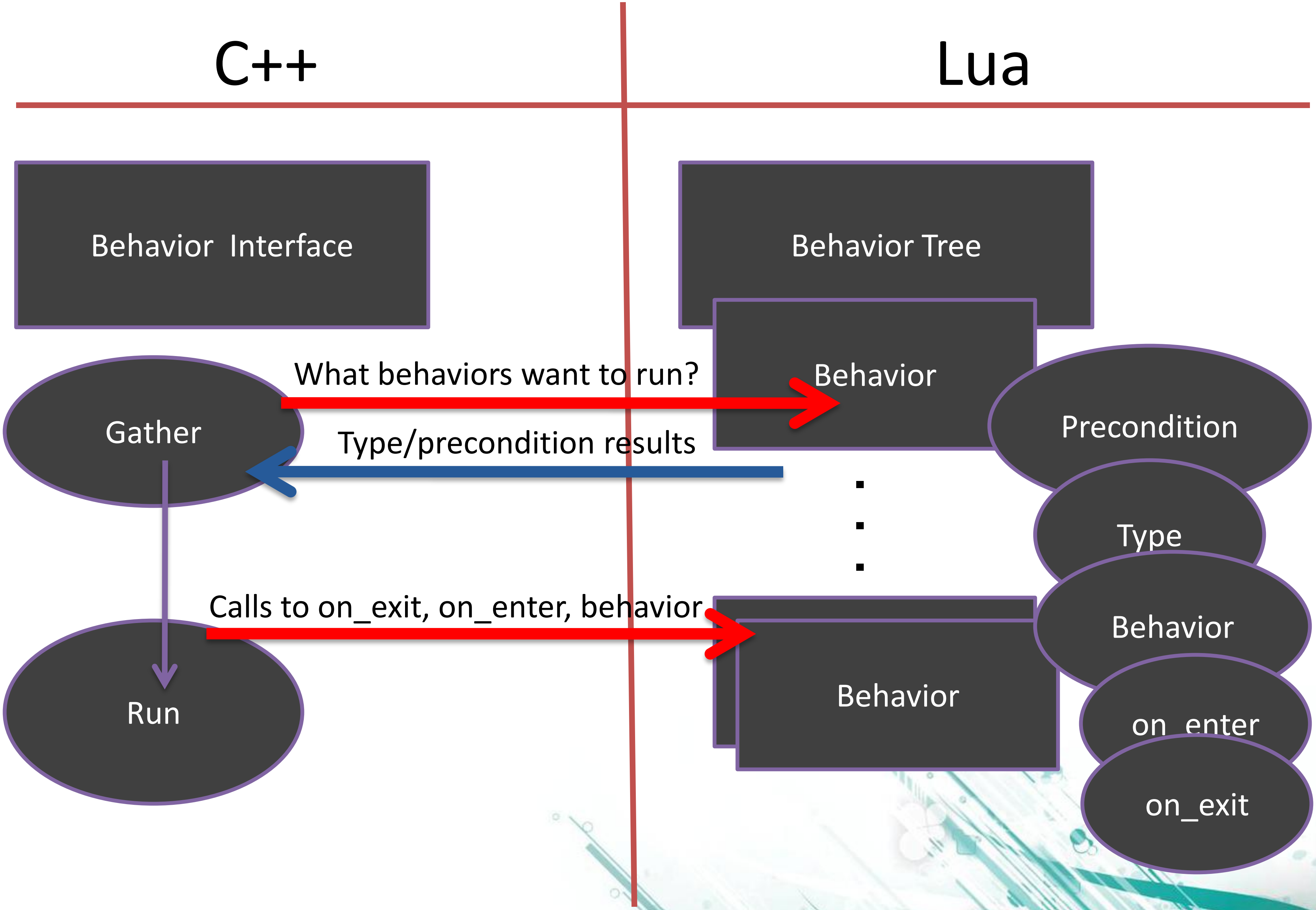
SCRIPT INTEGRATION

SCRIPT INTEGRATION

- Behavior trees are all about flexibility
  - Selector choice!
  - Reuseable goals!
- Rapid iteration is a key goal
  - Bigger games, more actors, individualized behaviors
  - Need to quickly change in response to prototyping and playtest
- Separate the algorithm from the behaviors

FLEXIBILITY FROM SCRIPT

- Lua at BHG, but could be any language
- What you'll need:
  - Scripting language integration
    - Calling script from code and vice versa
- Really nice to have:
  - Designers comfortable with scripting
    - You will need support time (more on that later)
  - Script debugging



C++

Lua

Behavior Interface

Behavior Tree

What behaviors want to run?

Behavior

Gather

Precondition

Type/precondition results

Type

Behavior

Calls to on_exit, on_enter, behavior

Behavior

on_enter

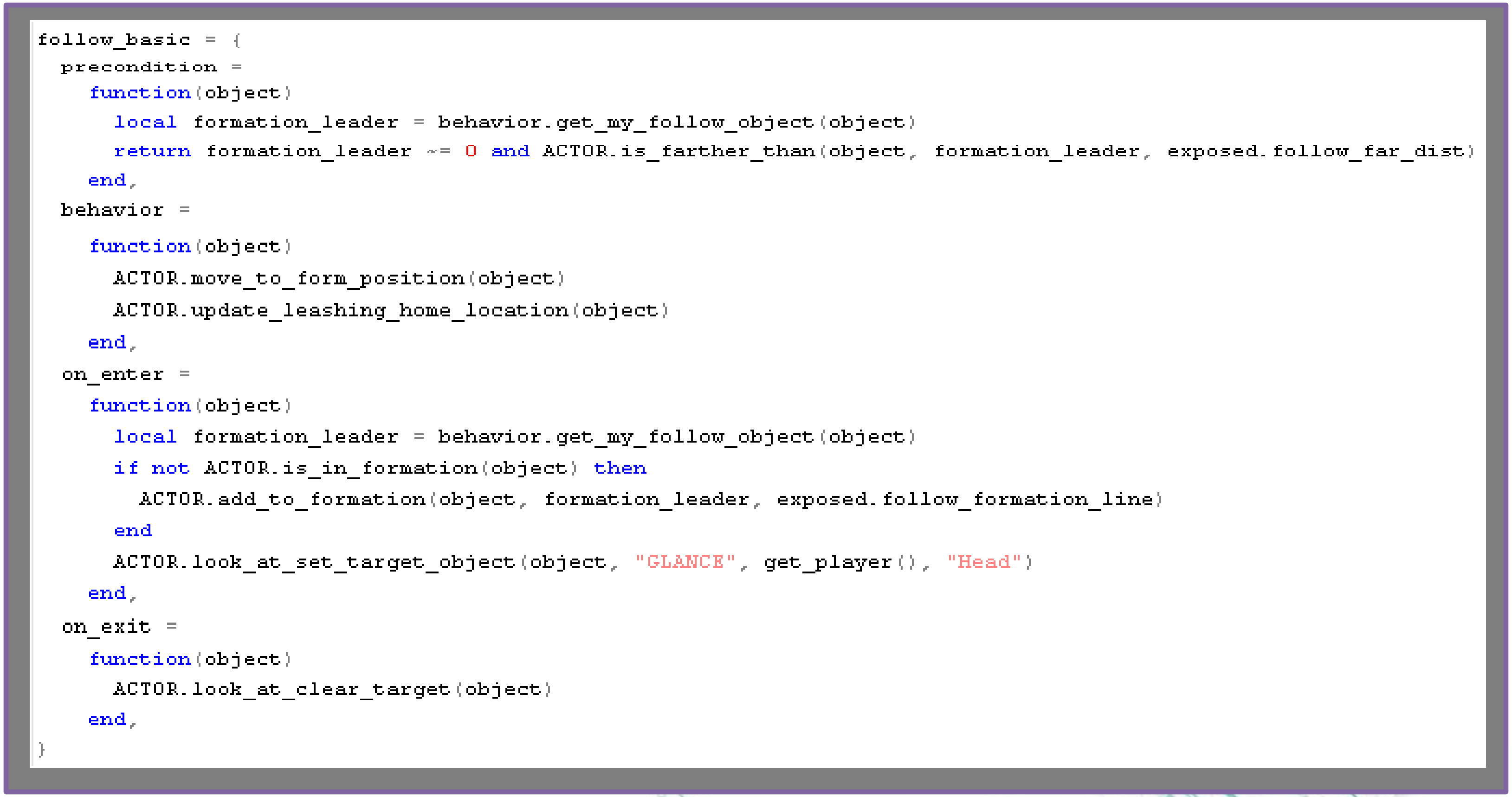
Run

on_exit

CREATING A BEHAVIOR SCRIPT

- Behaviors have a common structure
  - Precondition
  - Behavior
- Optional components
  - Type (priority, sequential, random)
  - on_enter, on_exit
  - Whatever else you decide your behaviors need
- In Lua, these can be known function names in a table

BEHAVIOR SCRIPT



```
follow_basic = {
  precondition =
    function(object)
      local formation_leader = behavior.get_my_follow_object(object)
      return formation_leader ~= 0 and ACTOR.is_farther_than(object, formation_leader, exposed.follow_far_dist)
    end,
  behavior =
    function(object)
      ACTOR.move_to_form_position(object)
      ACTOR.update_leashing_home_location(object)
    end,
  on_enter =
    function(object)
      local formation_leader = behavior.get_my_follow_object(object)
      if not ACTOR.is_in_formation(object) then
        ACTOR.add_to_formation(object, formation_leader, exposed.follow_formation_line)
      end
      ACTOR.look_at_set_target_object(object, "GLANCE", get_player(), "Head")
    end,
  on_exit =
    function(object)
      ACTOR.look_at_clear_target(object)
    end,
}
```

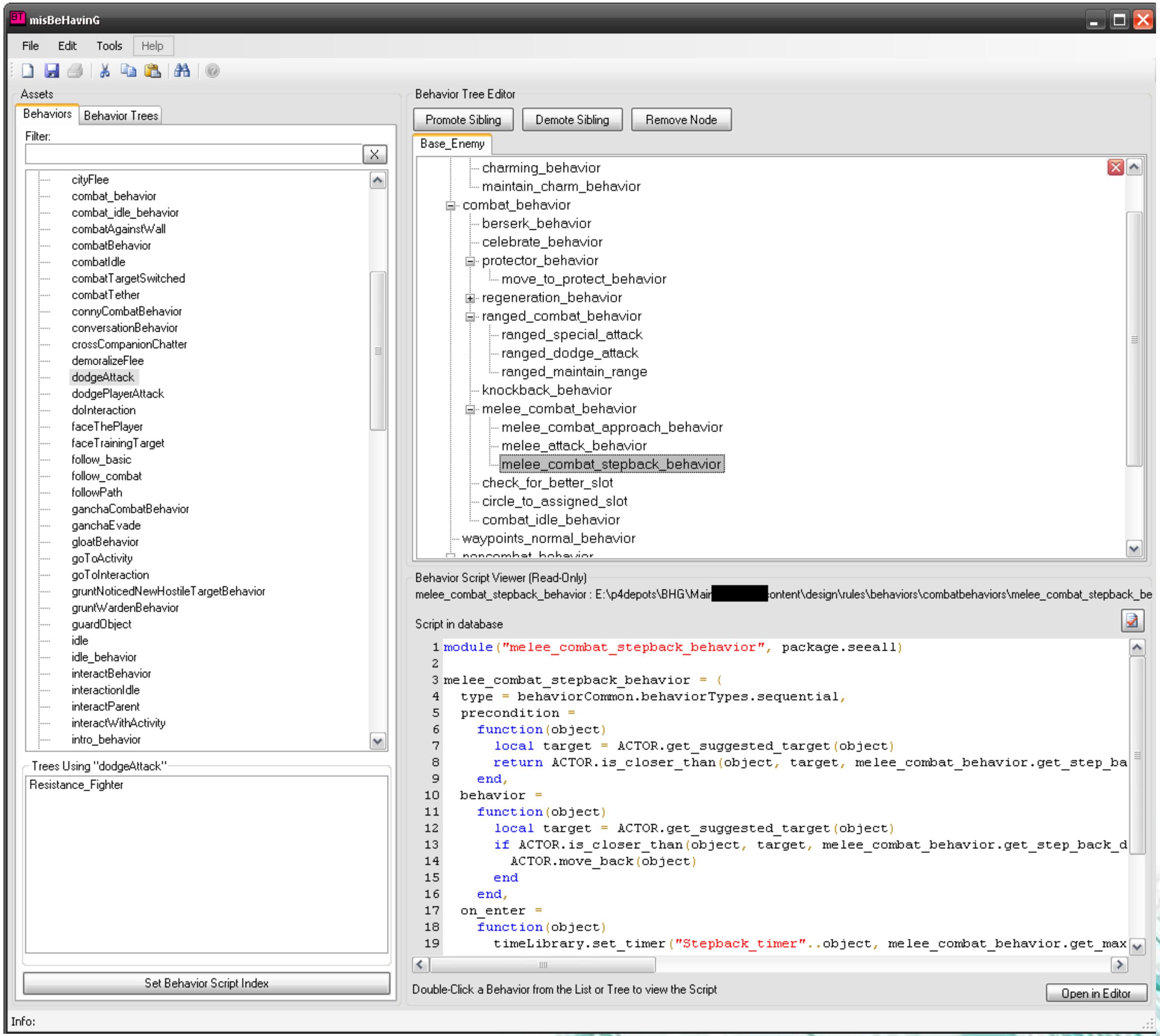
CREATING A TREE WITH BEHAVIOR SCRIPTS

- First implementation: Scripts that create behavior trees
- Lua functions to add, remove, insert behaviors from an existing tree
  - add_behavior(tree, behavior)
- Great flexibility, but hard to conceptualize
  - Creating trees in script was difficult to grasp
  - Especially when trying to reuse trees you didn't write

BEHAVIOR TREE TOOL

- External .NET app to manage trees and behaviors
- Easy to create new behaviors or reuse existing ones
- Statistics on commonly used behaviors
- Search for behavior/tree by name or usage

BEHAVIOR TREE TOOL



cityFlee
combat_behavior
combat_idle_behavior
combatAgainstWall
combatBehavior
combatIdle
combatTargetSwitched
combatTether
connyCombatBehavior
conversationBehavior
crossCompanionChatter
demoralizeFlee
dodgeAttack
dodgePlayerAttack
doInteraction
faceThePlayer
faceTrainingTarget
follow_basic
follow_combat
followPath
ganchaCombatBehavior
ganchaEvade
gloatBehavior
goToActivity
goToInteraction
gruntNoticedNewHostileTargetBehavior
gruntWardenBehavior
guardObject
idle
idle_behavior
interactBehavior
interactionIdle
interactParent
interactWithActivity
intro_behavior

Trees Using "dodgeAttack"

Resistance_Fighter

```
    charming_behavior
    maintain_charm_behavior
combat_behavior
    berserk_behavior
    celebrate_behavior
    protector_behavior
        move_to_protect_behavior
    regeneration_behavior
    ranged_combat_behavior
        ranged_special_attack
        ranged_dodge_attack
        ranged_maintain_range
    knockback_behavior
    melee_combat_behavior
        melee_combat_approach_behavior
        melee_attack_behavior
        melee_combat_stepback_behavior
    check_for_better_slot
    circle_to_assigned_slot
    combat_idle_behavior
waypoints_normal_behavior
noncombat_behavior
```

Behavior Script Viewer (Read-Only)

melee_combat_stepback_behavior : E:\p4depots\BHG\Mair[redacted]ontent\design\rules\behaviors\combatbehaviors\melee_combat_stepback_be

```
module("melee_combat_stepback_behavior", package.seeall)

melee_combat_stepback_behavior = {
  type = behaviorCommon.behaviorTypes.sequential,
  precondition =
    function(object)
      local target = ACTOR.get_suggested_target(object)
      return ACTOR.is_closer_than(object, target, melee_combat_behavior.get_step_ba
    end,
  behavior =
    function(object)
      local target = ACTOR.get_suggested_target(object)
      if ACTOR.is_closer_than(object, target, melee_combat_behavior.get_step_back_d
        ACTOR.move_back(object)
      end
    end,
  on_enter =
    function(object)
      timeLibrary.set_timer("Stepback_timer"..object, melee_combat_behavior.get_max
```

Double-Click a Behavior from the List or Tree to view the Script

BENEFITS OF USING SCRIPT

- Designers write behaviors so you don't have to
  - Currently 63 unique behaviors in our game
    - I wrote 7
  - Lots of time back for other tasks
- Faster implementation and iteration
  - No rebuilding code
  - Can reload scripts while the game runs
    - Need prep for this; flush behaviors, cached names, pointers?

COMMON QUESTIONS

- Performance-related
  - “Isn’t scripting slow?”
  - “How do you stay under CPU budget?”
- Behavior creation-related
  - “Are designers scripting well?”
  - “What if my designers aren’t scripters?”

KEEPING SCRIPT FAST

- Don’t let it be slow!
  - BHG limits lua to integer math
  - Prevent mid-frame garbage collection
- Limit scripting to where it makes sense
  - AI loop is not in script
  - No trig in script!
  - Anything “complicated enough” done in code
- Could put behaviors to code for performance
  - …but maybe not

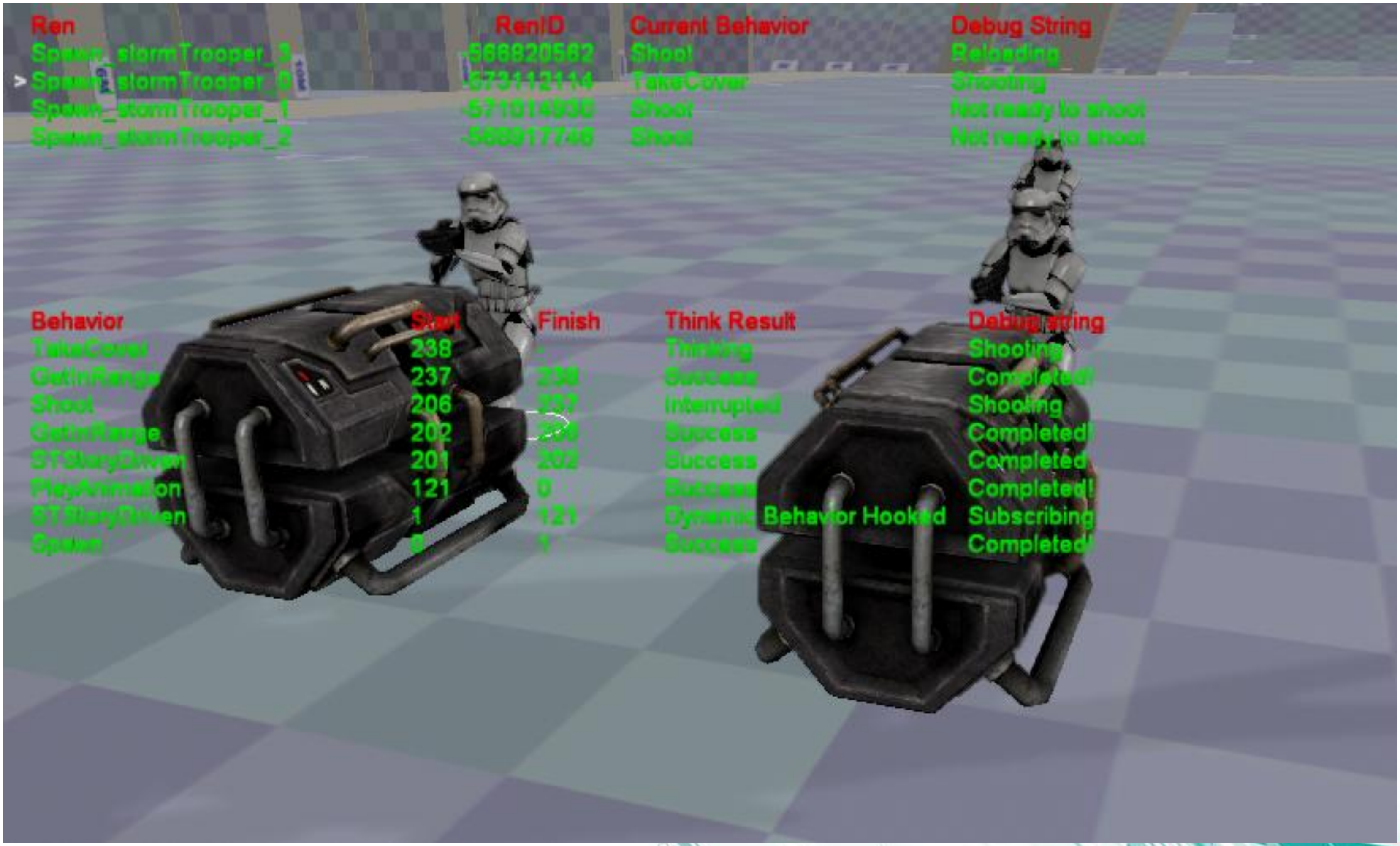
DESIGNER SCRIPTING

- Good enough is great!
- Does take code support time
  - ~10% of my time debugging for designers
  - ~10% on function requests (trig, &c.)
- Watch for things that should be done in code
- Strength in speed! Don’t stifle creativity
- Plan on reviewing trees and behaviors periodically

Behavior Trees Part 4, David

DEBUGGING

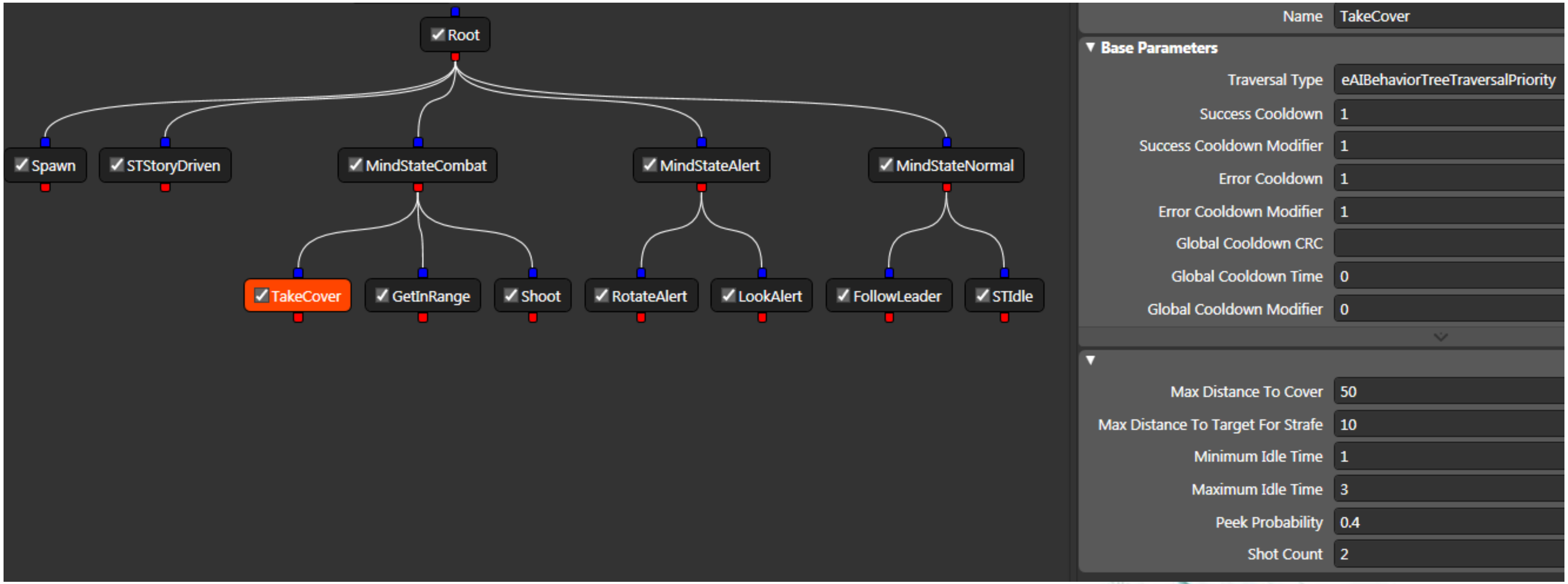
DEBUGGING



Ren	RenID	Current Behavior	Debug String
Spawn_stormTrooper_3	-566820582	Shoot	Reloading
> Spawn_stormTrooper_0	-573112114	TakeCover	Shooting
Spawn_stormTrooper_1	-571014930	Shoot	Not ready to shoot
Spawn_stormTrooper_2	-568917746	Shoot	Not ready to shoot

Behavior	Start	Finish	Think Result	Debug string
TakeCover	238	-	Thinking	Shooting
GetInRange	237	238	Success	Completed!
Shoot	206	237	Interrupted	Shooting
GetInRange	202	206	Success	Completed!
STStoryDriven	201	202	Success	Completed
PlayAnimation	121	0	Success	Completed!
STStoryDriven	1	121	Dynamic Behavior Hooked	Subscribing
Spawn	0	1	Success	Completed!

DEBUGGING



Name	TakeCover
▼ **Base Parameters**	
Traversal Type	eAIBehaviorTreeTraversalPriority
Success Cooldown	1
Success Cooldown Modifier	1
Error Cooldown	1
Error Cooldown Modifier	1
Global Cooldown CRC	
Global Cooldown Time	0
Global Cooldown Modifier	0
▼	
Max Distance To Cover	50
Max Distance To Target For Strafe	10
Minimum Idle Time	1
Maximum Idle Time	3
Peek Probability	0.4
Shot Count	2

Alex J. Champandard AiGameDev.com

Michael Dawe Big Huge Games

David Hernandez-Cerpa LucasArts

Behavior Trees:
Three Ways of Cultivating Game AI

AI Summit