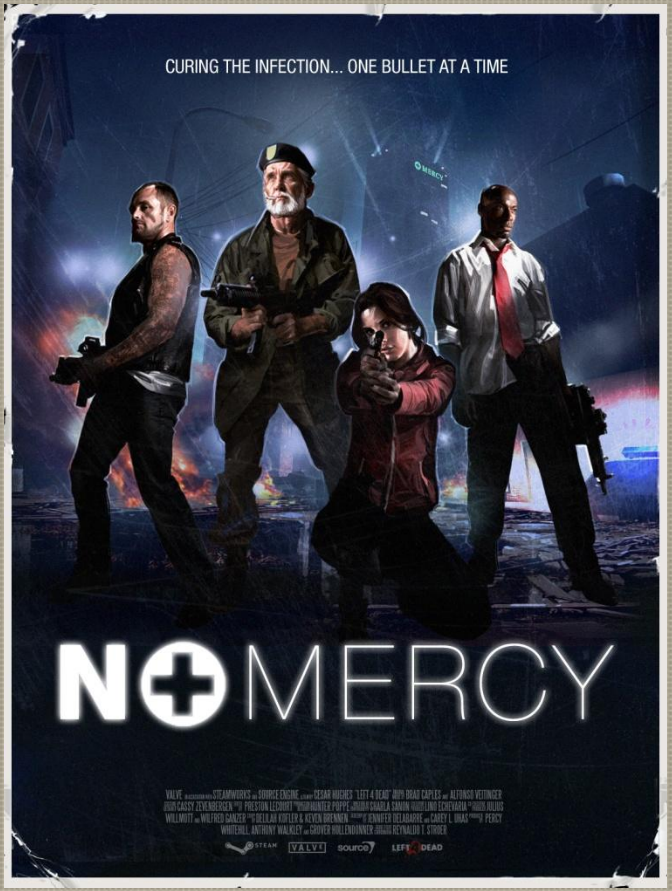
# The AI Systems of Left 4 Dead

**Michael Booth, Valve**

## What is Left 4 Dead?

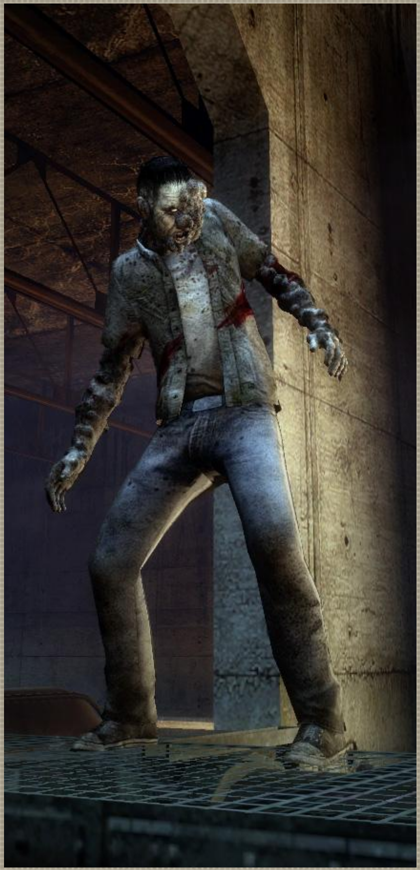
- Left 4 Dead is a replayable, cooperative, survival-horror game where four Survivors cooperate to escape environments swarming with murderously enraged “Infected” (ie: zombies)



## Left 4 Dead: The Survivor Team

## Left 4 Dead: Enraged Infected Mob

# Left 4 Dead: The Special Infected



VALVE

## Left 4 Dead: The Boss Infected

## Left 4 Dead

## Left 4 Dead

## Left 4 Dead

## Goals of Left 4 Dead AI

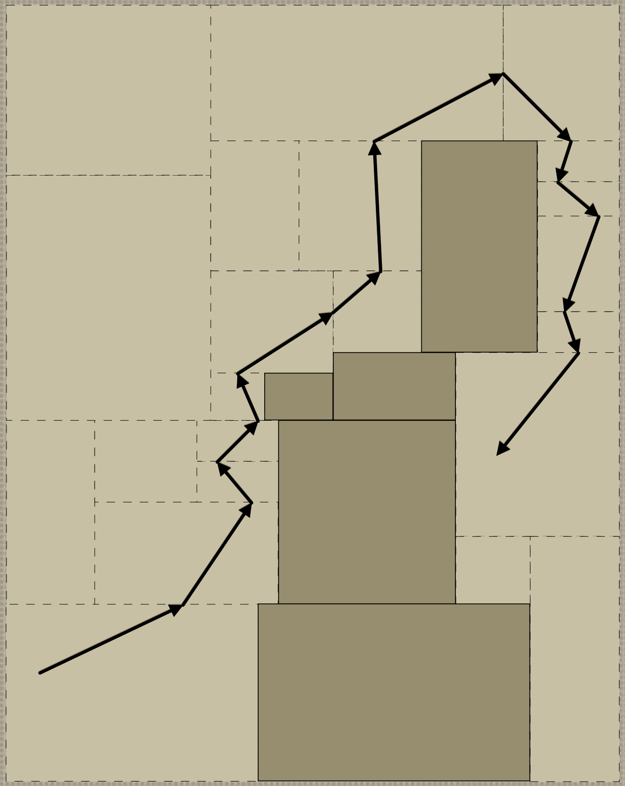
- Deliver Robust Behavior Performances
- Provide Competent Human Player Proxies
- Promote Replayability
- Generate Dramatic Game Pacing

## Goals of Left 4 Dead AI

- **Deliver Robust Behavior Performances**
- Provide Competent Human Player Proxies
- Promote Replayability
- Generate Dramatic Game Pacing

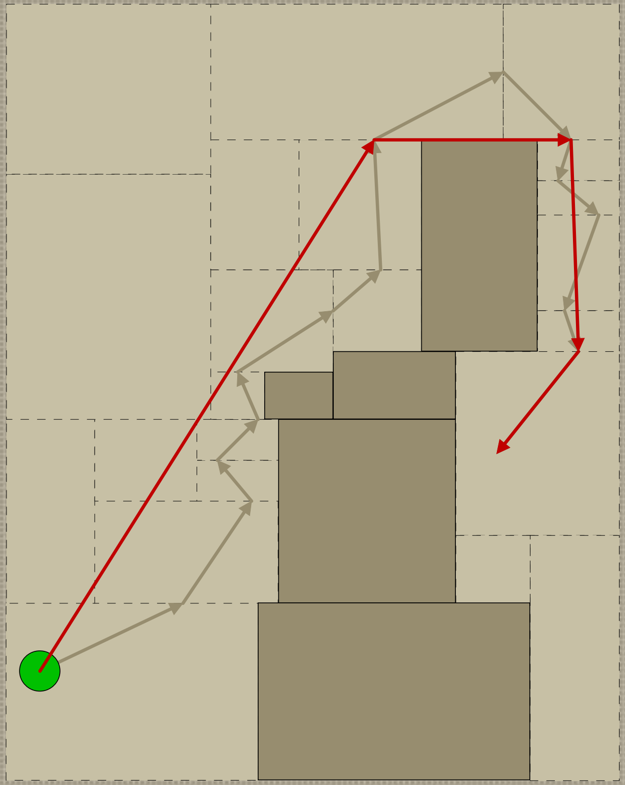
## Goal: Deliver Robust Behavior Performances

- Moving through the Environment
  - A* through Navigation Mesh
  - Creates jagged path
  - How to move fluidly?



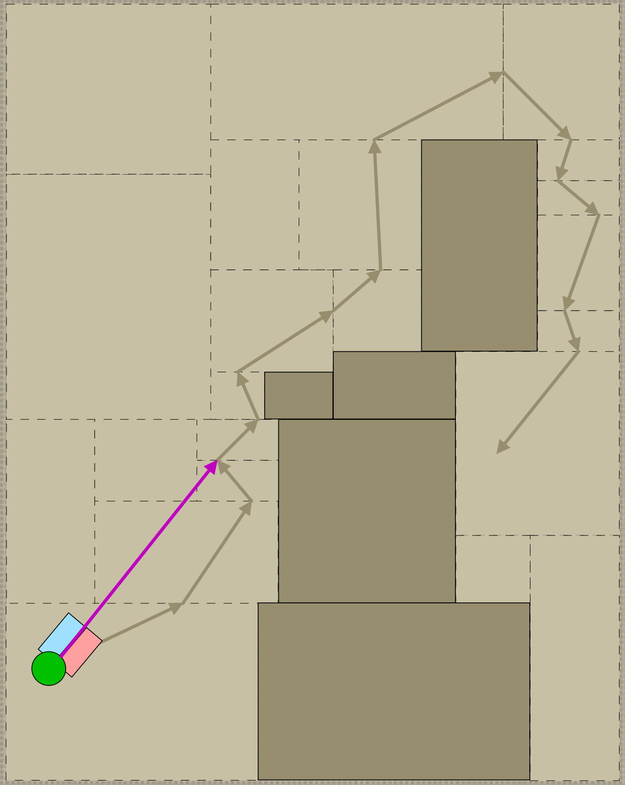
## Goal: Deliver Robust Behavior Performances

- Path Optimization
  - Collapse redundant path nodes
  - Good
    - Creates minimal and direct path
  - Bad
    - Nontrivial CPU cost at path build time
    - Paths are recomputed often in the game
    - Following path directly looks robotic



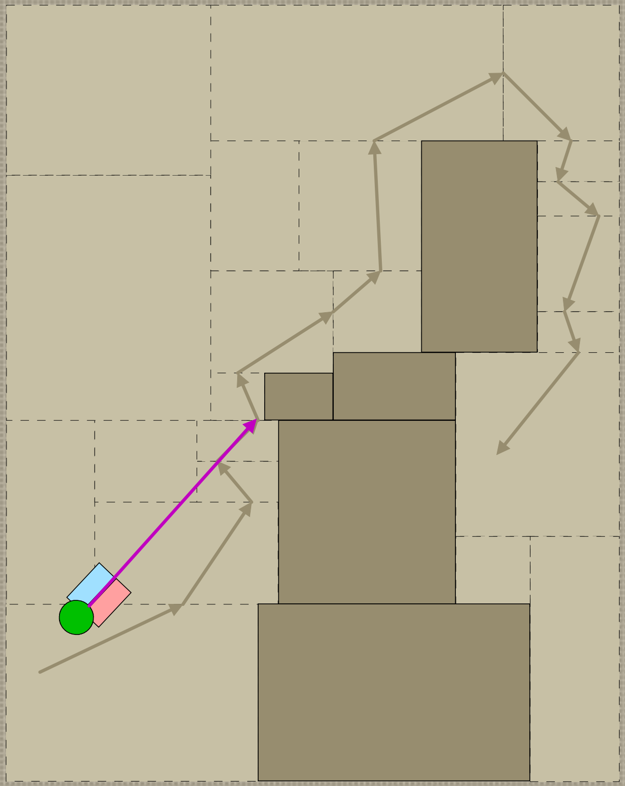
## Goal: Deliver Robust Behavior Performances

- Reactive Path Following
  - Move towards "look ahead" point farther down path
  - Use local obstacle avoidance
  - Good
    - (Re)pathing is cheap
    - Local avoidance handles small physics props, other bots, corners, etc
    - Superposes well with mob flocking behavior
    - Resultant motion is fluid
  - Bad
    - Can avoid off path too much, requiring repath



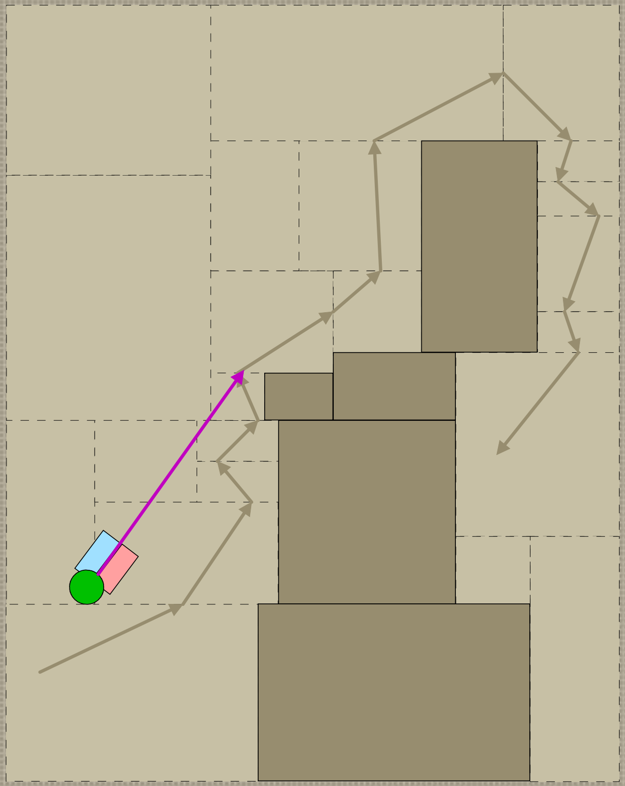
## Goal: Deliver Robust Behavior Performances

- Reactive Path Following



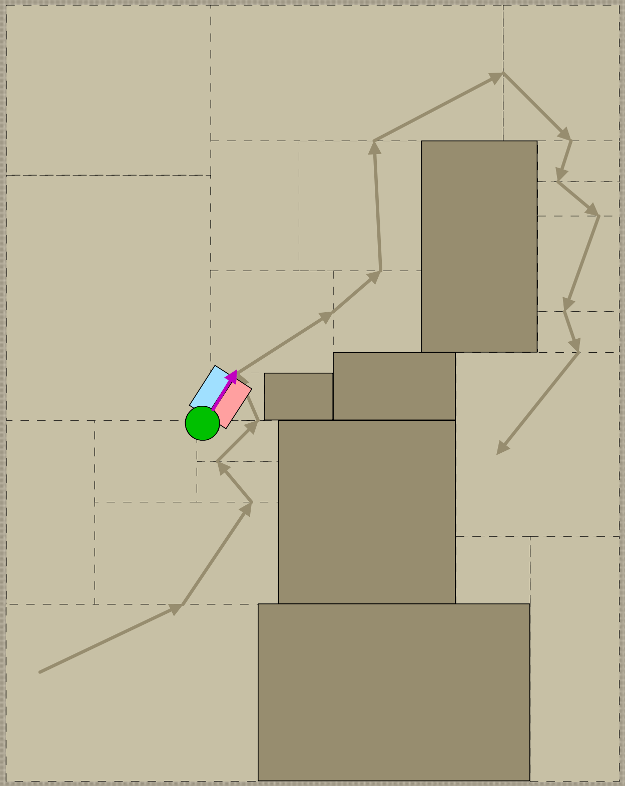
## Goal: Deliver Robust Behavior Performances

- Reactive Path Following



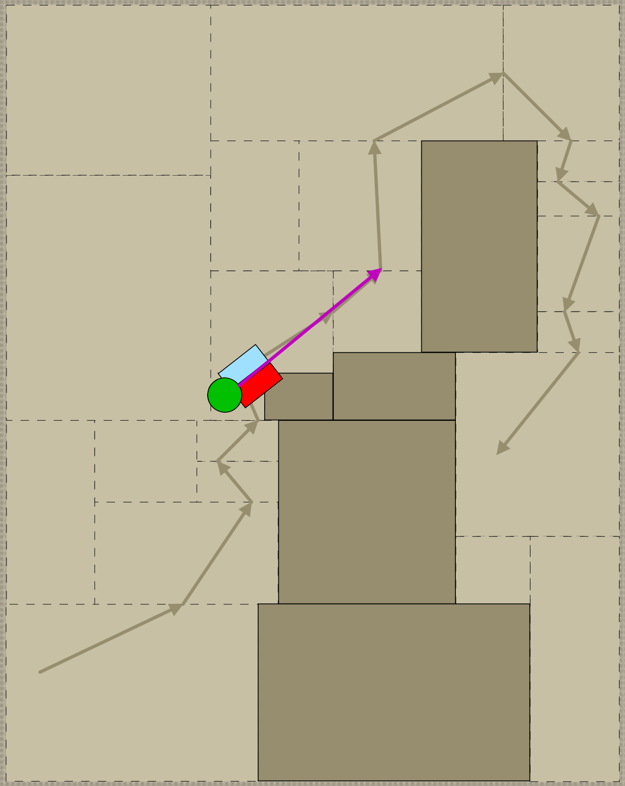
## Goal: Deliver Robust Behavior Performances

- Reactive Path Following



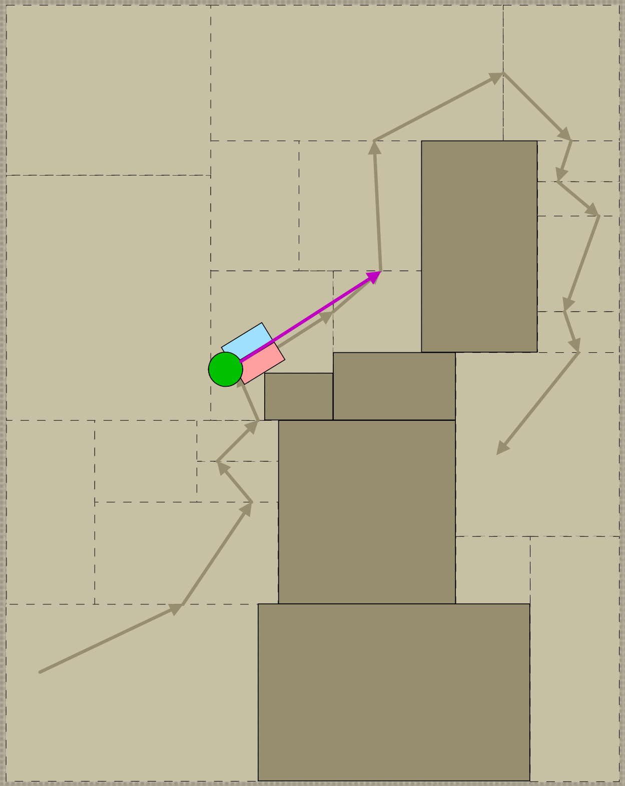
## Goal: Deliver Robust Behavior Performances

- Reactive Path Following



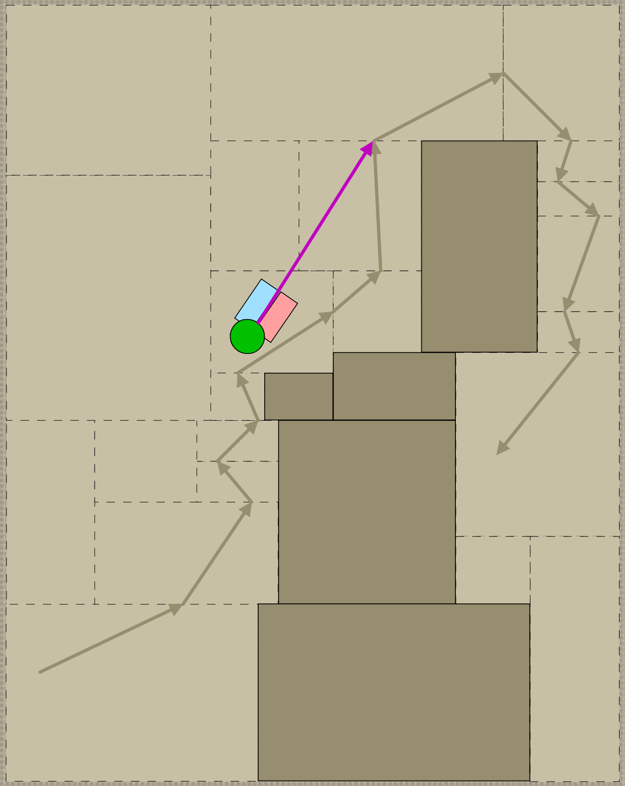
## Goal: Deliver Robust Behavior Performances

- Reactive Path Following



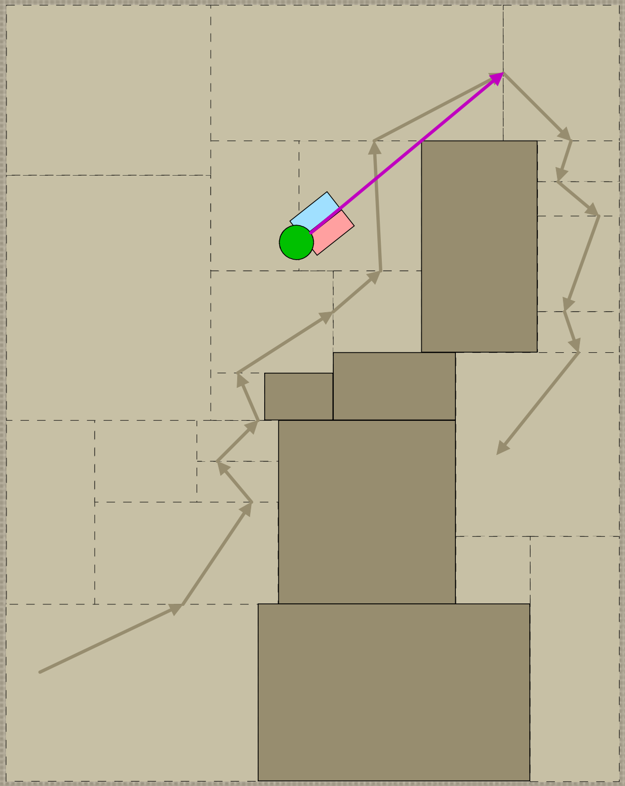
## Goal: Deliver Robust Behavior Performances

- Reactive Path Following



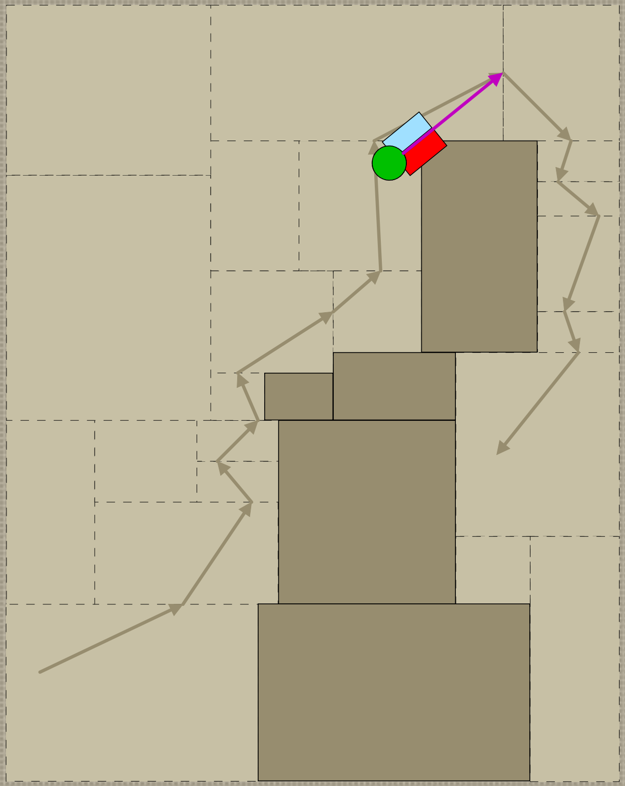
## Goal: Deliver Robust Behavior Performances

- Reactive Path Following



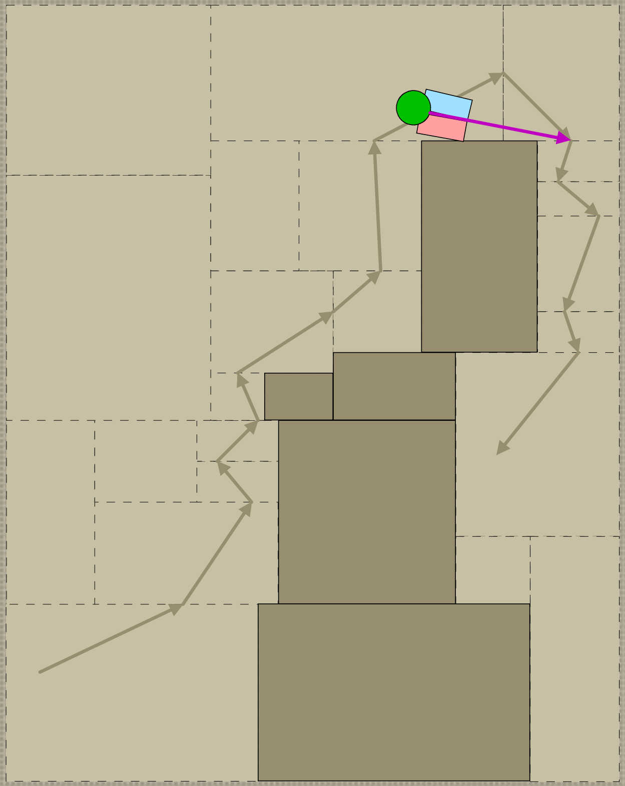
## Goal: Deliver Robust Behavior Performances

- Reactive Path Following



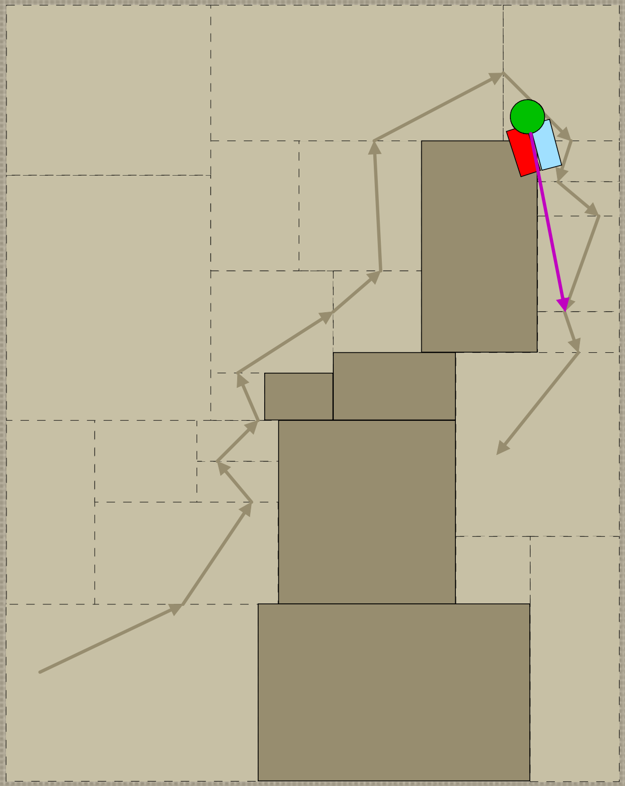
## Goal: Deliver Robust Behavior Performances

- Reactive Path Following



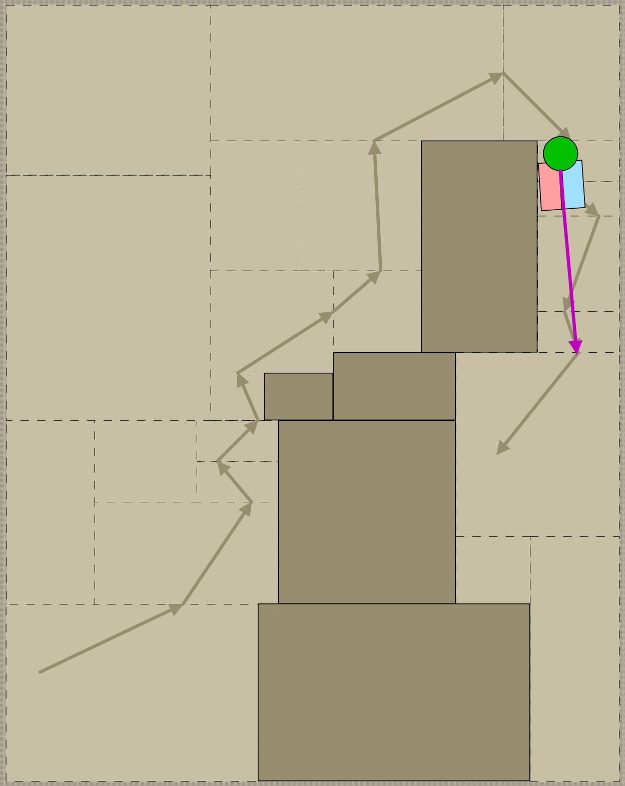
## Goal: Deliver Robust Behavior Performances

- Reactive Path Following



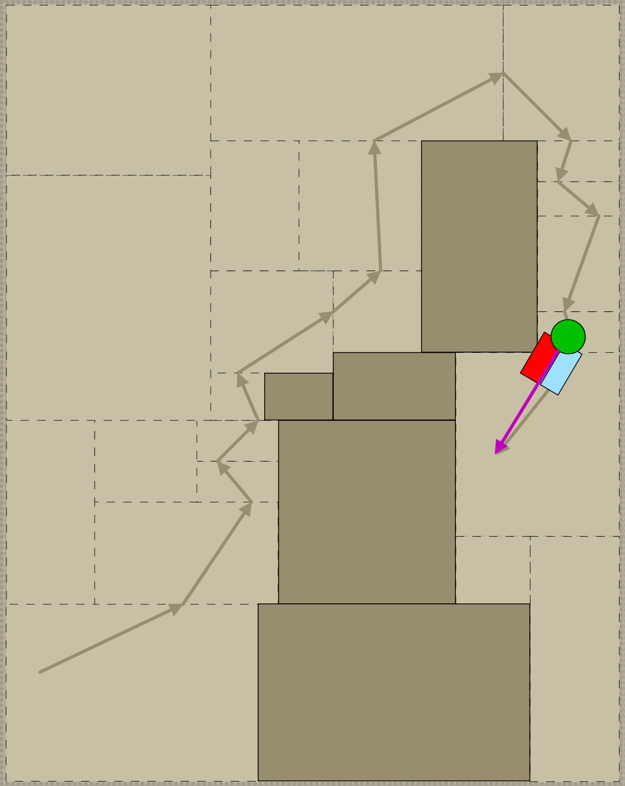
## Goal: Deliver Robust Behavior Performances

- Reactive Path Following



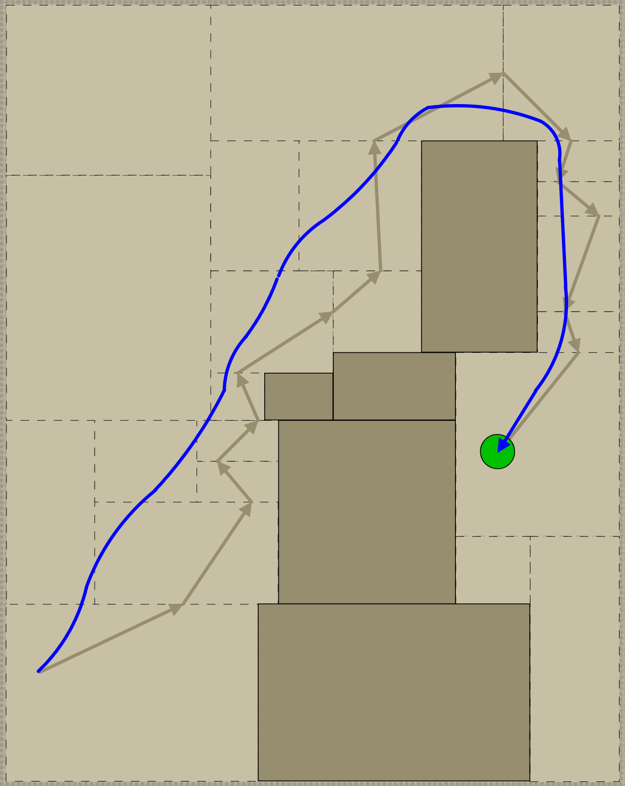
## Goal: Deliver Robust Behavior Performances

- Reactive Path Following



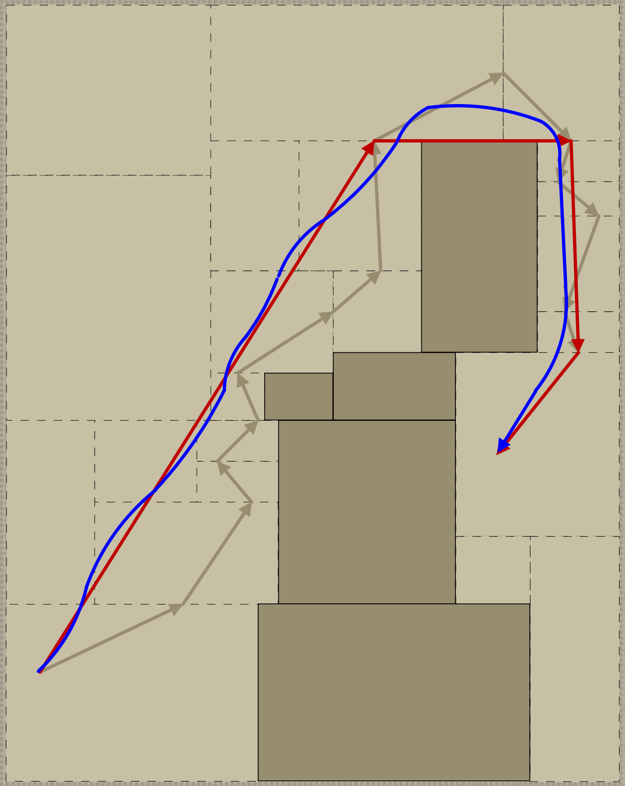
## Goal: Deliver Robust Behavior Performances

- Reactive Path Following



## Goal: Deliver Robust Behavior Performances

- Path Optimization vs Reactive Path Following
  - Resultant path is reasonably close to optimized path
  - Reactive Path Following is used for all actors in Left 4 Dead

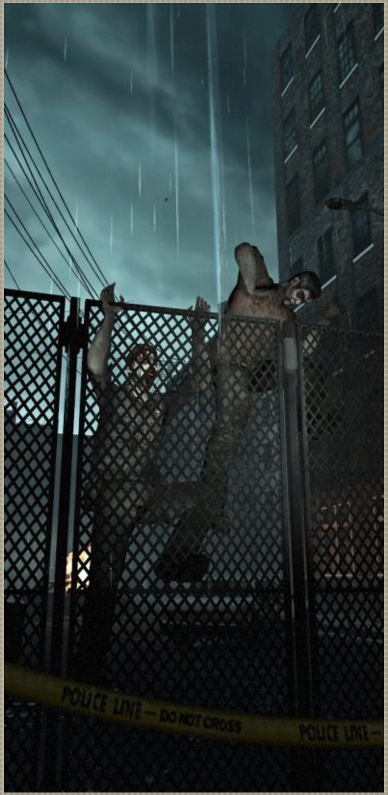


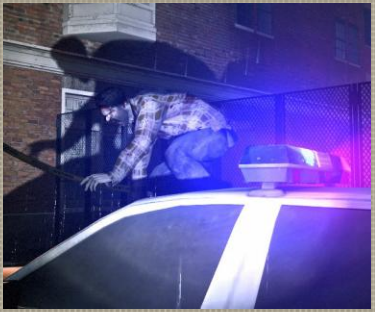
## Goal: Deliver Robust Behavior Performances

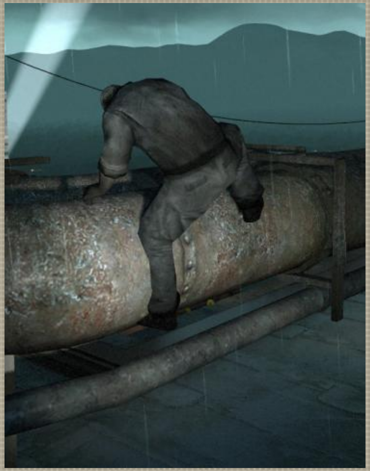
➢ Reactive Path Following: Climbing

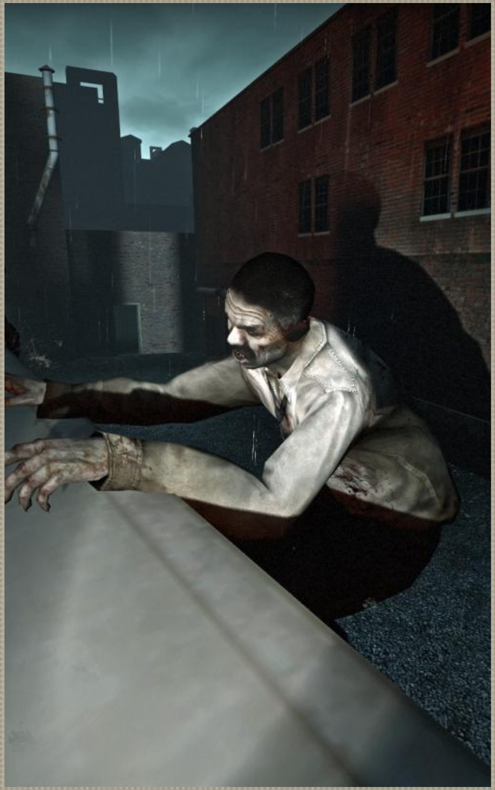
- To ensure the Common Infected horde are always dangerous, they have the ability to rapidly climb
- Climbing is algorithmic, using a similar technique to local obstacle avoidance
- Algorithmic climbing solves two major issues
  - Navigating the complex geometry of the post apocalyptic world of Left 4 Dead
  - Navigating over movable physics props

## Goal: Deliver Robust Behavior Performances



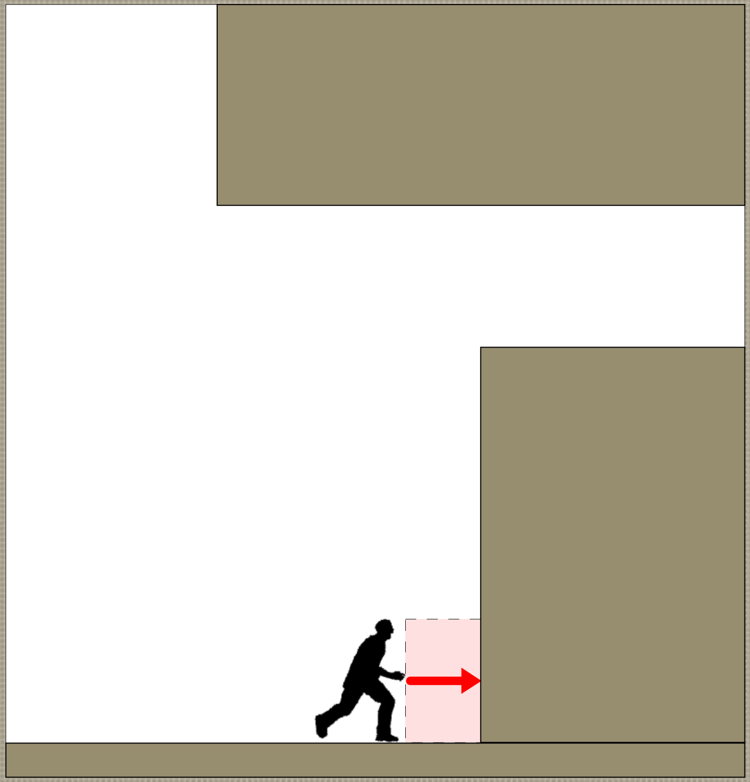






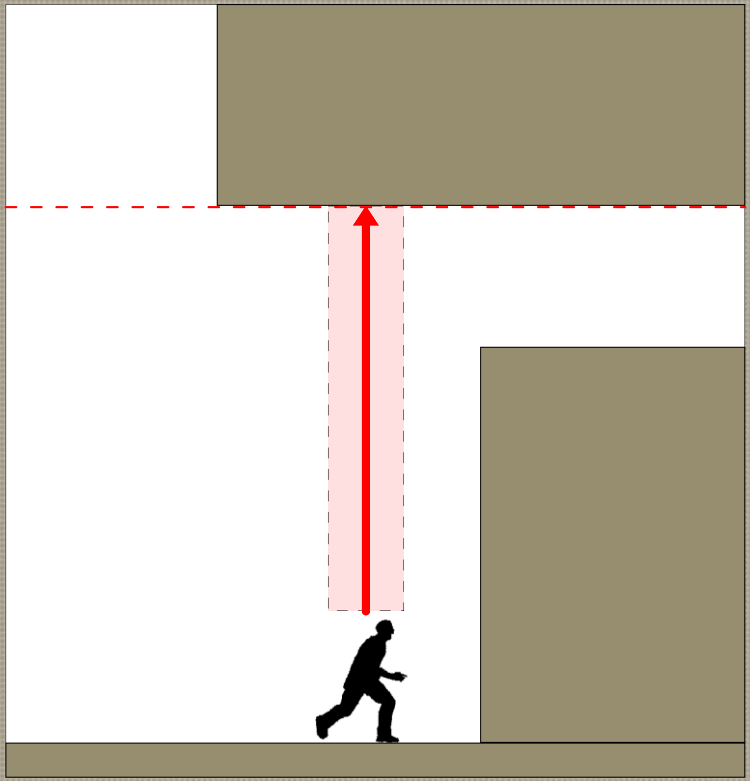
## Goal: Deliver Robust Behavior Performances

- Approach
  - Bot periodically tests for obstacles immediately ahead as it follows its path



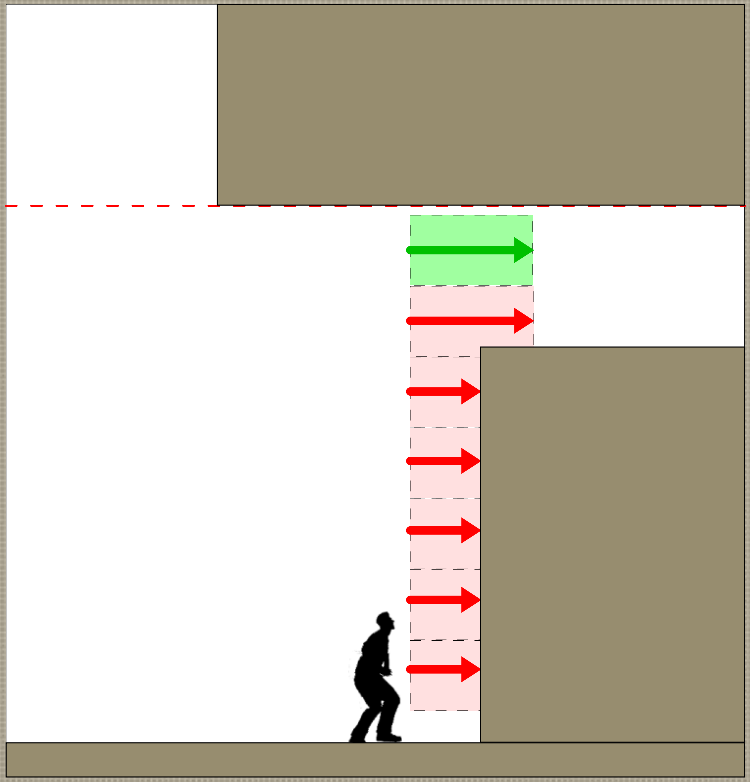
## Goal: Deliver Robust Behavior Performances

- Find Ceiling
  - Once an obstacle has been detected, another hull trace determines the available vertical space above the bot



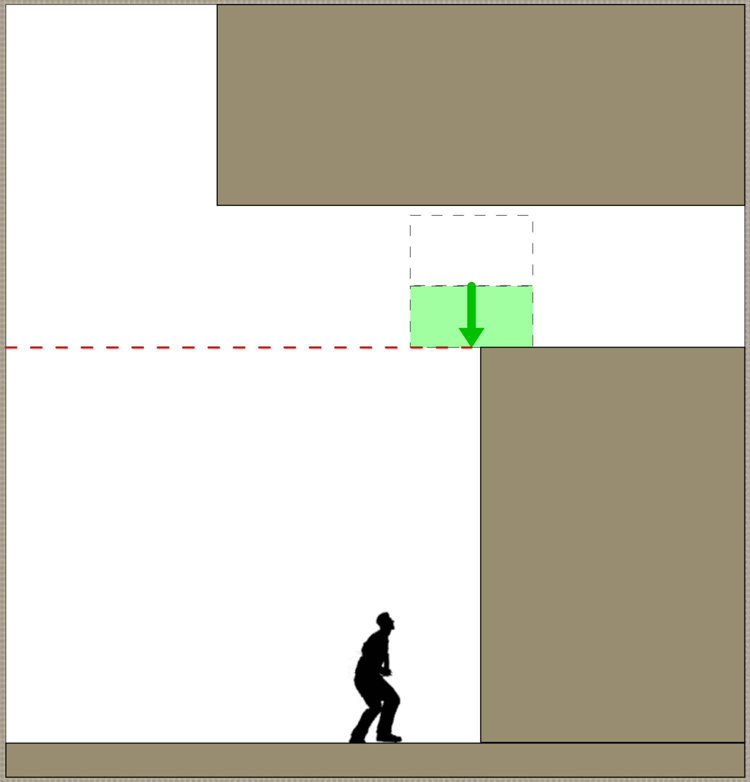
## Goal: Deliver Robust Behavior Performances

- Find Ledge
  - The vertical space is then scanned via a series of hull traces from lowest to highest to find the first unobstructed hull trace



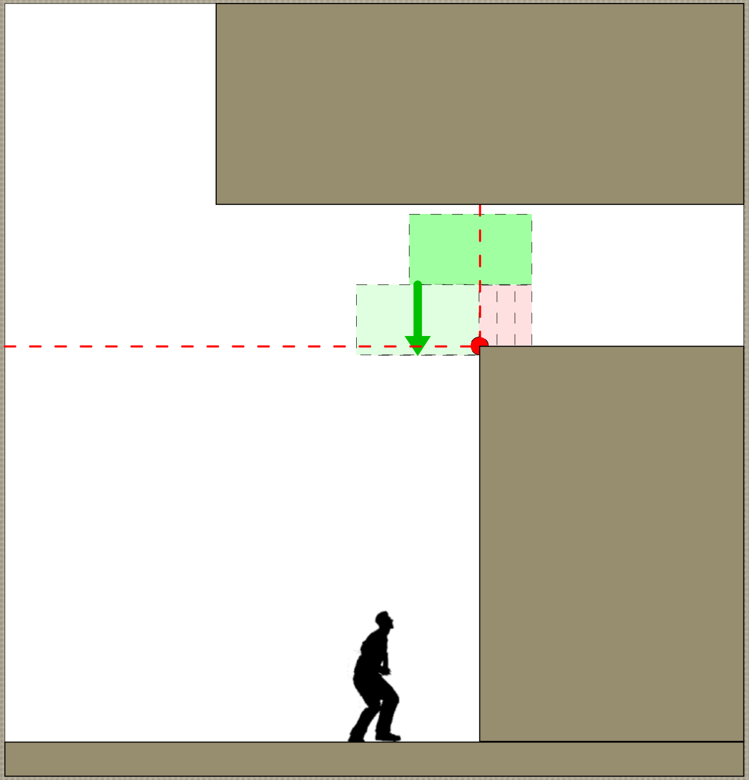
## Goal: Deliver Robust Behavior Performances

- Find Ledge Height
  - Another hull trace downward from the unobstructed trace finds the precise height of the ledge to be climbed



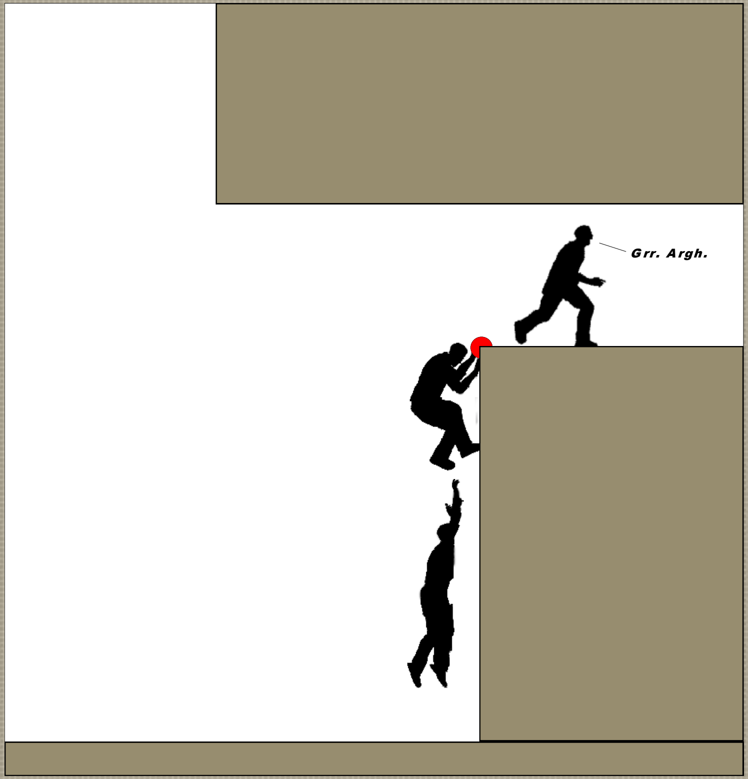
## Goal: Deliver Robust Behavior Performances

- Find Ledge Forward Edge
  - A final series of downward hull traces that step backwards determine the forward edge of the ledge



## Goal: Deliver Robust Behavior Performances

- Perform Climb
  - Pick closest match animation from dozens of motion captured climb animations of various height increments

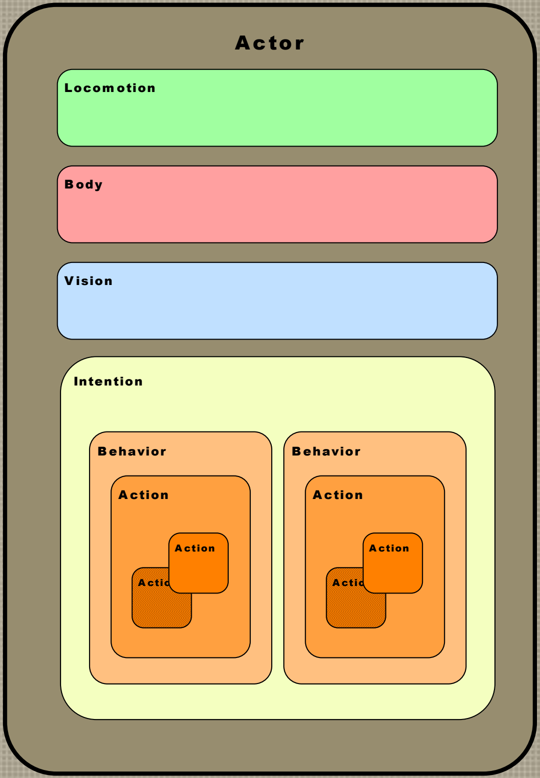


## Goal: Deliver Robust Behavior Performances

- Behaviors and decision making

## Goal: Deliver Robust Behavior Performances

- **Locomotion**
  Owns moving the actor to a new position in the environment (collision resolution, etc)
- **Body**
  Owns animation state
- **Vision**
  Owns line-of-sight, field of view, and "am I able to see <X>" queries. Maintains recognized set of entities.
- **Intention**
  Contains and manages a system of concurrent Behaviors (HFSM+Stack)
  - **Behavior**
    Contains and manages a system of Actions
  - **Action**
    The building blocks of intentional actions. An Action can contain a child Action, which executes concurrently.



## Goal: Deliver Robust Behavior Performances

- Encapsulation of Action processing
  - OnStart
    - Executed when the Action is transtioned into
    - Can return an immediate transition
  - Update
    - Does the "work" of the Action
    - Update can return a transition to a new Action
  - OnEnd
    - Is executed when the Action is transitioned out of
  - OnSuspend
    - Executed when Action has been put on hold for another Action
    - Can return a transition
  - OnResume
    - Executed when Action resumes after being suspended
    - Can return a transition (perhaps Action is no longer valid)

## Goal: Deliver Robust Behavior Performances

### ➢ Action Transitions

- **Continue()**
  No transition, continue this Action next frame.
- **ChangeTo( NextAction, “reason” )**
  Exit the current Action and transition into NextAction.
- **SuspendFor( NextAction, “reason” )**
  Put the current Action “on hold” (bury it) and enter NextAction.
- **Done( “reason” )**
  This Action is finished. Resume the “buried” Action.
- **“reason”**
  A string documenting the reason why the transition occurred (ie: “The player I was chasing died”). Reason strings help clarify the Action system and are invaluable for realtime debug output.

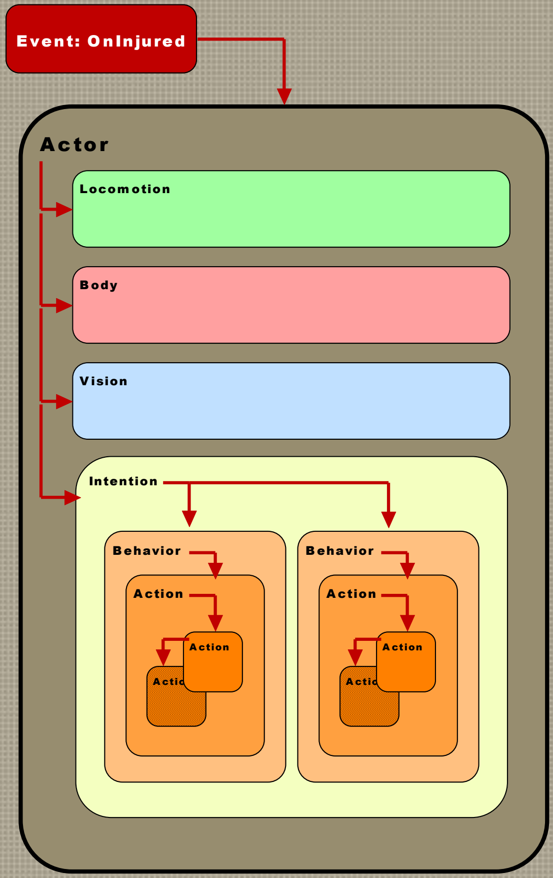
## Goal: Deliver Robust Behavior Performances

### Encapsulation of Action Transitions

- Returning ActionResults enforces **atomic transitions** since no behavior code can be run between the decision to change to a new Action and the actual change
- The **only** way to go from Action A to Action B is for Action A to return a transition to Action B
- There is no way to force a transition from Action A to Action B without Action A's "knowledge"
- Keeps related behavior code encapsulated within the Action

## Goal: Deliver Robust Behavior Performances

- **Event Propagation**
- Events propagate to all components
- In the Intention subsystem, events are
  - First handled by the innermost child Action
  - If child has no response, event is sent to buried Action
  - If no buried Actions have a response, event is sent to parent



## Goal: Deliver Robust Behavior Performances

- Example events
  - OnMoveToSuccess, OnMoveToFailure, OnStuck, OnUnStuck, OnAnimationActivityComplete, OnAnimationEvent, OnContact, OnInjured, OnKilled, OnOtherKilled, OnSight, OnLostSight, OnSound, OnPickUp, OnDrop
- Actions can implement **event handler** methods, providing context-specific reactions to game events

## Goal: Deliver Robust Behavior Performances

- **Contextual Query System**
- Actions can implement query methods, providing context-specific answers to questions
- Defines queries that can be asked of the Behavior/Action system as a “black box”
- Queries propagate from childmost Action outward until an answer is given, identically to Event propagation
- Allows concurrent Behaviors to coordinate
  - ie: We are able to opportunistically able to pick up an object, should we?

## Goal: Deliver Robust Behavior Performances

- Example Queries
  - SelectMoreDangerousThreat( me, subject, threat1, threat2 )
    - Allows bot to find the most dangerous threat to someone else (ie: an incapacitated player it wants to help)
  - ShouldPickUp
  - ShouldHurry
  - IsHindrance

## Goals of Left 4 Dead AI

- Deliver Robust Behavior Performances
- **Provide Competent Human Player Proxies**
- Promote Replayability
- Generate Dramatic Game Pacing

## Goal: Provide Competent Human Player Proxies

- Survivor Bots
  - Allowed us to assume 4 player Survivor team for game tuning and balance
  - Drop in/out ("Take a Break") incredibly valuable in the wild
  - Automated stress testing with 4 SurvivorBots and accelerated game time

## Goal: Provide Competent Human Player Proxies

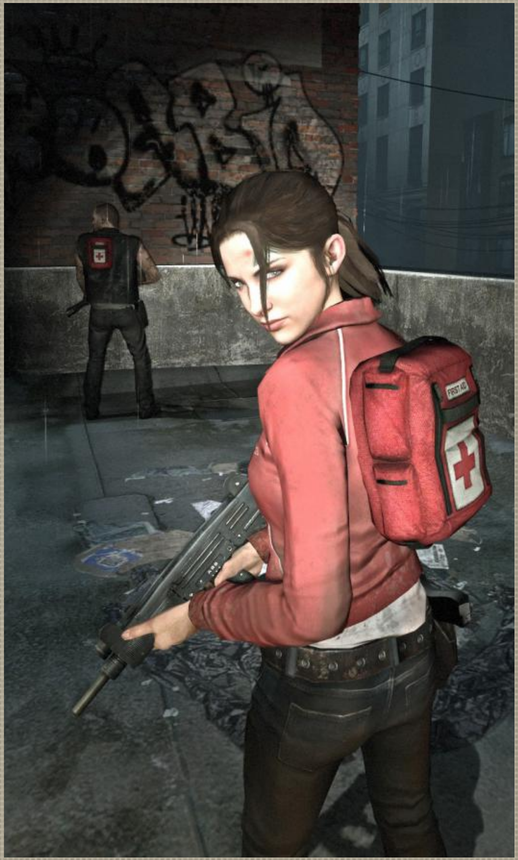
- Believability/Fairness
  - Players need to believe bot replacements are "fair"
  - Imperfect knowledge – simulated senses
  - Simulated aiming
  - Reaction times
  - Reliable and predictable decision making
    - Difficult issue: Rescuing incapacitated humans

## Goal: Provide Competent Human Player Proxies

- Trust
  - Cooperative nature of game requires close collaboration of Survivor team and implicit trust in teammate's action choices
  - SurvivorBots prioritize human teammates over bots
  - Game "cheats" to guarantee some undesirable events cannot occur
    - SurvivorBots cannot deal friendly fire damage
    - SurvivorBots never use Molotovs
    - If a SurvivorBot ever gets far out of place for any reason, it is teleported near the team when no human is looking at it

## Goal: Provide Competent Human Player Proxies

- SurvivorBots exercised complexity encapsulation
  - Two concurrent Behavior systems: Main and Legs
    - Main: Primary decision making, attention, target selection and attack
    - Legs: Slaved to Main, responsible for staying near team unless otherwise directed
  - Many hierarchical behavior states reacting to dozens of game events
    - Interrupting, resuming
  - Complex contextual decision making
    - Tank, Witch, Boomer, Smoker, Hunter, Mob rushes, wandering horde, Survivor teammates, item scavenging, healing self, healing teammates, weapon selection, Rescue Closets, incapacitation, limping, complex 3D terrain, movable physics obstacles, transient fires, ladders, elevators, in any combination
  - Replicating human performance based on actual gameplay experience
    - Behavior system built to reproduce decisions and actions players make while playing the game



## Goals of Left 4 Dead AI

- Deliver Robust Behavior Performances
- Provide Competent Human Player Proxies
- **Promote Replayability**
- Generate Dramatic Game Pacing

## Goal: Promote Replayability

- Replayability promotes long-term engagement with the game, resulting in growth of the game's community
  - Growing online community results in ongoing sales
  - Creates exposure opportunities for other related content
- Examples: Counter-Strike, Team Fortress, Day of Defeat
  - Thriving online communities with ongoing sales revenue
  - Comparatively few maps, played repeatedly for years on end
  - Unpredictable experience created by interactions between teams of players drives replayability

## Goal: Promote Replayability

- In Left 4 Dead, **Procedural Population** of enemies and loot strongly promotes replayability
  - Creates unpredictable encounters analogous to CS, TF, DoD
  - Game session is viewed as a **skill challenge** instead of a **memorization exercise**

## Goal: Promote Replayability

- Static placement of enemies and loot hinders replayability
    - Players are good at memorizing all static locations
    - Removes suspense of not knowing what will happen next
    - Results in “optimal strategy” that works every time
    - Players expect everyone to have memorized all encounters
    - Kills cooperation and degenerates into a race
- Even multiple sets of manually placed triggers/scripts fails
    - Players learn all of them, and if they don’t see A, they prepare for B or C, etc.

## Goal: Promote Replayability
## Procedurally Populated Environments

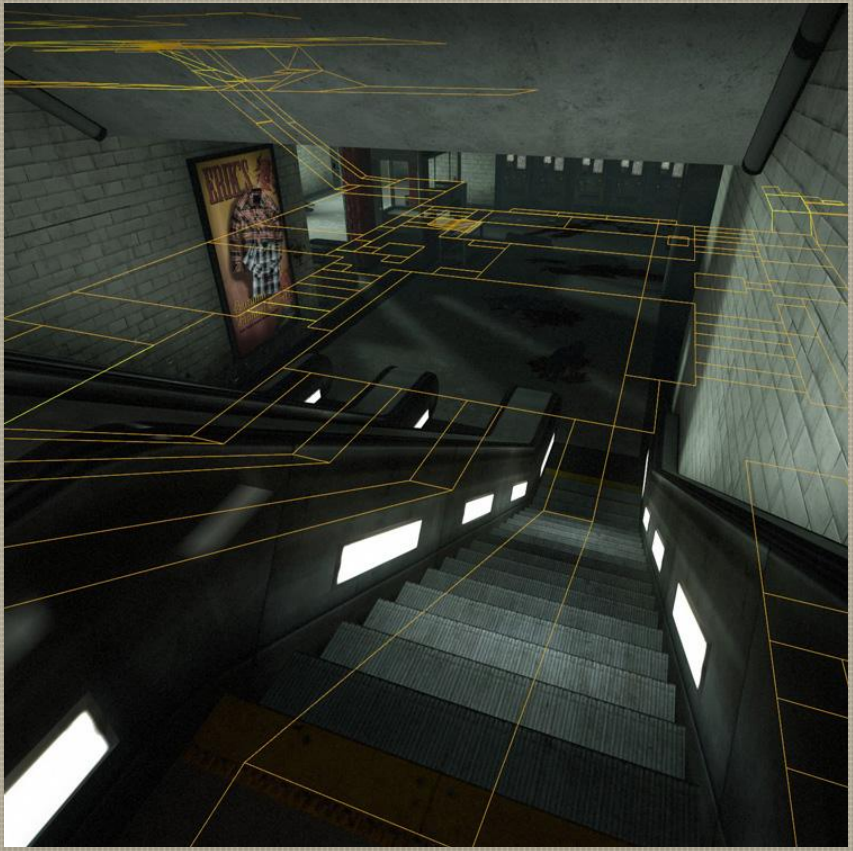
- How do we procedurally populate the environment with interesting distributions of enemies?
  - Using **Structured Unpredictability**
  - Part of the collection of systems called the **“AI Director”**

## Goal: Promote Replayability
## Procedurally Populated Environments

- First, a short summary of tools used to generate Structured Unpredictability
  - Navigation Mesh
  - Flow Distance
  - Potential Visibility
  - Active Area Set

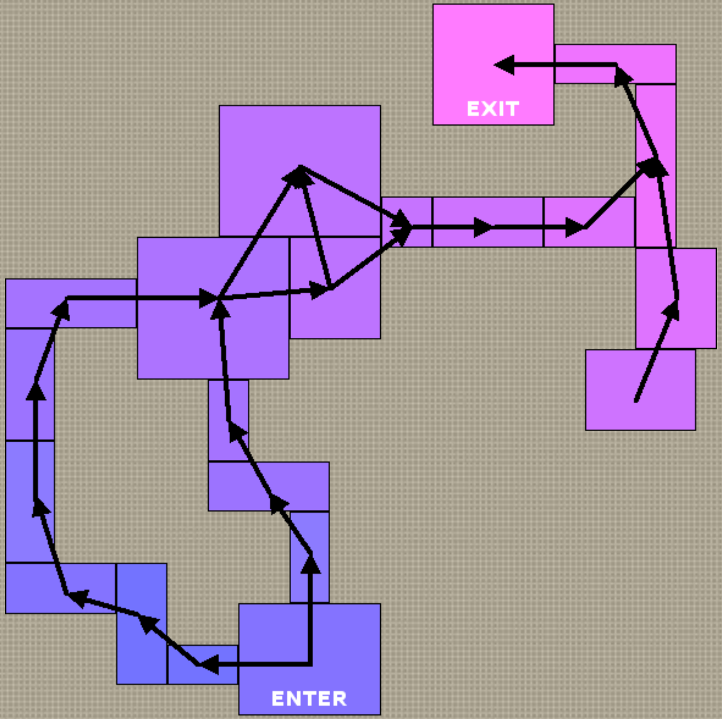
## Goal: Promote Replayability
## Procedurally Populated Environments

- The Navigation Mesh
  - Represents "walkable space"
  - Originally created for Counter-Strike Bot pathfinding
  - Useful for general spatial reasoning and spatially localized information
    - Has area A ever been seen by actor B?
    - Is area X potentially visible by area Y?
    - Where is a spot near the Survivors, not visible to any of them?
    - How far have we traveled to reach this area?



## Goal: Promote Replayability
## Procedurally Populated Environments

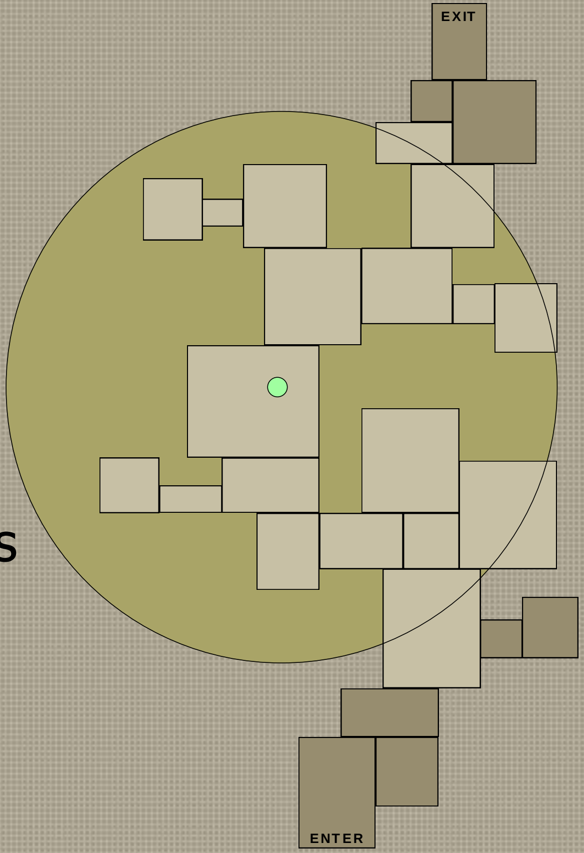
- "Flow Distance"
  - Travel distance from the starting safe room to each area in the navigation mesh
  - Following increasing flow gradient always takes you to the exit room
  - "Escape Route" = shortest path from start safe room to exit
  - Used as a metric for populating enemies and loot, and for answering questions such as "is this spot ahead or behind the Survivor group"



Goal: Promote Replayability
Procedurally Populated Environments

## The Active Area Set (AAS)

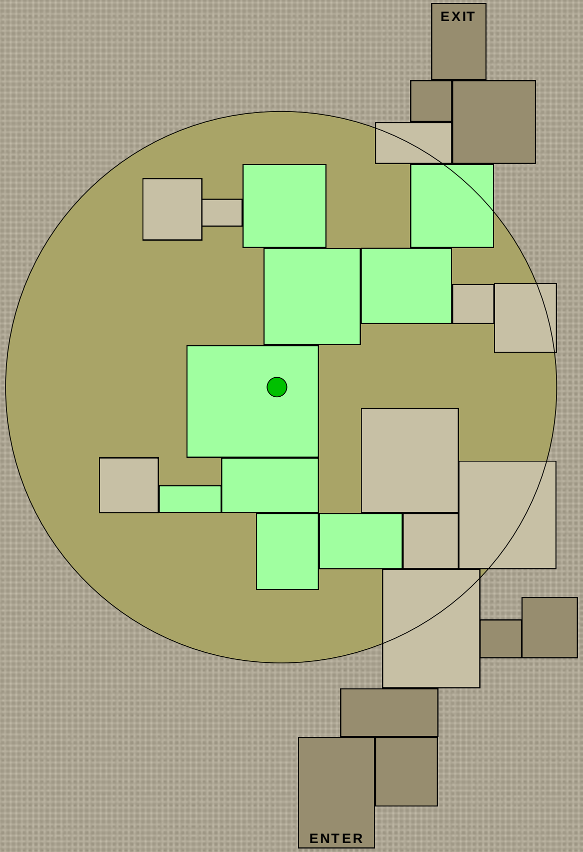
- The set of Navigation Areas surrounding the Survivor team
- The AI Director creates/destroys population as the AAS moves through the environment
- Allows for hundreds of enemies using a small set of reused entities



Goal: Promote Replayability
Procedurally Populated Environments

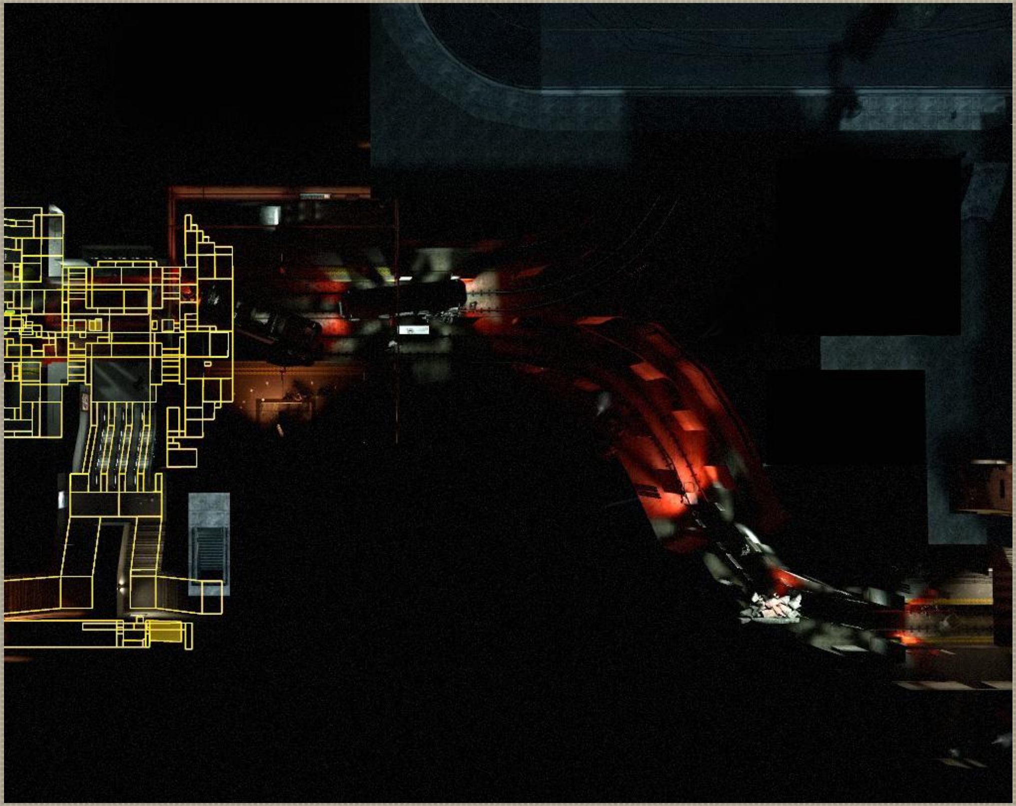
## Potentially Visible Areas

- The set of Navigation Areas potentially visible to any Survivor



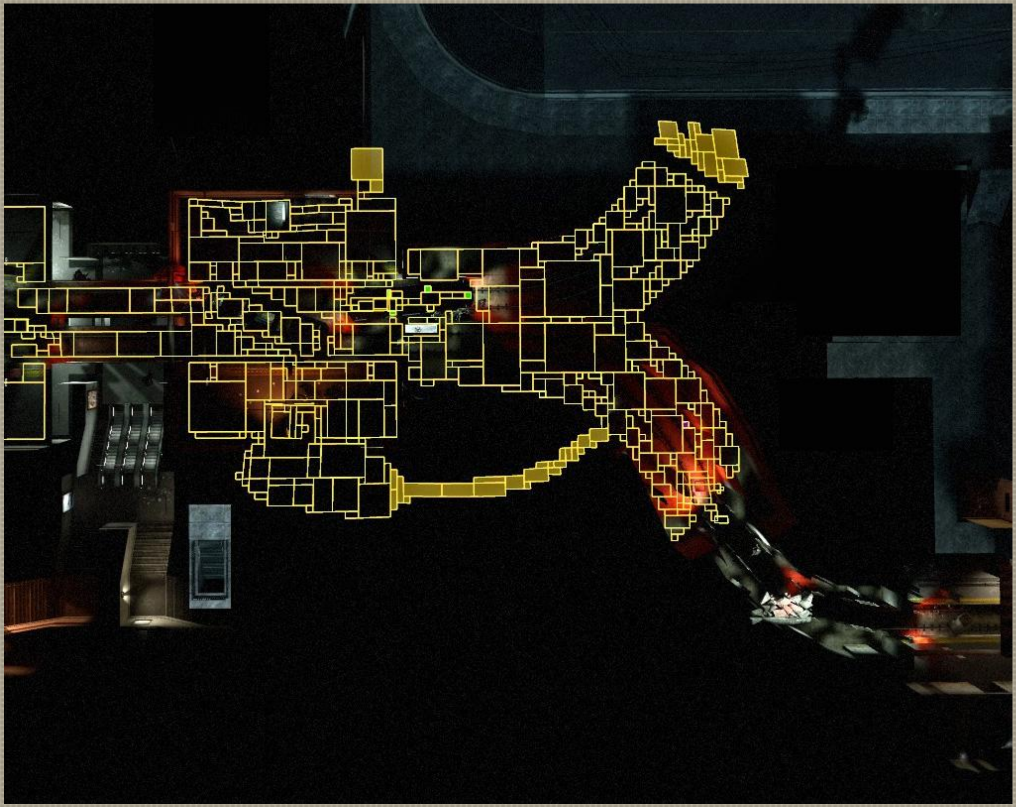
## Goal: Promote Replayability
## Procedurally Populated Environments

- The Active Area Set



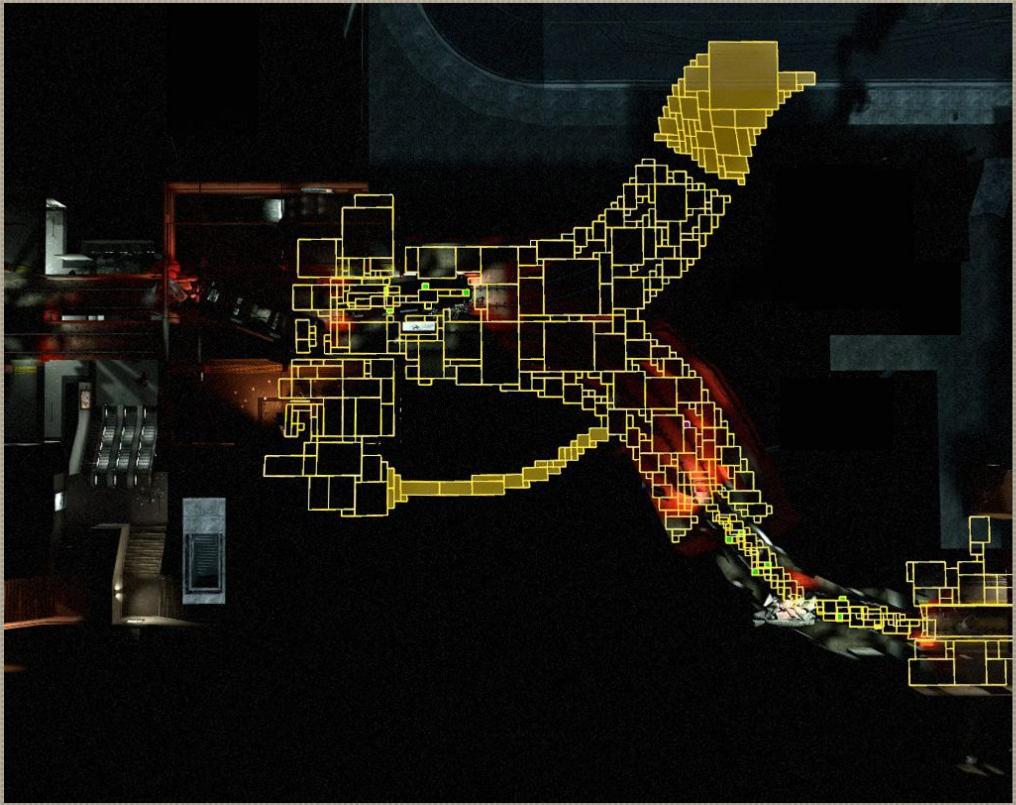
Goal: Promote Replayability
Procedurally Populated Environments

- The Active Area Set



## Goal: Promote Replayability
## Procedurally Populated Environments

- The Active Area Set



## Procedurally Populated Environments

- Populating via Structured Unpredictability
  - Not purely random, nor deterministically uniform
  - Population functions where space and/or time varies based on designer-defined amount of randomization
    - Example: Mob spawns
      - Occur at randomized interval between 90 and 180 seconds
      - Located at randomized spot “behind” the Survivor team
  - Structured Unpredictability = Superposition of several of these population functions

## Goal: Promote Replayability
## Procedurally Populated Environments

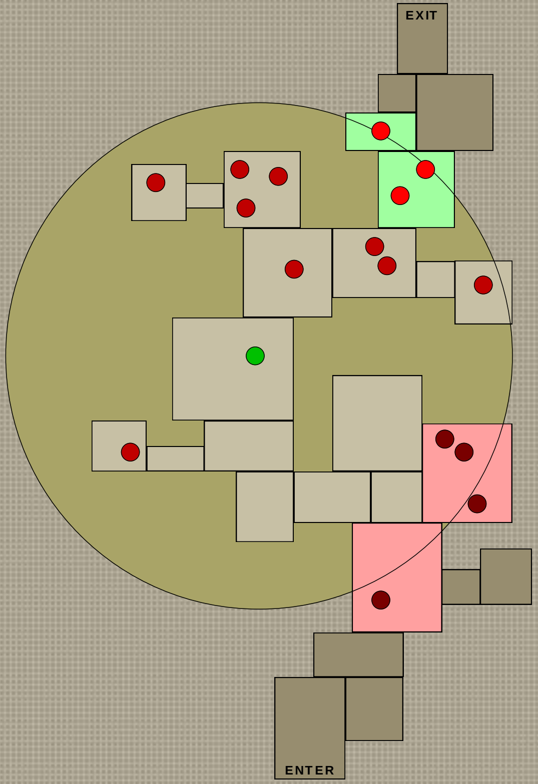
- Structured Unpredictability in Left 4 Dead
  - Wanderers (high frequency)
    - Common Infected that wander around in a daze until alerted by a Survivor
  - Mobs (medium frequency)
    - A large group (20-30) of enraged Common Infected that periodically rush the Survivors
  - Special Infected (medium frequency)
    - Individual Infected with special abilities that harass the Survivor team
  - Bosses (low frequency)
    - Powerful Infected that force the Survivors to change their strategy
  - Weapon Caches (low frequency)
    - Collections of more powerful weapons
  - Scavenge Items (medium frequency)
    - Pipe bombs, Molotovs, Pain Pills, Extra Pistols

## Goal: Promote Replayability
## Procedurally Populated Environments

- Populating Wandering Infected
  - Stored as a simple count (N) in each area
  - Counts are randomly determined at map (re)start based on Escape Route length and desired density
  - When an area **enters** the AAS
    - Create N Infected (if possible)
  - When an area **leaves** the AAS, or a **pending Mob needs more members**
    - Wanderers in the area are deleted and N is increased accordingly
  - Wanderer count (N) is zeroed:
    - When an area becomes visible to any Survivor
    - When the Director is in Relax mode

## Goal: Promote Replayability
## Procedurally Populated Environments

- Example
  - Green areas are entering the AAS and wanderers are being spawned
  - Red areas are leaving the AAS, and the Infected within will be de-spawned



## Goal: Promote Replayability
## Procedurally Populated Environments

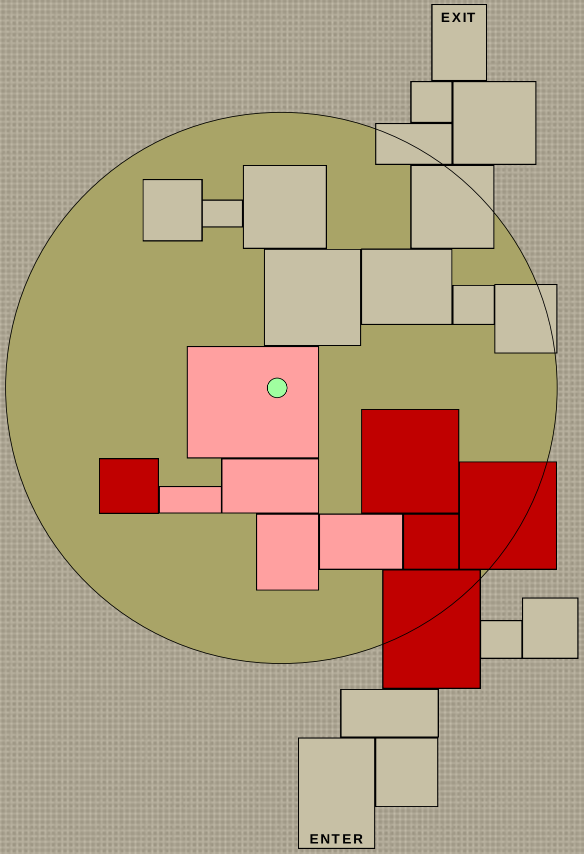
- Populating Mobs
  - Created at randomized intervals (90-180 seconds on Normal difficulty)
  - Boomer Vomit forces Mob spawn, resets random interval
  - Mob size grows from minimum just after spawn to maximum after a duration to balance difficulty of successive, frequent Mobs

## Goal: Promote Replayability
## Procedurally Populated Environments

- Populating the Special Infected
  - Created at individually randomized intervals
  - Use valid area in the AAS not visible by the Survivor team appropriate to Special's class
    - Boomers spawn ahead, since they are slow and can't chase well
    - Smokers attempt to select areas above the Survivor team

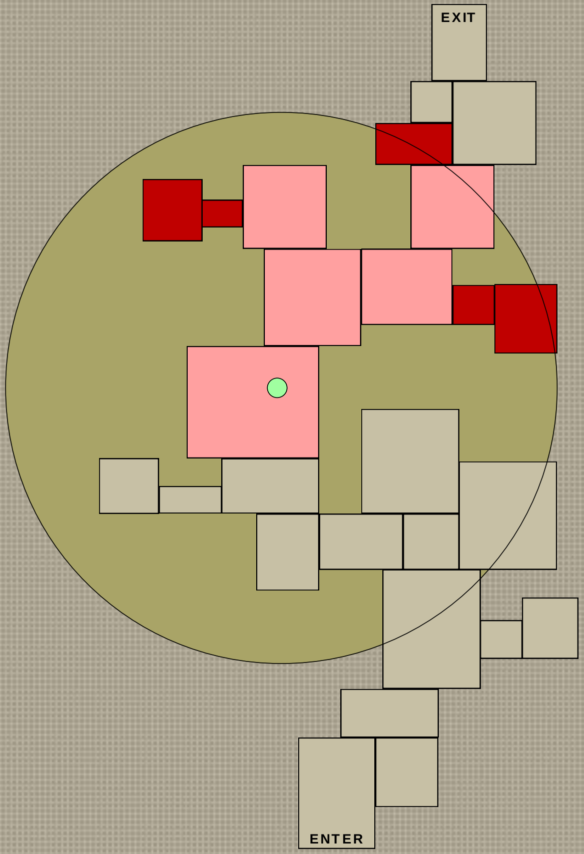
## Goal: Promote Replayability
## Procedurally Populated Environments

- **Where to spawn**
  - Behind Survivors
    - Only select valid areas in the AAS that are at or behind the Survivor team's "flow" distance
    - 75% of Mobs come from behind, since wanderers and Special/Boss Infected are usually engaged ahead of the team



## Goal: Promote Replayability
## Procedurally Populated Environments

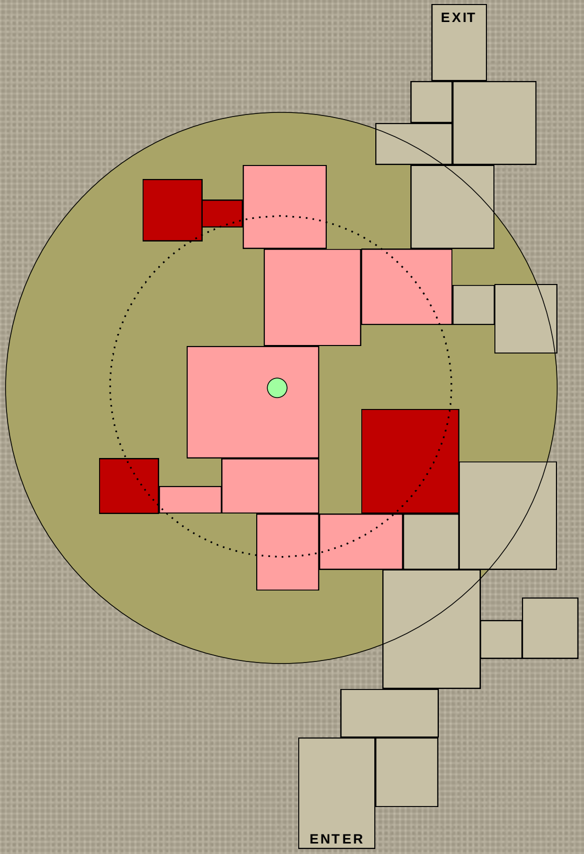
- Where to spawn
  - Ahead of Survivors
    - Only select valid areas in the AAS that are at or greater than the Survivor team's "flow" distance



Goal: Promote Replayability
Procedurally Populated Environments

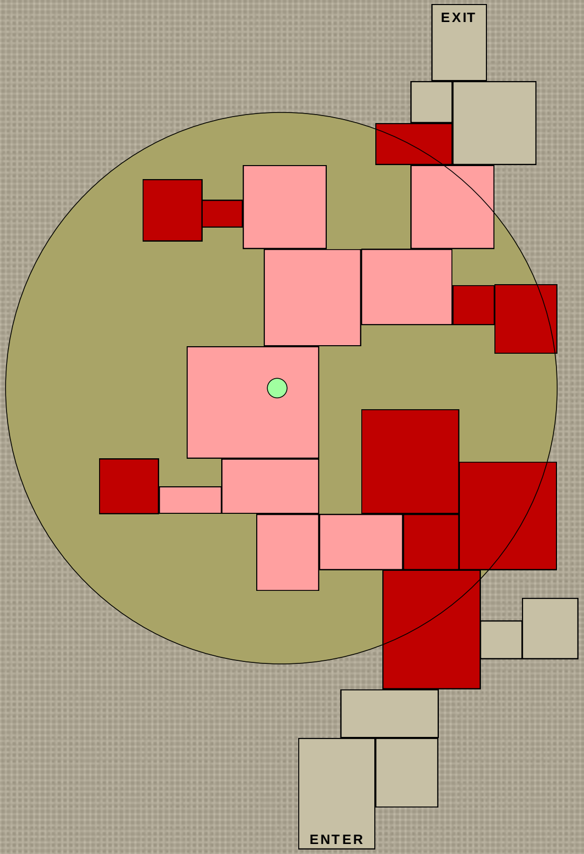
## Where to spawn

- Near Boomer Vomit Victim
  - Only select valid areas in the AAS that are near the Boomer Vomit Victim's "flow" distance



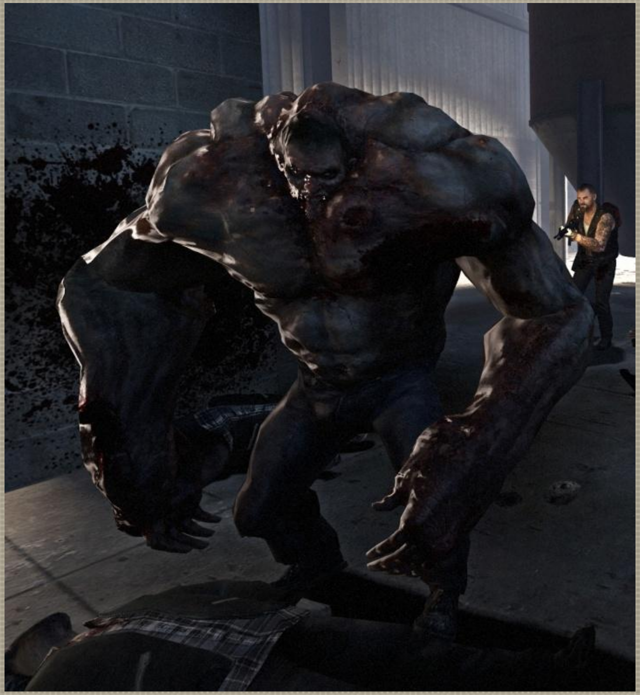
## Goal: Promote Replayability
## Procedurally Populated Environments

- Where to spawn
  - Anywhere
    - Any valid area in the AAS
    - Default if there are no valid areas in the more specific sets



## Goal: Promote Replayability
## Procedurally Populated Environments

- **Boss Population**
  - Positioned every N units along "escape route" +/- random amount at map (re)start.
  - Three Boss events are shuffled and dealt out: Tank, Witch, and Nothing.
  - Successive repeats are not allowed (ie: Tank, then Tank again)



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Tank | Witch | | Witch | | Tank | | Witch | Tank |

## Goal: Promote Replayability
## Procedurally Populated Environments

- **Weapon Caches**
  - Map designer creates several possible weapon caches in each map, the AI Director chooses which will actually exist
- **Scavenge Items**
  - Map designer creates many possible item groups throughout the map, the AI Direction chooses which groups actually exist
- **Why designer-placed?**
  - Prediction of possible locations beneficial in this case
  - Allows visual storytelling/intention
  - Solves item placement issues (leaning against wall, mounted in gun rack, etc)

## Goal: Promote Replayability

- Procedural Content
  - Promotes replayability
  - Solution for linear, yet replayable multiplayer experiences
  - Greatly multiplies output of development team
  - Improves community created content

## Goals of Left 4 Dead AI

- Deliver Robust Behavior Performances
- Provide Competent Human Player Proxies
- Promote Replayability
- **Generate Dramatic Game Pacing**

## Goal: Generate dramatic game pacing

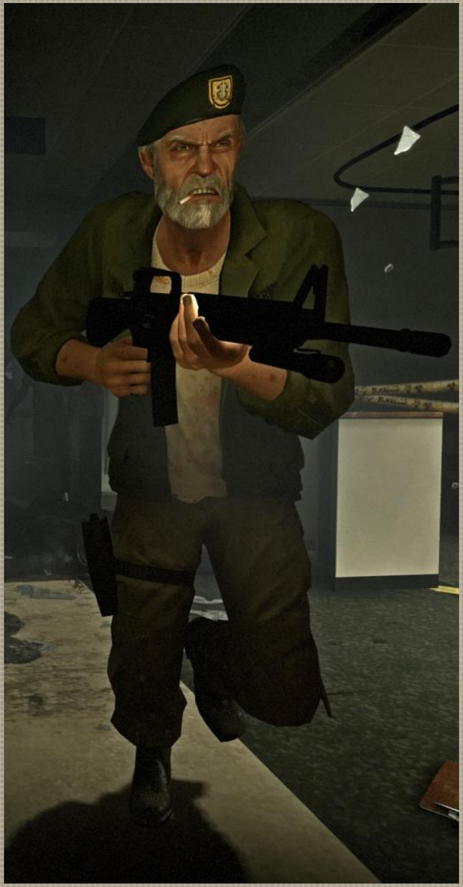
- What do we mean by "dramatic game pacing"?
  - Algorithmically adjusting game pacing on the fly to maximize "drama" (player excitement/intensity)
- Inspired by Observations from Counter-Strike
  - Natural pacing of CS is "spiky", with periods of quiet tension punctuated by unpredictable moments of intense combat
  - Constant, unchanging combat is fatiguing
  - Long periods of inactivity are boring
  - *Unpredictable* peaks and valleys of intensity create a powerfully compelling and replayable experience
  - Same scenario, often the same map, yet different and compelling experience each round

## Goal: Generate dramatic game pacing

- Adaptive Dramatic Pacing algorithm
  - Creates peaks and valleys of intensity similar to the proven pacing success of Counter-Strike
  - Algorithm:
    - Estimate the "emotional intensity" of each Survivor
    - Track the max intensity of all 4 Survivors
    - If intensity is too high, remove major threats for awhile
    - Otherwise, create an interesting population of threats
  - Another key system of the **"AI Director"**

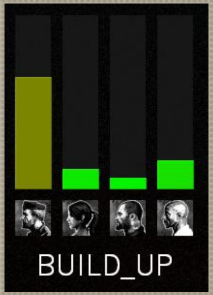
## Goal: Generate dramatic game pacing
## Adaptive Dramatic Pacing

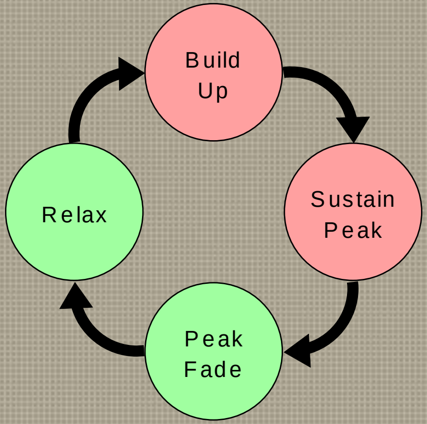
- Estimating the "emotional intensity" of each Survivor
  - Represent Survivor Intensity as a value
  - Increase Survivor Intensity
    - When injured by the Infected, proportional to damage taken
    - When the player becomes incapacitated
    - When player is pulled/pushed off of a ledge by the Infected
    - When nearby Infected dies, inversely proportional to distance
  - Decay Survivor Intensity towards zero over time
  - Do NOT decay Survivor Intensity if there are Infected actively engaging the Survivor



## Goal: Generate dramatic game pacing
## Adaptive Dramatic Pacing

- Use Survivor Intensity to modulate the Infected population
  - Build Up
    - Create full threat population until Survivor Intensity crosses peak threshold
  - Sustain Peak
    - Continue full threat population for 3-5 seconds after Survivor Intensity has peaked
    - Ensures minimum "build up" duration.
  - Peak Fade
    - Switch to minimal threat population ("Relax period") and monitor Survivor Intensity until it decays out of peak range
    - This state is needed so current combat engagement can play out without using up entire Relax period. Peak Fade won't allow the Relax period to start until a natural break in the action occurs.
  - Relax
    - Maintain minimal threat population for 30-45 seconds, or until Survivors have traveled far enough toward the next safe room, then resume Build Up.



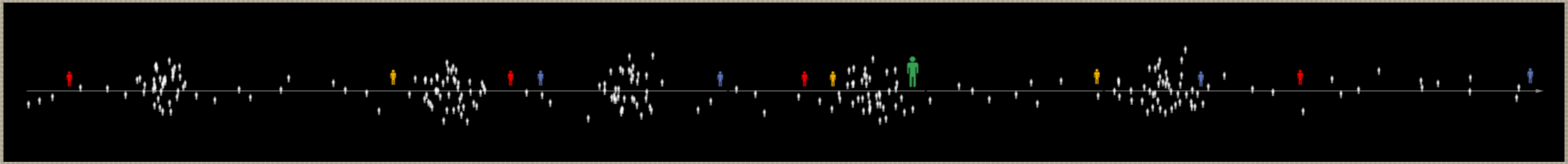


## Goal: Generate dramatic game pacing
## Adaptive Dramatic Pacing

- Full Threat Population (Build Up)
  - Wanderers
  - Mobs
  - Special Infected
- Minimal Threat Population (Relax)
  - No Wanderers until team is "calm"
  - No Mobs
  - No Special Infected (although existing Specials may attack)
- Boss Encounters NOT affected by adaptive pacing
  - Overall pacing affected too much if they are missing
  - Boss encounters are intended to change up the pacing anyhow

## Goal: Generate dramatic game pacing
## Adaptive Dramatic Pacing

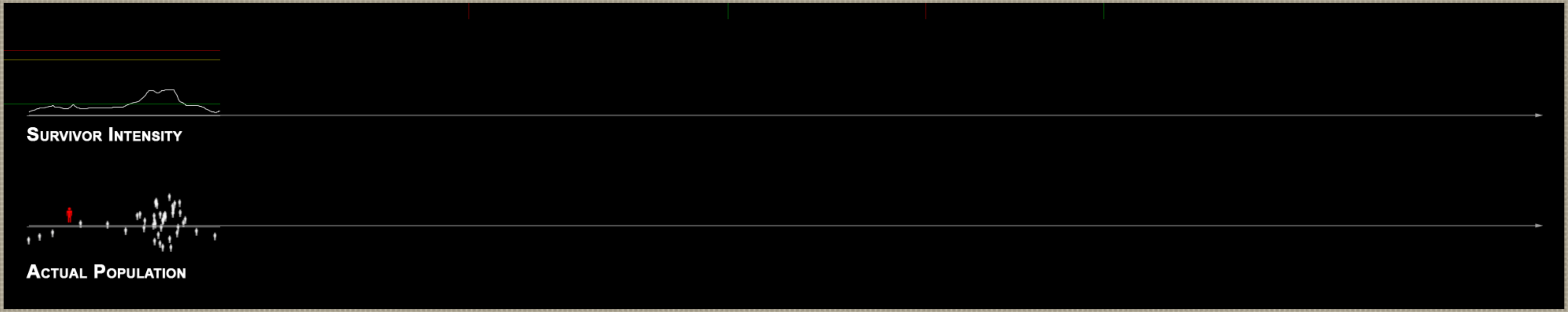
- An example procedurally generated population



VALVE

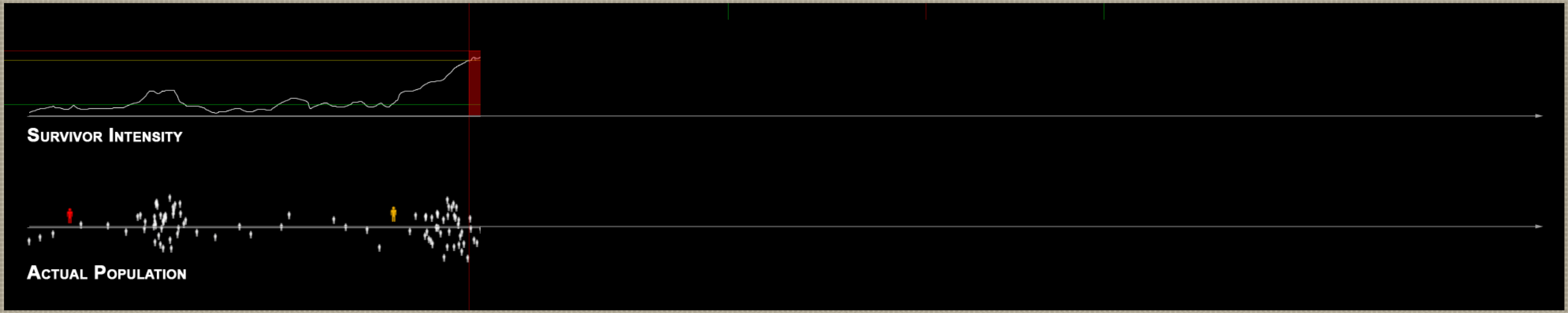
## Goal: Generate dramatic game pacing
## Adaptive Dramatic Pacing

- How the AI Director modulates the population based on the Survivor team's "emotional intensity"



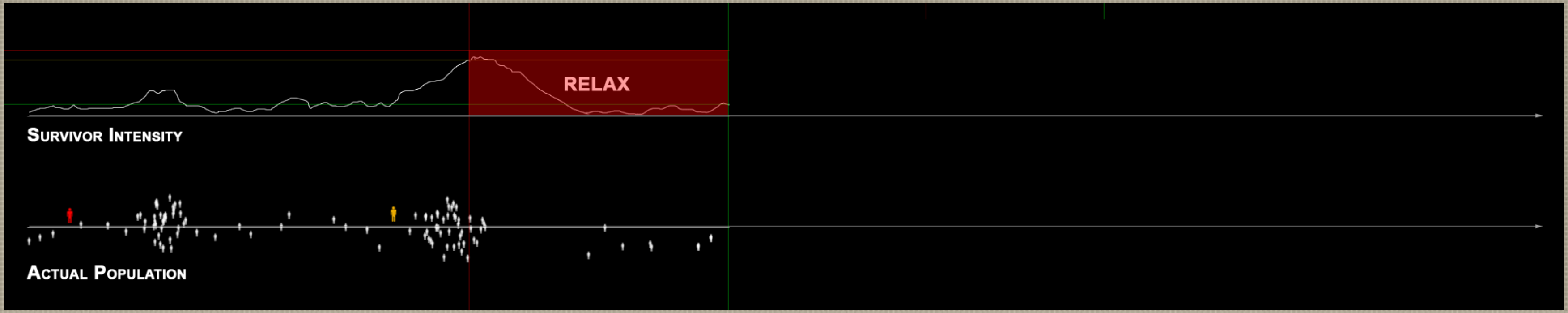
## Goal: Generate dramatic game pacing
## Adaptive Dramatic Pacing

- How the AI Director modulates the population based on the Survivor team’s “emotional intensity”



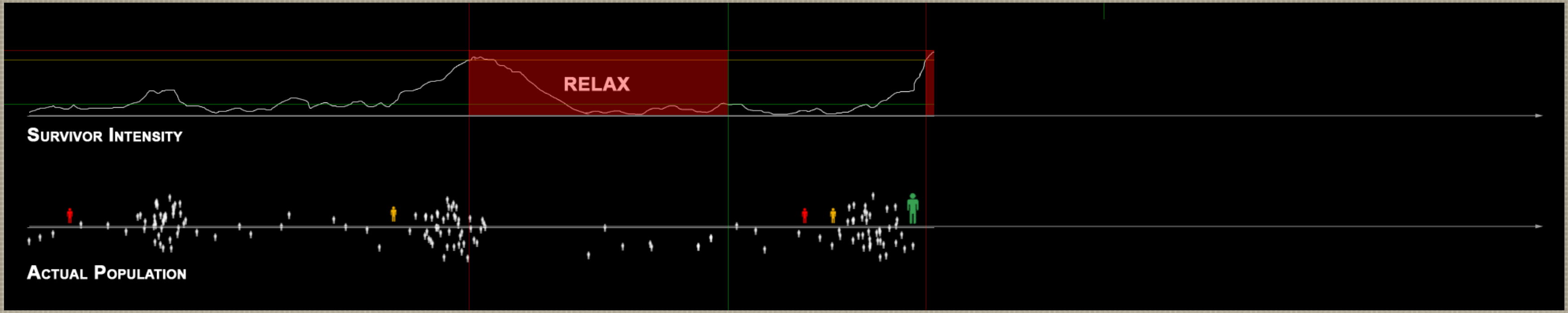
## Goal: Generate dramatic game pacing
## Adaptive Dramatic Pacing

- How the AI Director modulates the population based on the Survivor team's "emotional intensity"



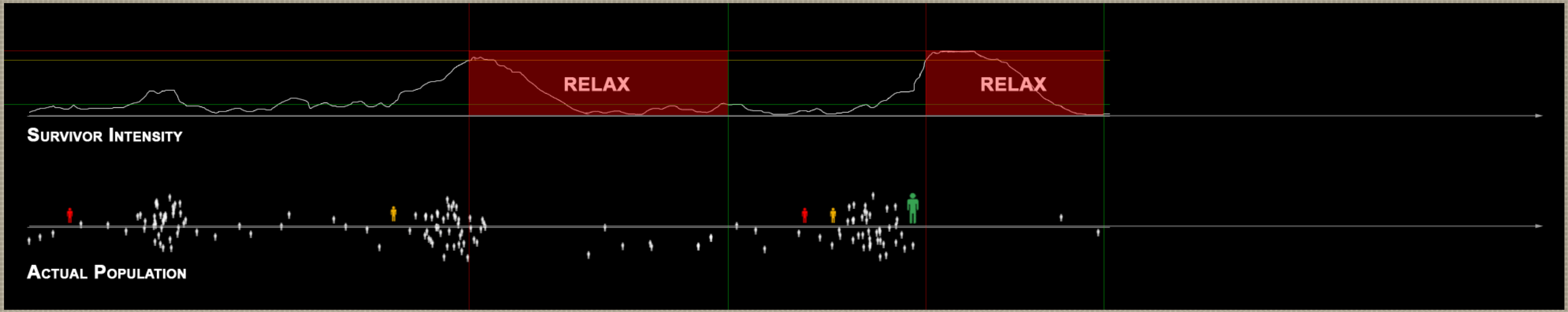
## Goal: Generate dramatic game pacing
## Adaptive Dramatic Pacing

- How the AI Director modulates the population based on the Survivor team’s “emotional intensity”



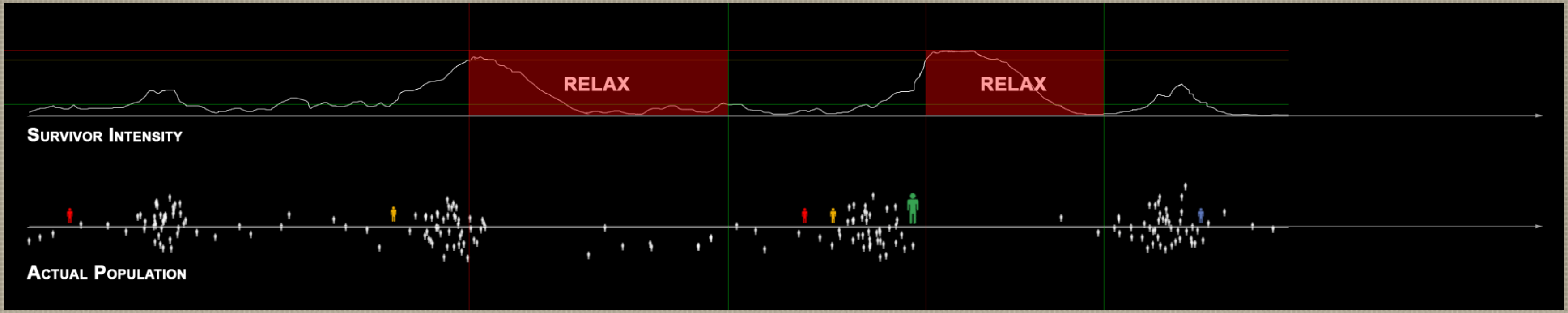
Goal: Generate dramatic game pacing
Adaptive Dramatic Pacing

- How the AI Director modulates the population based on the Survivor team’s “emotional intensity”



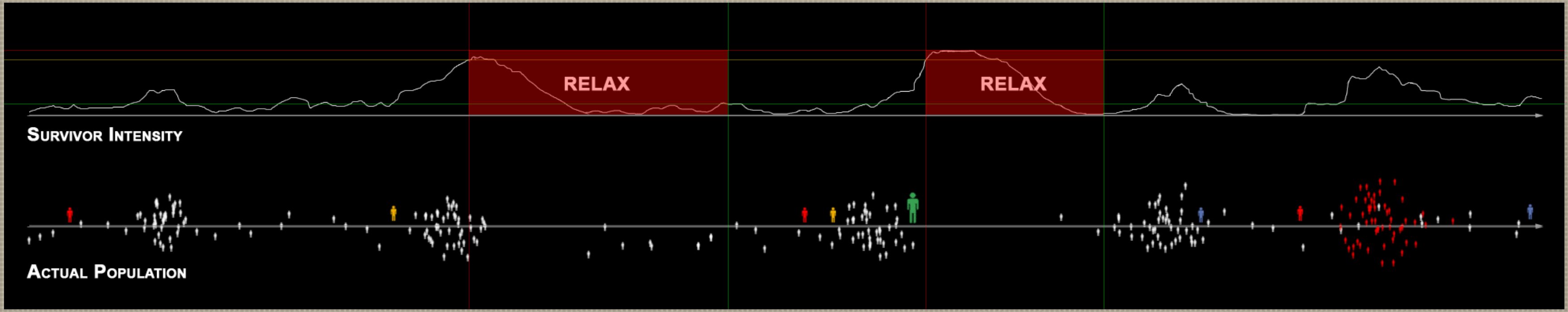
## Goal: Generate dramatic game pacing
## Adaptive Dramatic Pacing

- How the AI Director modulates the population based on the Survivor team's "emotional intensity"



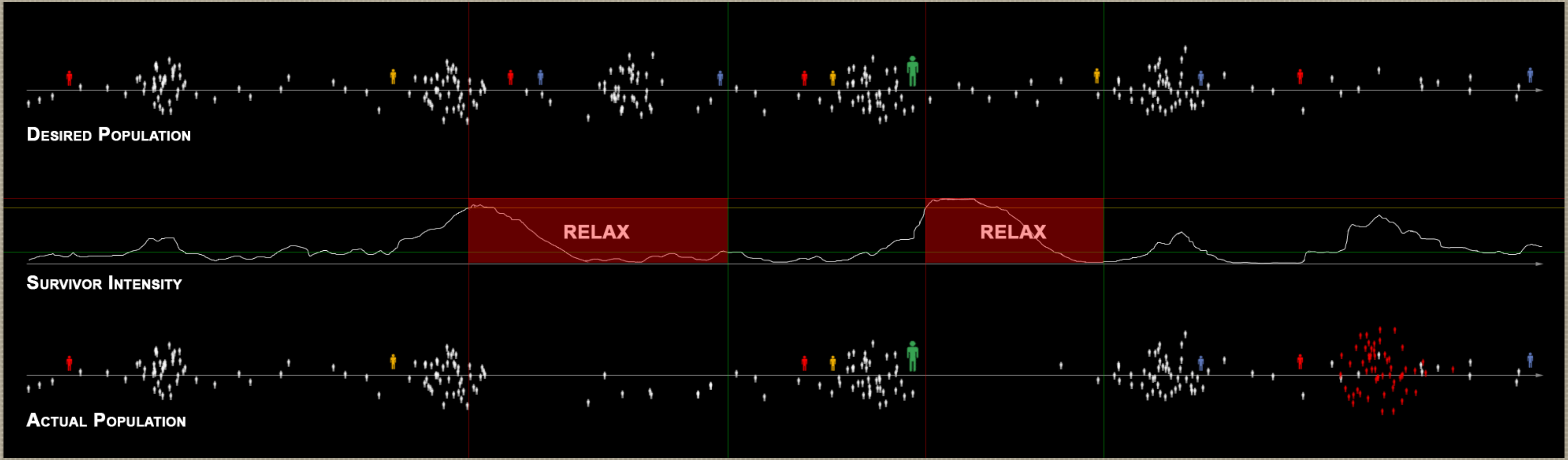
## Goal: Generate dramatic game pacing
## Adaptive Dramatic Pacing

- How the AI Director modulates the population based on the Survivor team's "emotional intensity"



Goal: Generate dramatic game pacing
Adaptive Dramatic Pacing

- Comparing population after modulation by the AI Director

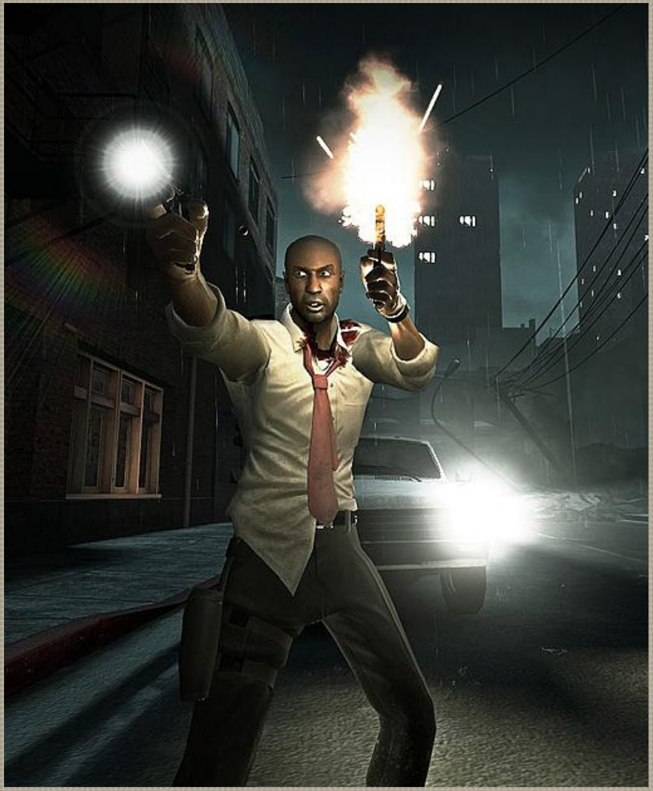


## Goal: Generate dramatic game pacing

- Adaptive Dramatic Pacing reacts to Survivor team
  - Generates reliable peaks of intensity without completely overwhelming the team
  - Because of player variation and procedural threat population, timing and location of peaks will differ
    each time the game is played
- Algorithm adjusts pacing, not difficulty
  - Amplitude (difficulty) is not changed, frequency (pacing) is
- Simple algorithms can generate compelling pacing schedules
  - Survivor Intensity estimation is crude, yet the resulting pacing works

## Future Work

- Just the beginning…
  - Expand repertoire of verbs available to the AI Director
  - Explore further procedural content generation opportunities
  - How to utilize these systems in other game projects?



## For more information…

- www.L4D.com
- www.valvesoftware.com
- mike.booth at valvesoftware dot com

## *THANK YOU!*