

ANO LETIVO 2019/2020

# Introdução à Inteligência Artificial

## **Bomberman**

Renato Valente - 89077

Jacinto Lufilakio - 89162



4 de Dezembro de 2019

# 1. Índice

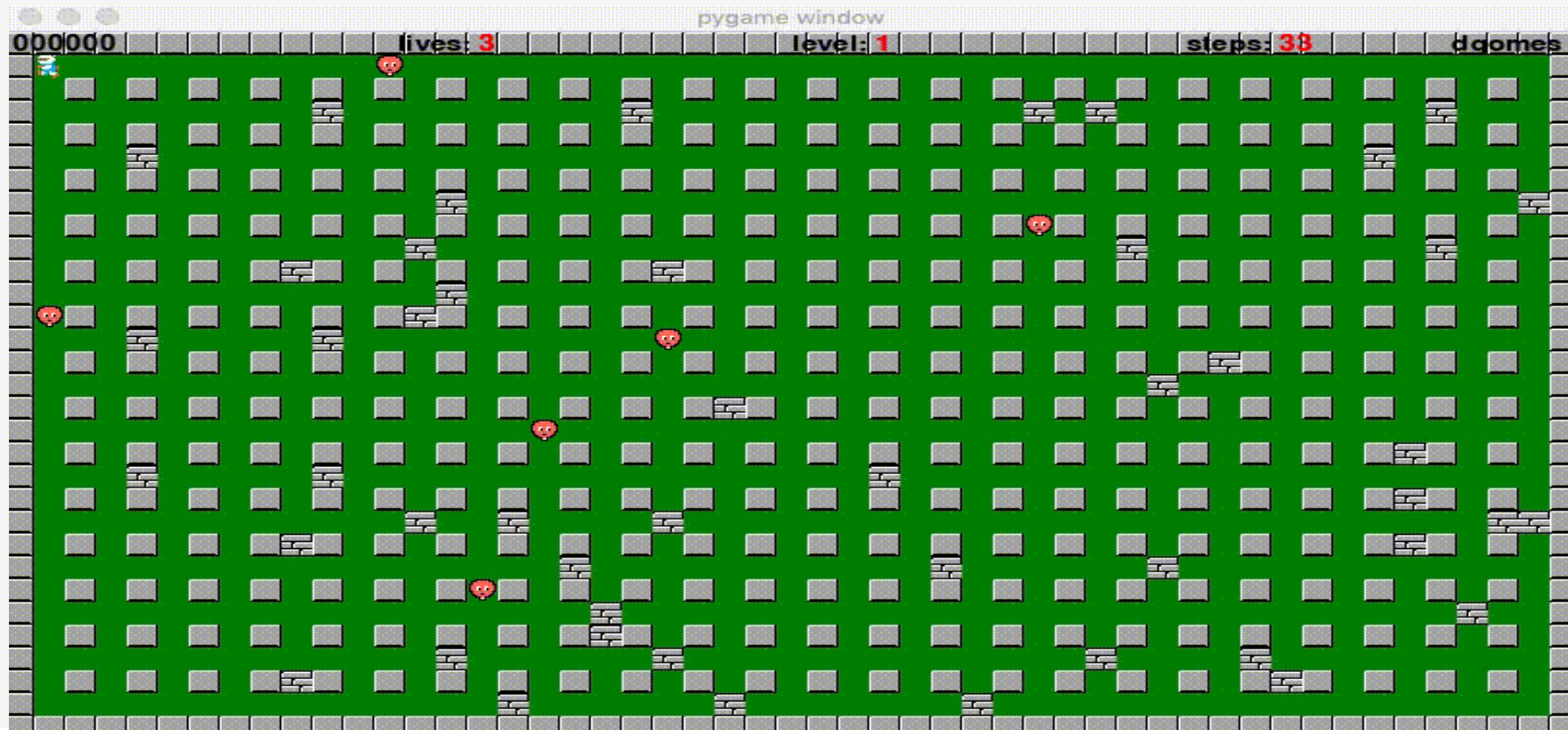
---

<u>1. Índice-----</u>	<u>2</u>
<u>2. Introdução-----</u>	<u>3</u>
<u>3. Implementação-----</u>	<u>4</u>
<u>3.1 Classes e Funções-----</u>	<u>5</u>

## 2. Introdução

Este trabalho tem como objetivo o desenvolvimento de código para a resolução de um problema de inteligência artificial, em que nos foi proposto solucionar um jogo (Bomberman) com vários níveis, em que o objetivo principal é derrotar os inimigos sem que eles nos derrotem e encontrar a saída do jogo que está escondida atrás de uma parede destrutível.

Repositório: <https://github.com/detiuaveiro/trabalho-de-grupo-bomberman-bomberman-ia-89077-89162>



# 3. Implementação

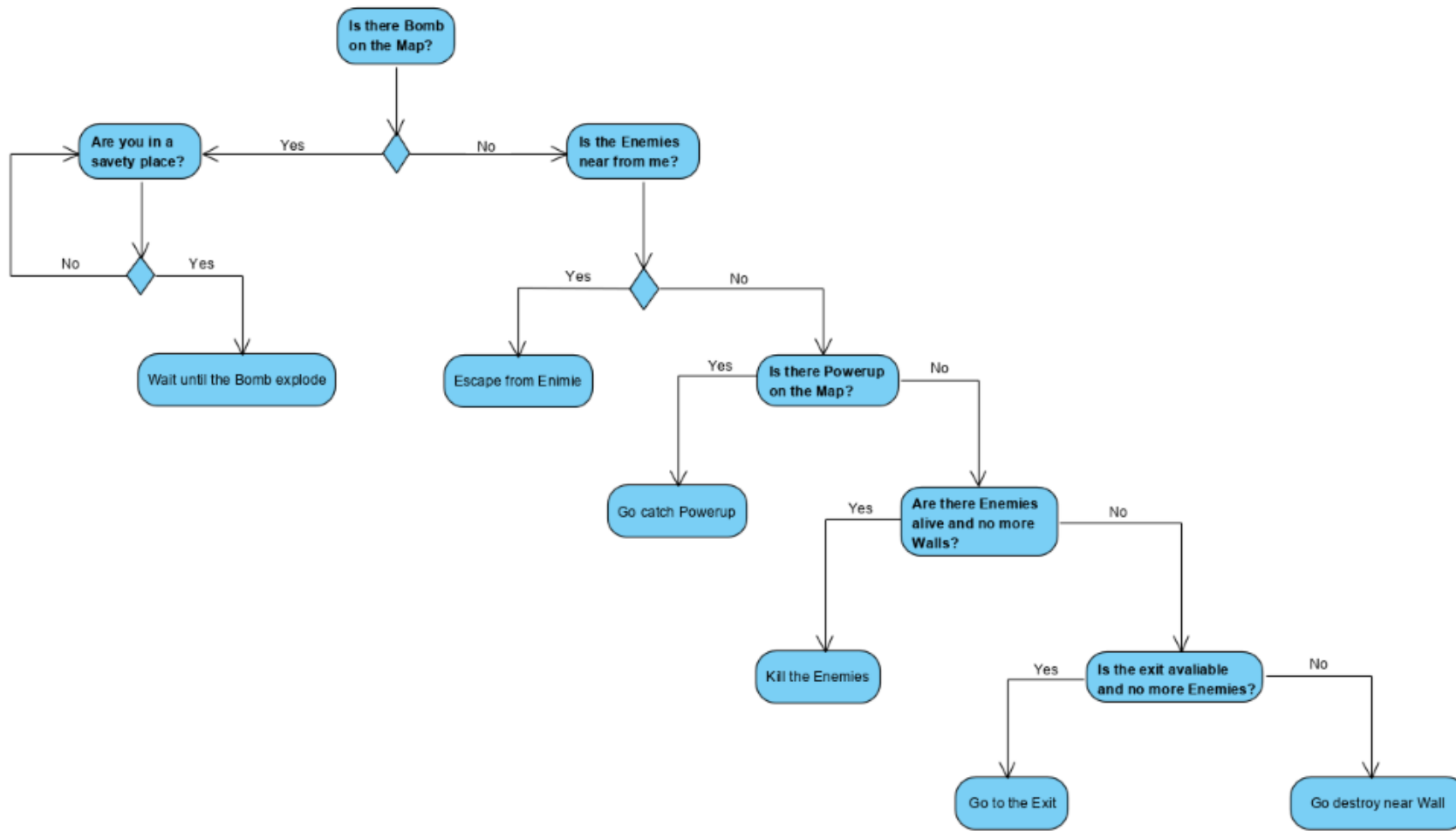


Imagem 1: esquema de decisão do agente

## 3.1 Classes e Funções

---

Estamos a utilizar uma pesquisa em árvore (através da classe `tree_search_bomb.py`), usando o método de pesquisa *greedy*. O domínio para a pesquisa é gerado numa função da classe `student.py` chamada *domain* que em seguida cria um domínio do problema através da classe `paredes.py`.

Funções importantes:

- `bomb_fled()`: -Cria vários concorrentes a posições livres (dando a coordenada de uma bomba e o raio) que posteriormente será verificada através da função `verify_range_bomb()`;
- `verify_range_bomb()`: -Verifica se a posição dada está fora do raio de atuação da bomba;
- `pos_last()`: - Armazena os steps em que o agente fica parado no mesmo sitio e tenta sair desse estado;
- `count_close()`: - É usado pra contar o numero de steps em que ele fica a tentar matar um inimigo, se esse numero ultrapassar 200 ele envia teclas aleatórias (*random keys*) para fazer com que o inimigo não o faça entrar em *loop*.