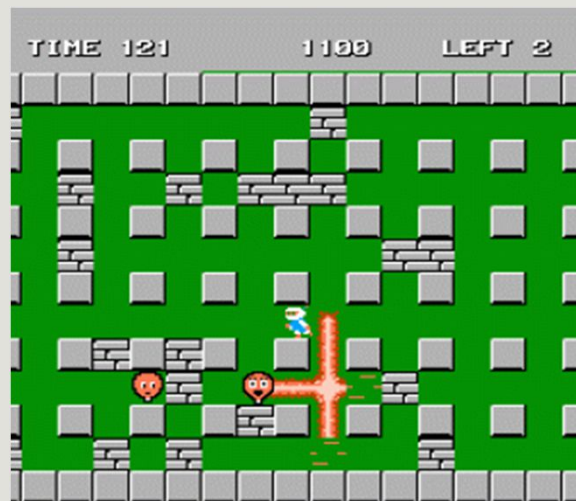


# BOMBERMAN



---

Trabalho Realizado:

Henrique Silva, 88857

Diogo Correia, 90327

IIA, MIECT

“bomberman-ia-88857-90327”

# Walls

---

- closest\_entity -> Parametros de entrada são a posição do bomberman e array de entidades. Retorna a posição [x,y] da entidade mais próxima (*enemie ou wall*).
- deploy bomb -> Usando a variável *deploy\_bomb\_counter*, permite ao bomberman fugir em forma de "L" após colocar uma bomba. Através de funções dentro da deploy bomb (*away\_from\_wall*, *is\_wall..*) a função indica qual a key mais apropriada à situação.
  - Como deploy bomb necessita ser chamada 8 ou 10 vezes seguidas, foi criada uma flag *after\_deploy*, que permite distinguir as diferentes situações, ficando false após explosão da bomba.
- *is\_stuck\_flag* -> Foi criado um algoritmo que verifica se o bomberman ficou stuck, e caso seja True, ele coloca uma bomba, e a função deploy bomb entra em ação, através da variável *after\_deploy*.

# Enemies

- 
- Através de states, são distinguidos os diferentes alvos:
    - State0 -> Oneal
    - State1 -> Balloom
    - State2 -> Wall
    - State3 -> Doll
  - Se ao procurar as walls encontrar algum enemy, o bomberman tenta matá-lo, usando a função in\_range para verificar se o inimigo se encontra ao alcance da bomba.
  - Após destruir todas as walls, se ainda houver ballooms vai para o canto e fica à espera que eles lá cheguem para os matar. Isto é verificado através do boolean corner\_killing.
  - Se após matar os ballooms ainda houverem outros inimigos, o bomberman vai atrás deles, definindo como destiny o closest\_entity dos inimigos restantes.
  - Após todos os inimigos mortos, e ter apanhado o powerup(has\_powerup, powerup\_discover, powerup\_pickedup), o seu destiny toma como valor o state[exit].