

# VictimSim2: Manual de Uso

## Sumário

1	Simulador .....	2
1.1	Funcionamento .....	2
2	Ambiente .....	2
2.1	Arquivos de configuração do ambiente .....	3
3	Vítima .....	3
3.1	Arquivo de dados dos sinais vitais .....	4
4	Agentes .....	5
4.1	Arquivo de configuração de um agente .....	5
4.2	Criação de um agente .....	5
5	Métricas de Desempenho .....	5
5.1	Definições básicas .....	5
5.2	Métricas de Busca .....	6
5.3	Métricas de Resgate .....	6

## 1 Simulador

O simulador foi criado para testar algoritmos de IA em um cenário de catástrofe natural, atentado ou acidente. O ambiente é representado por um grid 2D no qual há vítimas, obstáculos e agentes.

Os cenários que podem ser simulados são compostos por agentes que tentam localizar as vítimas e por agentes (os mesmos ou outros) que levam um kit de socorros as vítimas localizadas.

O simulador contabiliza o número de vítimas localizadas pelas leituras de sinais vitais realizadas pelos agentes. A contagem do número de vítimas resgatadas (ou salvas) é feita pelo número de kits de socorro deixados, sendo um por vítima.

### 1.1 Funcionamento

O motor do simulador está na classe `vs.Environment`<sup>1</sup>, mais precisamente implementada no método `run`. Enquanto houver um agente no estado `ACTIVE` ou `IDLE` o simulador permanece no ciclo de invocação dos agentes. Abaixo, um pseudo-código do motor da simulação:

```
Enquanto há ao menos um agente IDLE ou ACTIVE:
  Para cada agente ACTIVE:
    Invoque o método agente.Deliberate()
    Se o agente estourou o tempo limite então mudar estado para DEAD
    Se o agente não tem mais ações
      Se o agente está na posição base então mudar estado para ENDED
      Se o agente não está na posição base então mudar estado para DEAD
```

A interação entre os agentes e o simulador acontece pelo método `deliberate` presente na classe `vs.AbstAgent` e que deve ser implementado por toda classe que a especializa. A ideia é que os agentes executem o ciclo de *percepção – deliberação – atuação*.

## 2 Ambiente

A figura 1 apresenta um exemplo ilustrativo de um ambiente na forma de um grid de 4 colunas (`GRID_WIDTH`) por 5 linhas (`GRID_HEIGHT`). A célula base está nas coordenadas (2, 3).

Os agentes exploradores são definidos pelo conjunto  $A_e = \{E_1, E_2\}$ , os socorristas pelo conjunto  $A_s = \{S_1, S_2\}$ , o conjunto das vítimas dispersas no ambiente,  $V = \{v_1, v_2, v_3\}$  e os obstáculos. Neste caso, só há obstáculos intransponíveis representados pelos quadrados pretos.

---

<sup>1</sup> vs é o diretório onde se encontram os códigos principais do simulador (vs = victim simulator)

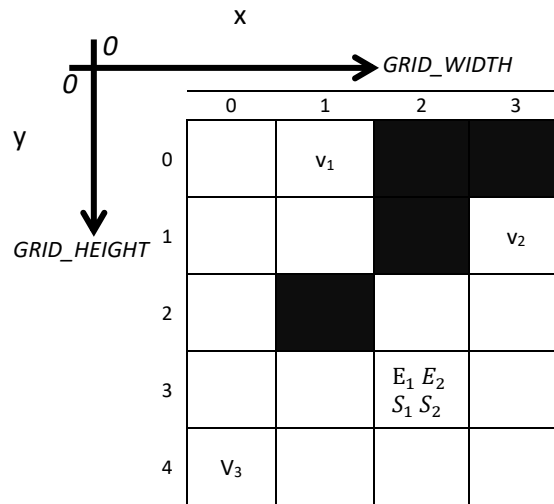


Figura 1: Ambiente com os agentes na posição base, vítimas ( $V=\{v_1, v_2, v_3\}$ ) e obstáculos na cor preta.

Cada célula tem um grau de facilidade e/ou dificuldade de acesso que vai de  $[0, 100]$  que chamaremos simplesmente de *dificuldade de acesso*. Ela afeta somente a ação *walk* dos agentes conforme especificado abaixo:

- **[0, 1[**: a célula oferece facilidade de movimentação;
- **1**: a célula não oferece facilidade nem dificuldade (é o padrão);
- **[1, 100[**: a célula oferece dificuldade, mas ainda é transponível;
- **100**: a célula é intransponível (e.g. uma parede, rocha).

Por exemplo, suponha que um agente tenha um tempo padrão de 1.5 para andar na diagonal e entra em uma célula de dificuldade 2. O simulador calcula que o tempo de *walk* é igual a  $1.5 * 2$ .

A célula base é um dos parâmetros de entrada para o simulador do ambiente. As vítimas estão distribuídas no grid e há somente uma vítima por célula. Nenhuma célula pode conter vítima e obstáculo intransponível.

## 2.1 Arquivos de configuração do ambiente

O ambiente muda de acordo com as configurações contidas nos arquivos abaixo. Os agentes que você criar **não devem ler estes arquivos**.

**env\_size.txt**: tamanho do ambiente que é um grid de altura x comprimento

```
BASE 0,0          ## posição inicial dos agentes
GRID_WIDTH 30     ## largura do grid em células
GRID_HEIGHT 30    ## altura do grid em células
WINDOW_WIDTH 400  ## tamanho da janela em pixels
WINDOW_HEIGHT 400 ## altura da janela em pixels
DELAY 1.0         ## delay para atrasar a atualização da GUI em segundos
```

**env\_obst**: contém o posicionamento das paredes no grid em coordenadas (lin, col)

```
x1,y1,d1      ## coordenada (x, y) do obstáculo 1 e d1 dificuldade de acesso
...
xm,ym,dm      ## coordenada (x, y) do obstáculo m e dm dificuldade de acesso
```

## 3 Vítima

Cada vítima ocupa uma célula do ambiente, não pode haver mais de uma vítima por célula nem vítimas em células com obstáculos intransponíveis. As vítimas não se movem. Suas posições são determinadas na configuração do ambiente (ver env\_victims.txt).

Uma vítima possui um conjunto de sinais vitais conforme a Tabela 1. Observar que os agentes não possuem acesso ao valor da gravidade nem ao seu label.

Sigla	Extenso	Faixa de valores
<b>pSist</b>	Pressão sistólica	[5, 22]
<b>pDiast</b>	Pressão diastólica	[0, 15]
<b>qPA</b>	Qualidade de pressão	[-10, 10] sendo 0: o melhor equilíbrio entre pSist e pDiast -10: a pior qualidade quando pSist e/ou pDiast estão baixas +10: é a pior qualidade quando pSist e/ou pDiast estão altas
<b>pulso</b>	Batimentos de pulso/minuto	[0, 200] bpm
<b>resp</b>	Frequência respiratória/minuto	[0,22] fpm
<b>grav</b>	Valor de gravidade de ferimentos (float)	[0, 100]
<b>label</b>	Classe de risco de morte	1=CRÍTICO 2=INSTÁVEL 3=POTENCIALMENTE ESTÁVEL 4=ESTÁVEL

Tabela 1: Sinais vitais das vítimas. Agentes não leem diretamente os valores de gravidade ou o label.

### 3.1 Arquivo de dados dos sinais vitais

Este arquivo não deve ser acessado diretamente pelos agentes, somente em problemas que envolvam o uso de técnicas supervisionadas de aprendizado de máquina.

**env\_victims.txt:** contém o posicionamento das vítimas no grid em coordenadas (lin, col). Este arquivo está relacionado ao env\_vital\_signals.txt (deve haver uma correspondência 1:1)

```

x1,y1          ## coordenada (x, y) da vítima 1
...
xn,yn          ## coordenada (x, y) da vítima n

```

**env\_vital\_signals.txt:**

Para uma vítima  $i$  temos 5 sinais vitais que resultam a gravidade  $g_i$  da vítima com a respectiva classe. Todos os valores são números reais criados de modo randômico dentro dos intervalos apresentados.

*$i$  pSist, pDiast, qPA, pulso, resp, grav, label*

#### Exemplo

i	si1	si2	si3	si4	si5	g1	y1
	pSist	pDiast	qPA	pulso	resp	grav	label
0,	8.5806,	2.2791,	-8.4577,	56.8384,	9.2229,	33.5156,	2

Observar que cada linha de env\_victims.txt se relaciona posicionalmente com a respectiva linha de env\_vital\_signals.txt.

env_victims.txt	env_vital_signals.txt
3, 1	0, 8.5806, 2.2791, -8.4577, 56.8384, 9.2229, 33.5156, 2
1, 0	1, ...
0, 4	2, ...

Tabela 2: relação entre os arquivos env\_victims.txt e env\_vital\_signals.txt

## 4 Agentes

Um agente pode andar nas verticais, horizontais e diagonais sempre uma célula por vez. Dois ou mais agentes podem ocupar uma mesma célula. Um ou mais agentes podem ocupar a mesma célula que uma vítima.

De forma geral, um agente é capaz de detectar colisão com obstáculos intransponíveis, final de grid e vítimas. Também, é capaz de ler os sinais vitais de uma vítima.

Os agentes não podem acessar o sistema de coordenadas do ambiente. Portanto, devem criar um sistema de coordenadas.

### 4.1 Arquivo de configuração de um agente

Cada agente deve ter um arquivo de configuração. **Todos os parâmetros devem estar com valores mesmo que não sejam utilizados para evitar erros.** Os custos das ações são os padrões afetados pela dificuldade de acesso às células.

```
NAME EXPLORER1      ## nome do agente para print de mensagens
COLOR (255, 0, 127) ## cor para desenho do agente
TRACE_COLOR (255,153,204) ## cor para deixar pegadas nas células visitadas
TLIM 40.0           ## tempo limite para o agente executar sua tarefa
COST_LINE 1.0        ## tempo padrão para andar uma célula na hor. ou vertical
COST_DIAG 1.5        ## tempo padrão para andar uma célula na diagonal
COST_READ 2.0        ## tempo padrão para ler os sinais vitais de uma vítima
COST_FIRST_AID 1.0   ## tempo padrão para deixar o kit de primeiro-socorros
```

### 4.2 Criação de um agente

Criar uma classe que implementa a classe abstrata vs.**AbstAgent**.

**Consulte os métodos públicos de vs.AbstAgent e os exemplos disponíveis em**

<https://github.com/tacla/VictimSim2>

## 5 Métricas de Desempenho

### 5.1 Definições básicas

Considerar as definições dos conjuntos abaixo nas fórmulas de cálculo de desempenho que são calculadas pelo simulador.

- $V$ : conjunto de vítimas dispersas no ambiente, sendo  $|V|$  o número de vítimas (cardinalidade do conjunto). Este parâmetro é calculado a partir da leitura do arquivo de entrada. As vítimas estão classificadas de acordo com o risco de morte. Portanto, o conjunto  $V$  é composto pela união dos seguintes conjuntos disjuntos ( $V = \bigcup_{i=1}^4 V_i$ ):
  - $V_1$ : estado **CRÍTICO** (CLASSE=1)
  - $V_2$ : estado **INSTÁVEL** (CLASSE=2)
  - $V_3$ : estado **POTENCIALMENTE INSTÁVEL** (CLASSE=3)
  - $V_4$ : estado **ESTÁVEL** (CLASSE=4)

- $V_e$ : conjunto de vítimas localizadas pelo explorador tal que  $|V_e| \leq |V|$ . O conjunto das vítimas encontradas também é composto pela união dos conjuntos disjuntos ( $V_e = \bigcup_{i=1}^4 V_{e_i}$ ):
  - $V_{e_1}$ : estado **CRÍTICO** (CLASSE=1)
  - $V_{e_2}$ : estado **INSTÁVEL** (CLASSE=2)
  - $V_{e_3}$ : estado **POTENCIALMENTE INSTÁVEL** (CLASSE=3)
  - $V_{e_4}$ : estado **ESTÁVEL** (CLASSE=4)
- $t_e$ : tempo efetivamente gasto pelo explorador tal que  $t_e \leq T_e$

As definições para o conjunto das vítimas salvas são idênticas às das vítimas encontradas. Portanto, basta substituir o subscrito  $e$  por  $s$ .

## 5.2 Métricas de Busca

As métricas de desempenho de busca de vítimas são as seguintes (calculadas pelo simulador).

Sigla	Descrição	Fórmula
<b>pve</b>	porcentual de vítimas encontradas	$pve =  V_e / V $
<b>pte</b>	porcentual do tempo de exploração utilizado	$pte = t_e/T_e$
<b>veg</b>	vítimas encontradas ponderada por classe de gravidade. Retrata a capacidade do agente em localizar vítimas em estado mais grave, daí o uso da ponderação mais alta para as vítimas mais graves.	$veg = \frac{6 V_{e_1}  + 3 V_{e_2}  + 2 V_{e_3}  +  V_{e_4} }{6 V_1  + 3 V_2  + 2 V_3  +  V_4 }$
<b>peg</b>	acumulado da gravidade das vítimas encontradas sobre o acumulado dos valores de gravidade de todas as vítimas	$peg = \frac{\sum_{i=1}^{ V_e } grav_i}{\sum_{j=1}^{ V } grav_j}$

Tabela 3: Métricas para avaliação de desempenho dos agentes de busca de vítimas.

## 5.3 Métricas de Resgate

As métricas de desempenho vítimas socorridas são as seguintes (calculadas pelo simulador).

Sigla	Descrição	Fórmula
<b>pvs</b>	porcentual de vítimas socorridas (receberam kit de socorro)	$pvs =  V_s / V $
<b>pts</b>	porcentual do tempo de socorro utilizado	$pts = t_s/T_s$
<b>vsg</b>	vítimas socorridas ponderada por classe de gravidade. Retrata a capacidade do agente em socorrer vítima em estado mais grave, daí o uso da ponderação mais alta para as vítimas mais graves.	$vsg = \frac{6 V_{s_1}  + 3 V_{s_2}  + 2 V_{s_3}  +  V_{s_4} }{6 V_1  + 3 V_2  + 2 V_3  +  V_4 }$
<b>psg</b>	acumulado da gravidade das vítimas socorridas sobre o acumulado dos valores de gravidade de todas as vítimas	$psg = \frac{\sum_{i=1}^{ V_s } grav_i}{\sum_{j=1}^{ V } grav_j}$

Tabela 4: Métricas para avaliação de desempenho dos agentes de busca de vítimas.