# Turtles

Entidades

Nichos (é possível criar vários nichos, no entanto, como implementar a competição em vários nichos e principalmente como representar a aptidão nos diferentes nichos – apt máxima? Apt média? Apt ponderada pelo tamanho dos recursos do nicho?)

# Interface

Turtles de diferentes formatos para os diferentes papéis - ok

Papéis híbridos (formas?)

Tamanho variável de acordo com o nível de recursos - ok

Cor de acordo com a aptidão (relativa ou absoluta?). A experiência com a relativa é que todos ficam vermelhos, no máximo amarelos. Com a absoluta se tem todas as cores. Outra possibilidade é de acordo com a capacidade de sobrevivência. Se o agente tem marketshare para receber a quantidade de recursos que deve pagar por período e mais um pouco ele fica verde. Se receber muito próximo do que precisa pagar, amarelo. Se receber menos, vermelho (porque passará a diminuir até morrer). ok

Entidades fixas ao mapa. Não se movem ok (talvez outros formatos? Intervalos constantes para ficar fácil de ver...)

Gráfico de fitness médio (e desvio padrão) ok

Gráfico de recursos médio (e desvio padrão) ok

Gráfico de vontade de compartilhar média (e desvio padrão)

Gráfico de motivação para aprender média (e desvio padrão)

Gráfico de interações bem sucedidas por iteração

Gráfico de número de bits trocados por iteração

Indicador do maior fitness ok

Indicador do maior número de recursos ok

Histograma do número de recursos (ver modelo distribuição de riqueza) ok

# Mecanismos

## Algoritmos genéticos

Roleta (ok, modelo lottery) - ok

Crossover

Mutação

## Morte e nascimento

Morte dos sem recursos (ok)

Nascimento de startups (ainda a fazer)

## Recursos

Cobrança de recursos com mínimo e adicional proporcional ao tamanho – ok

Market share de múltiplos nichos – ainda trabalhando com apenas um nicho

Recursos recebidos de acordo com atividade de investigação (mutação)

Recursos recebidos de acordo com orçamento fixo

Recursos recebidos de acordo com atividade de difusão (ensino, consultoria, outsourcing de projetos)

Recursos recebidos por crossover (licenciamento)

## Avaliação de ações

Avaliação das ações executadas (melhora na aptidão absoluta? Melhora na aptidão relativa? Quantidade de recursos empregados x recursos recebidos – pode até ter um horizonte de payback como em quantas iterações haverá o payback)

Adaptação dos parâmetros individuais

Lista de reputação (é mesmo necessário? O método da roleta dispensa a lista geral)

Histórico de interações (esta sim pode ser necessária – cada entidade manteria uma lista das entidades com quem já lidou e uma classificação da experiência (positiva ou negativa) que será adicionada à reputação da outra entidade.

Mecanismo de escolha de procurar parceiros ou não (motivação em aprender).

Mecanismo de escolha em aceitar parceria ou não (disposição em compartilhar).

# Variáveis no modelo

## entities-own

science-knowledge

;; lets the model know which entities have active scientific knowledge

science?

;; stores the technological knowledge of the entity. It is a characteristic of Consumers and Diffusers

tech-knowledge

;; lets the model know which entities have active technological knowledge

technology?

;; stores the Hamming distance of the entity (currently just from one niche)

fitness

;; stores the amount of resources kept by the entity

resources

;; Stores the entity's reputation, given its resources and fitness in its niche

reputation

;; does the entity assume a generator role in the ecosystem?

generator?

;; does the entity assume a generator role in the ecosystem?

consumer?

;; does the entity assume a generator role in the ecosystem?

diffuser?

;; does the entity assume a generator role in the ecosystem?

integrator?

;; entity's willingness to share knowledge with others

willingness-to-share

;; entity's motivation to learn from others

motivation-to-learn

;; entity's creation performance

creation-performance

]

niches-own [

;; total-resources of a niche (put this on a slider in the future)

niche-resources

;; stores the demand DNA of the niche

niche-demand

;; it is the average fitness of the entities competing on the niche

average-fitness

]

;;;;;;;;;;;;;;;;;;;;;;; globals ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

globals [

;; holds counter value for which instruction is being displayed

current-instruction

;; stores the niche-demand DNA for comparison

niche-demand-now

;; it is the sum of the fitness of every entity competing on the niche

total-fitness

;; agentset of possible partners for crossover

possible-partners

]