


Zombies

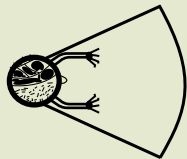
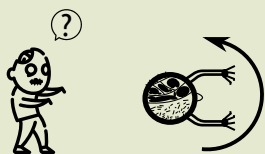
a.k.a the "Zs" representation in the GUI : 

Zs are **walking** around randomly at **walkSpeed**
They can rotate up to **zombieMaxRotation** degrees

Zs perceive the world within a **zombiePerception** radius

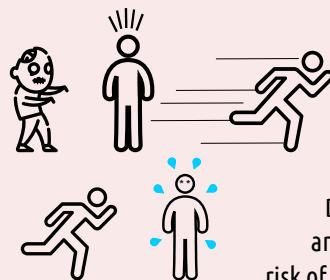
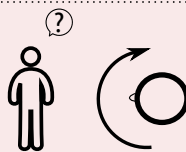
When a H is within **zombiePerception** radius, a Z will engage **pursuit**
For the duration of **pursuit**, Zs run at **zombieRunspeed**

Each **pursuing** Z releases pheromones on ground tiles
Zs are attracted by pheromones inside their **zombiePerception** radius




SOLDIERS

are Hs with their own parameter values, usually better than civilians.



Humans

a.k.a. the "Hs" representation in the GUI : 

Hs are **walking** around randomly at **walkSpeed**
They can rotate up to **humanMaxRotation** degrees

If Hs are **informed** about rescue zone locations, they move towards the nearest one
informed Hs can share the rescue zone locations with others with a **humanInformProbability** chance

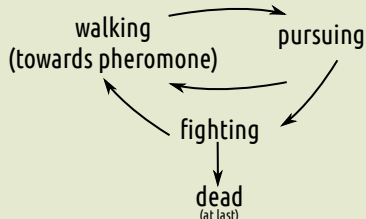
Hs perceive the world within a **humanPerception** radius when a Z is within **humanPerception** radius, Hs try to **flee**

During **flee**, Hs run at **humanRunspeed** and have a **humanExhaustionProbability** risk of becoming **exhausted** and stop running



Upon seeing another H running, Hs will start running with them with a **humanFollowProbability** chance

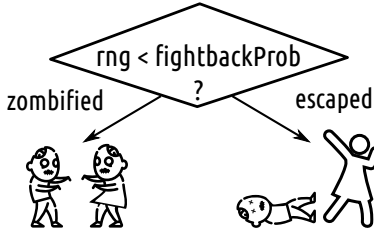
Zs behaviour State machine



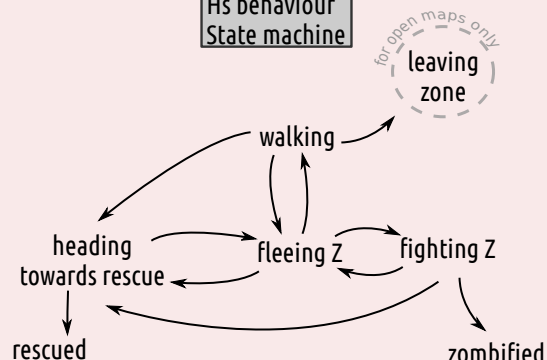
Fight



if H is in **infectionRange** of Z,
H has **humanFightBackProbability** chance to fight and kill Z

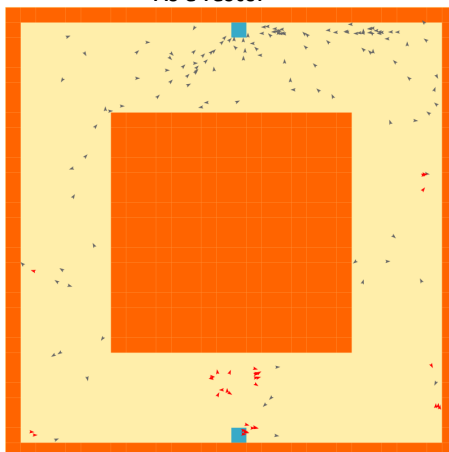


Hs behaviour State machine



The world

As a raster





Stadium: 4 rescue zones of width 1

 Ground  Wall  Rescue

 Zombie  Soldier
 Human  Red cross


Environment and global parameters

 **zombiePheromoneEvaporation** set the rate of evaporation.

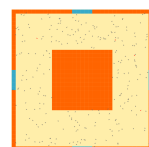
 There is **numberZombies** of Zs at the beginning,

 and **numberHumans** of Hs,

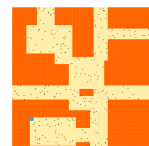
among which a **humanInformedRatio** is aware of the rescue point location

 and **armySize** soldiers, if **ArmyOption** is set to Army

Other built-in worlds



stadium:
2 rescue zones of width 1



jaude:
one rescue zone of width 1



1 simulation step: 1 s
1 simulation: 500 steps

Metrics

The `SimulationResult` object provides the following metrics (usually defined as `val result = zombieInvasion(...)` in a `ScalaTask`). Some of the metrics are time series, and can thus be aggregated via a temporal step parameter named `by`. The default value for `by` is 20. It means that the 500 steps of these time series are sampled by the fixed step width defined through `by` (for convenience and size of output data).

Agent-related indicators

- `humansDynamic`(by: Int = 20): Array[Int] sampled time series (each by time steps) of humans
- `walkingHumansDynamic`(by: Int = 20): Array[Int] sampled time series of walking humans
- `runningHumansDynamic`(by: Int = 20): Array[Int] sampled time series of running humans
- `zombiesDynamic`(by: Int = 20): Array[Int] sampled time series of zombies
- `walkingZombiesDynamic`(by: Int = 20): Array[Int] sampled time series of walking zombies
- `runningZombiesDynamic`(by: Int = 20): Array[Int] sampled time series of running zombies

Event-related indicators

- `rescuedDynamic`(by: Int = 20): Array[Int] sampled time series of rescued humans
- `killedDynamic`(by: Int = 20): Array[Int] sampled time series of killed zombies
- `zombifiedDynamic`(by: Int = 20): Array[Int] sampled time series of zombified humans
- `fleeDynamic`(by: Int = 20): Array[Int] sampled time series of humans fleeing from zombies
- `pursueDynamic`(by: Int = 20): Array[Int] sampled time series of zombies pursuing humans
- `humansGoneDynamic`(by: Int = 20): Array[Int] sampled time series of humans who left the world
- `zombiesGoneDynamic`(by: Int = 20): Array[Int] sampled time series of zombies who left the world

Global indicators

- `totalZombified`: Int total number of zombified humans over the course of the simulation
- `halfZombified`: Int time at which half of humans are zombified
- `peakTimeZombified`(window: Int = 20): Int time at which the zombification is the most intense (smoothed over a window size window)
- `peakSizeZombified`(window: Int = 20): Int number of zombification when zombification is the most intense (smoothed over a window size window)
- `totalRescued`: Int total number of humans rescued
- `halfTimeRescued`: Int time at which half of the humans have been rescued
- `peakTimeRescued`(window: Int = 20): Int time at which rescue is the most intense (smoothed over a window size window)
- `peakSizeRescued`(window: Int = 20): Int number of rescue at the time of `peakTimeRescued`

Spatial indicators

- `spatialMoranZombified`: Double spatial autocorrelation of the location of zombification events cumulated over time. Takes values between -1 (strongest negative autocorrelation) 0 (no spatial autocorrelation) and 1 (strongest autocorrelation)
- `spatialDistanceMeanZombified`: Double average distance between zombification events
- `spatialEntropyZombified`: Double entropy of zombification events, or how zombification is uniformly distributed across cells ($\in [0;1]$)
- `spatialSlopeZombified`: Double level of aggregation of zombification events, can be interpreted as "clustering" intensity