

ANDREI GARCÍA CUADRA
APRENDIZAJE AUTOMÁTICO
TUTORIAL I
NIA: 100405804

1. Inicialización.....	3
2. Modos de ejecución.....	8
3. Respuestas a enunciado	12
3.1. ¿Qué información se muestra en la interfaz? ¿Y en la terminal? ¿Cuál es la posición que ocupa Pac-Man inicialmente?.....	12
3.2. Según tu opinión, ¿qué datos podrían ser útiles para decidir lo que tiene que hacer Pac-Man en cada momento?	15
3.3. Revisa la carpeta layouts. ¿Cómo están definidos los laberintos en estos ficheros? Diseña un laberinto nuevo, guárdalo y ejecútalo en el juego. Incluye una captura de pantalla del mismo en la memoria	16
3.4. Ejecuta el agente BasicAgent aa tal y como se indica a continuación y Describe qué información se muestra por pantalla sobre el estado del juego en cada turno. De esta información, ¿cuál crees que podría ser más útil para decidir automáticamente la siguiente acción de Pac-Man?	19
3.5. Programa una función de nombre printLineData() dentro del agente BasicAgent AA del fichero busterAgents.py. Esta función debe devolver una cadena de caracteres con la información que se considere relevante del estado del Pac-Man. Dicha función será llamada desde el bucle principal del juego en Game.py para que se escriba en un fichero. Este paso es de gran importancia ya que servirá como primera versión de la fase de extracción de características y será imprescindible para las siguientes prácticas. Por cada turno de juego, se debe guardar una línea con todos los datos concatenados del estado del juego que se calculan por omisión, separados por el carácter coma (,). Además, cada vez que se inicie una nueva partida o se abra el juego de nuevo, las nuevas líneas deben guardarse debajo de las antiguas. No se debe reiniciar el fichero de texto al empezar una nueva partida, por lo que el fichero de texto resultante tendrá tantas líneas como turnos se hayan jugado en todas las partidas.	21
3.6. Programa un comportamiento “inteligente” para el Pac-Man. Para ello, en la clase BasicAgent AA, se pide modificar el método chooseAction() que hasta ahora presentaba un comportamiento aleatorio. Pac-Man debe perseguir y comerse a todos los fantasmas de la pantalla. Compara los resultados ejecutando el juego con los fantasmas estáticos y en movimiento aleatorio (-g RandomGhost). Haz varias ejecuciones y determina cuántos turnos de media suele tardar en finalizar.....	23
3.7. El agente programado en el ejercicio anterior no utiliza ninguna técnica de aprendizaje automático. ¿Qué ventajas crees que puede tener el aprendizaje automático para controlar a Pac-Man?	27
4. Conclusiones.....	28

1. INICIALIZACIÓN

En primer lugar procedemos a la instalación del entorno para poder ejecutar pacman:

```
conda create -name pacman python=2.7
```

```

andreigarcia -- zsh -- 80x24

ca-certificates-2021 | 115 KB    ######| 100%
certifi-2021.10.8   | 152 KB    ######| 100%
zlib-1.2.11          | 96 KB     ######| 100%
ncurses-6.3          | 856 KB    ######| 100%
libcxx-12.0.0        | 805 KB    ######| 100%
pip-21.2.4           | 1.8 MB    ######| 100%
openssl-1.1.1m       | 2.2 MB    ######| 100%
tk-8.6.11            | 3.0 MB    ######| 100%
setuptools-58.0.4    | 791 KB    ######| 100%
sqlite-3.37.0         | 1.2 MB    ######| 100%
python-3.8.12         | 10.3 MB   ######| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate pacman
#
# To deactivate an active environment, use
#
#     $ conda deactivate
(base) andreigarcia@MacBook-Pro-de-Andrei ~ %

```

Para posteriormente hacer uso del mismo:

```
conda activate pacman
```

* Denotamos la versión 2.7 para hacer funcionar el mismo, según indica el manual.

```
[andreigarcia: ~ -zsh - 80x24] certifi-2021.10.8 | 152 KB | ##### | 100% zlib-1.2.11 | 96 KB | ##### | 100% ncurses-6.3 | 856 KB | ##### | 100% libcxx-12.0.0 | 805 KB | ##### | 100% pip-21.2.4 | 1.8 MB | ##### | 100% openssl-1.1.1m | 2.2 MB | ##### | 100% tk-8.6.11 | 3.0 MB | ##### | 100% setuptools-58.0.4 | 791 KB | ##### | 100% sqlite-3.37.0 | 1.2 MB | ##### | 100% python-3.8.12 | 10.3 MB | ##### | 100% Preparing transaction: done Verifying transaction: done Executing transaction: done # # To activate this environment, use # $ conda activate pacman # # To deactivate an active environment, use # $ conda deactivate [(base) andreigarcia@MacBook-Pro-de-Andrei ~ % conda activate pacman (pacman) andreigarcia@MacBook-Pro-de-Andrei ~ %]
```

Instalamos numpy para verificar el funcionamiento de conda;

`conda install numpy`

```
andreigarcia — conda install numpy — 80x24

intel-openmp-2021.4.0          hecd8cb5_3538      961 KB
mkl-2021.4.0                   hecd8cb5_637       101.0 MB
mkl-service-2.4.0              py38h9ed2024_0    45 KB
mkl_fft-1.3.1                  py38h4ab4a9b_0    165 KB
mkl_random-1.2.2               py38hb2f4e1b_0    273 KB
numpy-1.21.2                   py38h4b4dc7a_0    23 KB
numpy-base-1.21.2              py38he0bd621_0   4.6 MB
-----
                                         Total:        107.0 MB

The following NEW packages will be INSTALLED:

blas                      pkgs/main/osx-64::blas-1.0-mkl
intel-openmp                pkgs/main/osx-64::intel-openmp-2021.4.0-hecd8cb5_3538
mkl                       pkgs/main/osx-64::mkl-2021.4.0-hecd8cb5_637
mkl-service                pkgs/main/osx-64::mkl-service-2.4.0-py38h9ed2024_0
mkl_fft                    pkgs/main/osx-64::mkl_fft-1.3.1-py38h4ab4a9b_0
mkl_random                 pkgs/main/osx-64::mkl_random-1.2.2-py38hb2f4e1b_0
numpy                      pkgs/main/osx-64::numpy-1.21.2-py38h4b4dc7a_0
numpy-base                 pkgs/main/osx-64::numpy-base-1.21.2-py38he0bd621_0
six                        pkgs/main/noarch::six-1.16.0-pyhd3eb1b0_0

Proceed ([y]/n)?
```

APRENDIZAJE AUTOMÁTICO

Nos desplazamos a la carpeta de pacman donde se ha descargado el código fuente disponible desde AulaGlobal para la ejecución del software:

```
cd /Users/andreigarcia/Documents/Formación/UC3M/2021-2022/C2/Aprendizaje\ au-  
tomático/pacman
```

```
pacman -- zsh -- 80x24

Downloading and Extracting Packages
setup-tools-44.0.0      | 513 KB      | #####| 100%
python-2.7.18            | 9.3 MB      | #####| 100%
certifi-2020.6.20        | 155 KB      | #####| 100%
pip-19.3.1               | 1.7 MB      | #####| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate pacman
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(pacman) andreigarcia@MacBook-Pro-de-Andrei ~ % conda activate pacman

[(pacman) andreigarcia@MacBook-Pro-de-Andrei ~ % cd /Users/andreigarcia/Documents]
[Formación/UC3M/2021-2022/C2/Aprendizaje\ automático/pacman
(pacman) andreigarcia@MacBook-Pro-de-Andrei pacman % ]
```

Y ejecutamos el juego. En este caso, visualizaremos un error:

[python busters.py](#)

```
pacman -- zsh -- 80x24
certifi-2020.6.20 | 155 KB | ##### | 100%
pip-19.3.1 | 1.7 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate pacman
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(pacman) andreigarcia@MacBook-Pro-de-Andrei ~ % conda activate pacman

[(pacman) andreigarcia@MacBook-Pro-de-Andrei ~ % cd /Users/andreigarcia/Documents]
[Formación/UC3M/2021-2022/C2/Aprendizaje\ automático/pacman]
[(pacman) andreigarcia@MacBook-Pro-de-Andrei pacman % python busters.py
Traceback (most recent call last):
  File "busters.py", line 25, in <module>
    from builtins import zip
ImportError: No module named builtins
(pacmanc) andreigarcia@MacBook-Pro-de-Andrei pacman % ]
```

Para solucionarlo y según he investigado en Stackoverflow, basta con instalar el paquete future. Dada la recomendación, lo instalaremos utilizando conda en lugar de pip:

conda install future

```
pacman -- zsh -- 80x24
- future

The following packages will be downloaded:

  package           | build
  -----            | -----
  future-0.18.2     | py27_0      650 KB
  -----
                           Total:      650 KB

The following NEW packages will be INSTALLED:

  future           pkgs/main/osx-64::future-0.18.2-py27_0

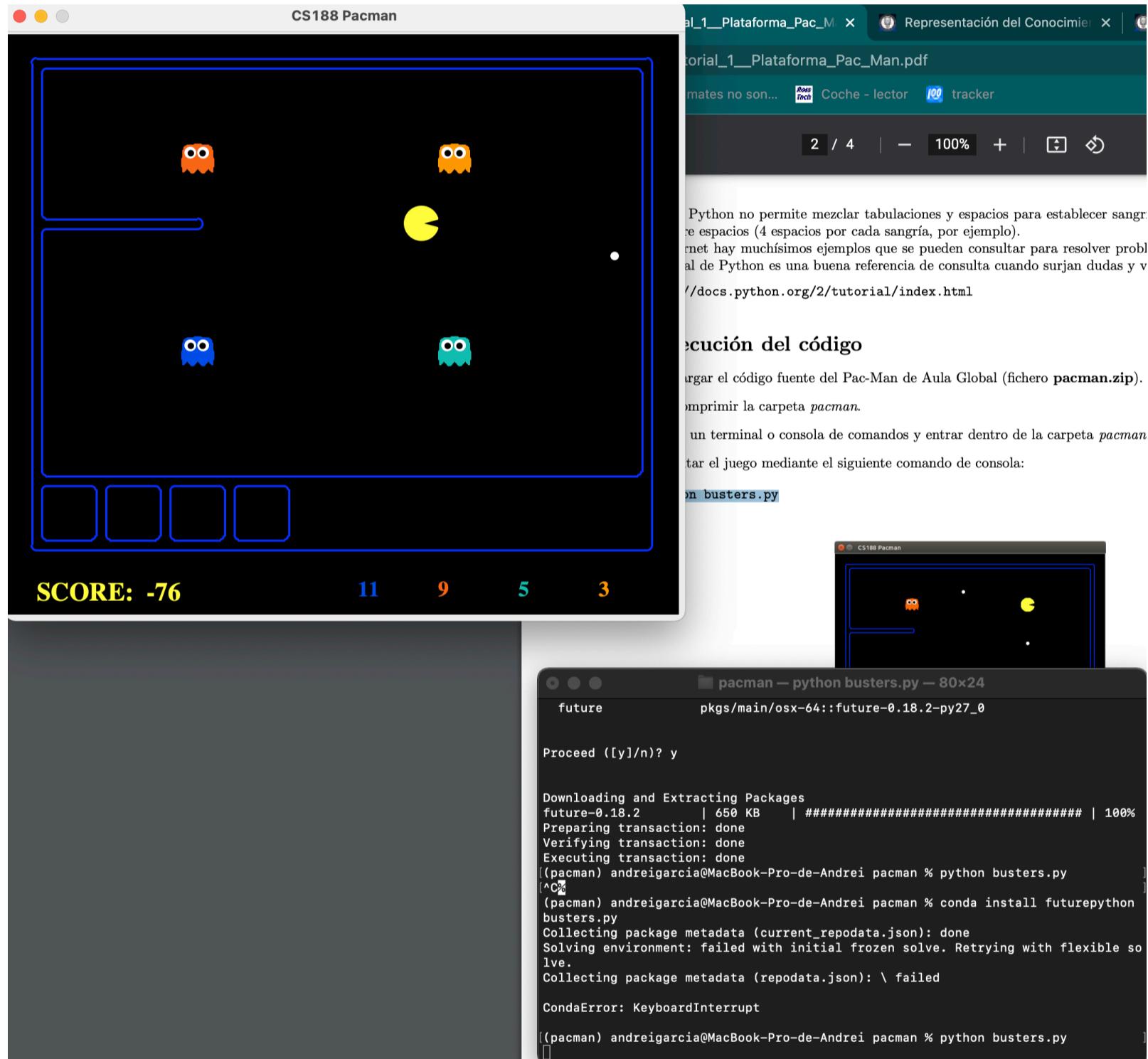
Proceed ([y]/n)? y

Downloading and Extracting Packages
future-0.18.2 | 650 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

APRENDIZAJE AUTOMÁTICO

Y ejecutamos el juego:

```
python busters.py
```



2. MÓDOS DE EJECUCIÓN

Ejecución estándar:

```
python busters.py -n 3 -l oneHunt -p RandomPAgent -g RandomGhost -k 4 -t 0.2
```

- **-h, —help** Muestra el menú de ayuda.
- **-q, —quietTextGraphics** No mostrar trazas *verbose* en la interfaz gráfica.
- **-f, —fixRandomSeed** Usa valores aleatorios reutilizados para jugar la misma partida reiteradamente.
- **-s, —showGhosts** Muestra los fantasmas en la pantalla.
- **-z ZOOM_VALUE, —numGames=ZOOM_VALUE** Acercar la vista de juego.
 - Tipo: Float
 - Valor por defecto: 1.0
 - Valor máximo: No definido.
 - Valor mínimo: 0.1
- **-n GAMES, —numGames=GAMES** El número de juegos.
 - Tipo: Entero
 - Valor por defecto: 1
 - Valor máximo: No definido.
 - Valor mínimo: 1
- **-l LAYOUT_FILE, —layout=LAYOUT_FILE** El tablero del juego.
 - Tipo: Enum
 - Valor por defecto: oneHunt
 - Valores disponibles:
 - A. 20Hunt
 - B. bigHunt
 - C. capsuleClassic
 - D. classic
 - E. contestClassic
 - F. mediumClassic
 - G. Minimap
 - H. minimaxClassic
 - I. newMap
 - J. oneHunt
 - K. openClassic
 - L. openHunt
 - M. originalClassic

- N. sixHunt
- O. smallClassic
- P. smallHunt
- Q. testClassic
- R. trappedClassic
- S. trickyClassic

• **-p TYPE, —pacman=TYPE** El tipo de agente Pac-Man (módulo pacmanAgents).

- Tipo: Enum
- Valor por defecto: BustersKeyboardAgent
- Valores disponibles:
 - A. BustersKeyboardAgent (*Control por teclado*)
 - B. RandomPAgent (*Aleatorio*)

• **-g TYPE** El tipo de agente de fantasma.

- Tipo: Enum
- Valor por defecto: RandomGhost
- Valores disponibles:
 - A. RandomGhost (*Aleatorio*)
 - B. StaticGhost (*Estático*)
 - C. DirectionalGhost (*Fantasma que prioriza comer a Pacman*).

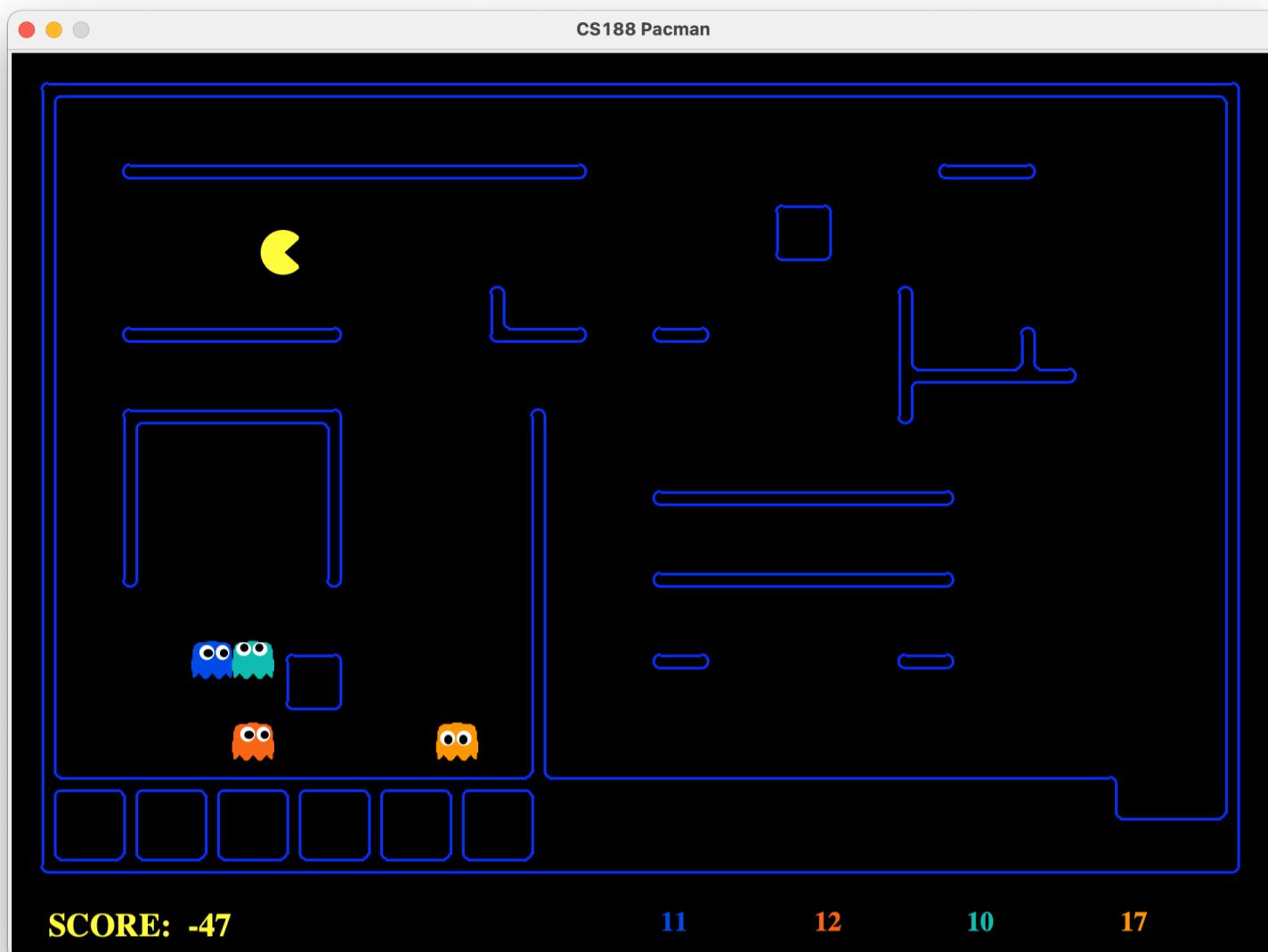
• **-k NUMGHOSTS** El número máximo de fantasmas.

- Tipo: Entero
- Valor por defecto: 4
- Valor máximo: 4
- Valor mínimo: 1

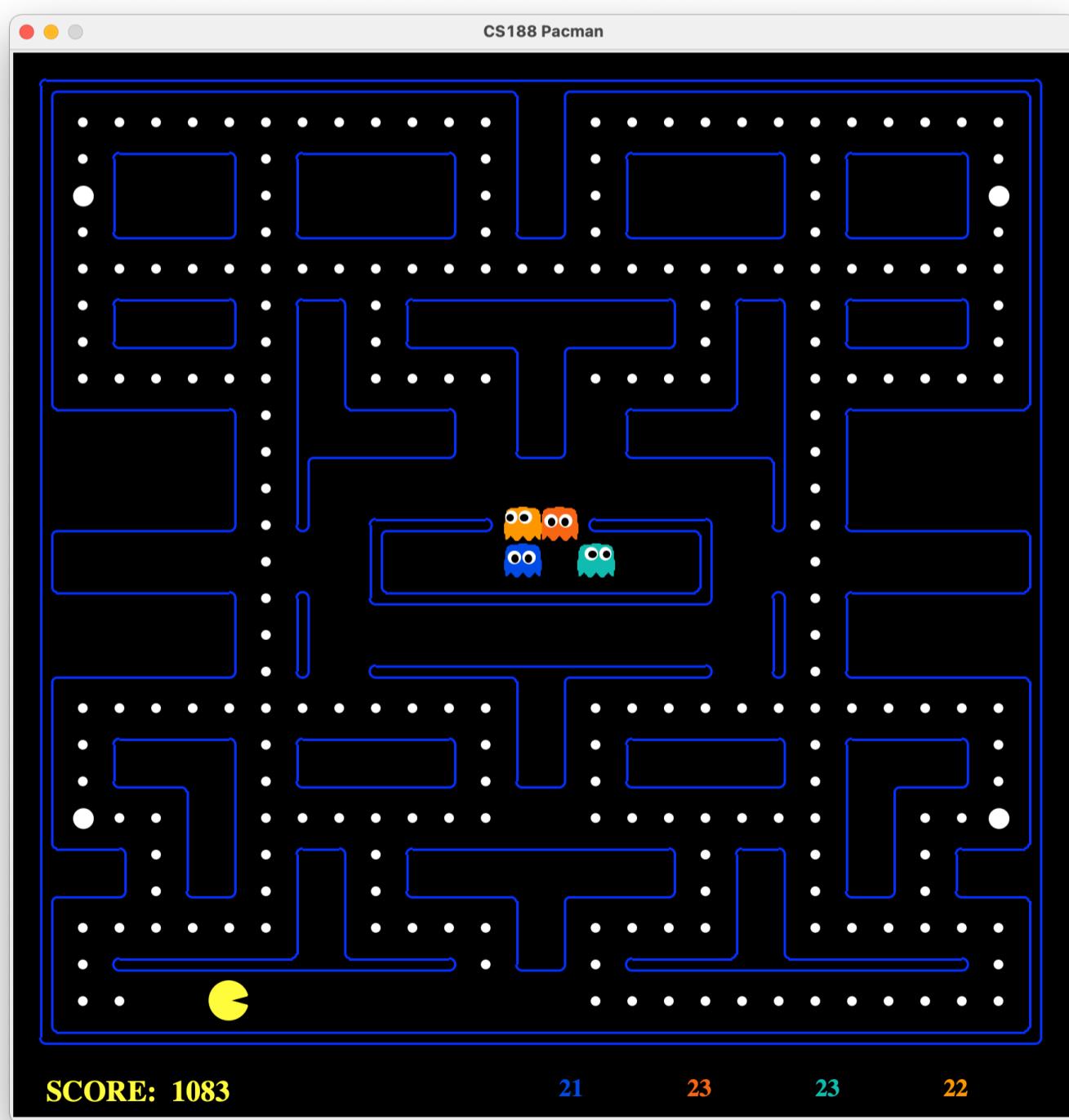
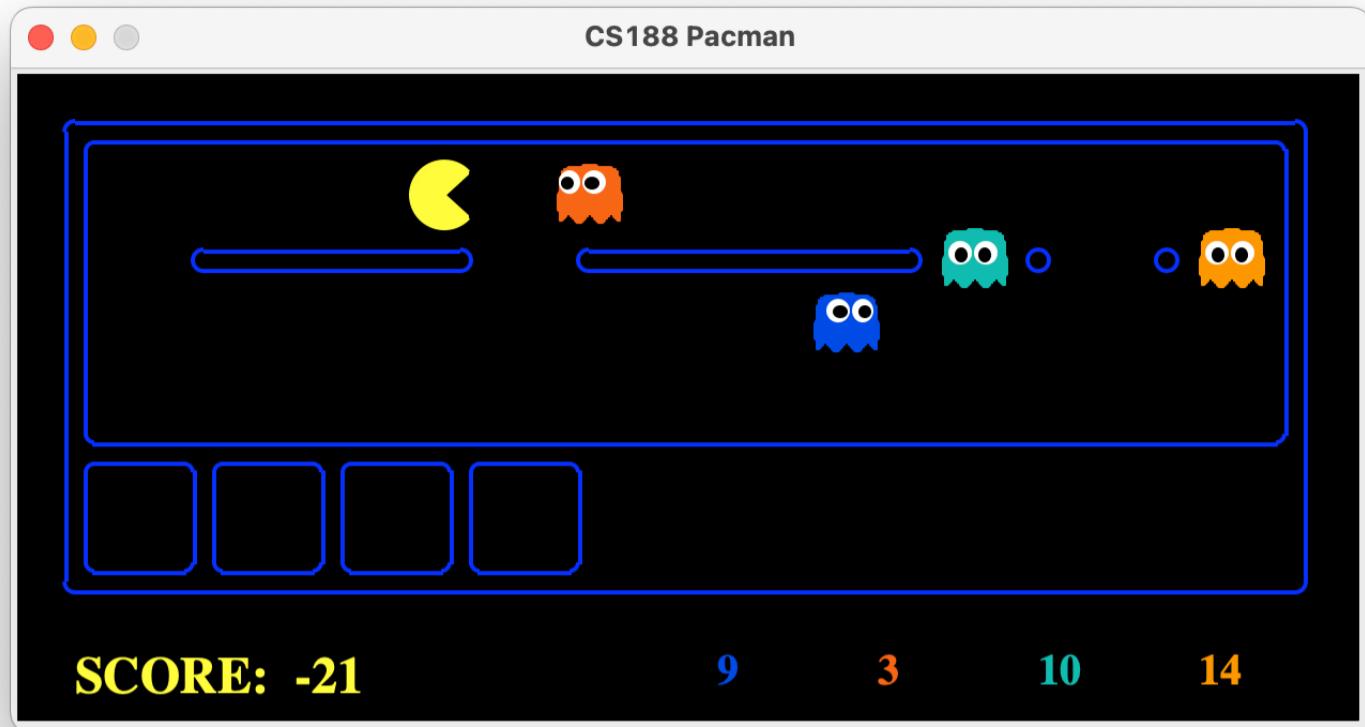
• **-t SECONDS** Tiempo de delay entre frames.

- Tipo: Entero
- Valor por defecto: 0.1
- Valor máximo: No definido.
- Valor mínimo: 0.1; en caso de ser menor que 0, utiliza el teclado.

APRENDIZAJE AUTOMÁTICO



APRENDIZAJE AUTOMÁTICO



3. RESPUESTAS A ENUNCIADO

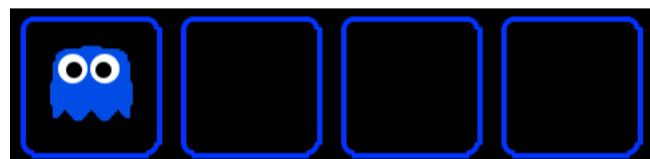
3.1. ¿QUÉ INFORMACIÓN SE MUESTRA EN LA INTERFAZ? ¿Y EN LA TERMINAL? ¿CUÁL ES LA POSICIÓN QUE OCUPA PAC-MAN INICIALMENTE?

Por pantalla se muestran los siguientes datos:

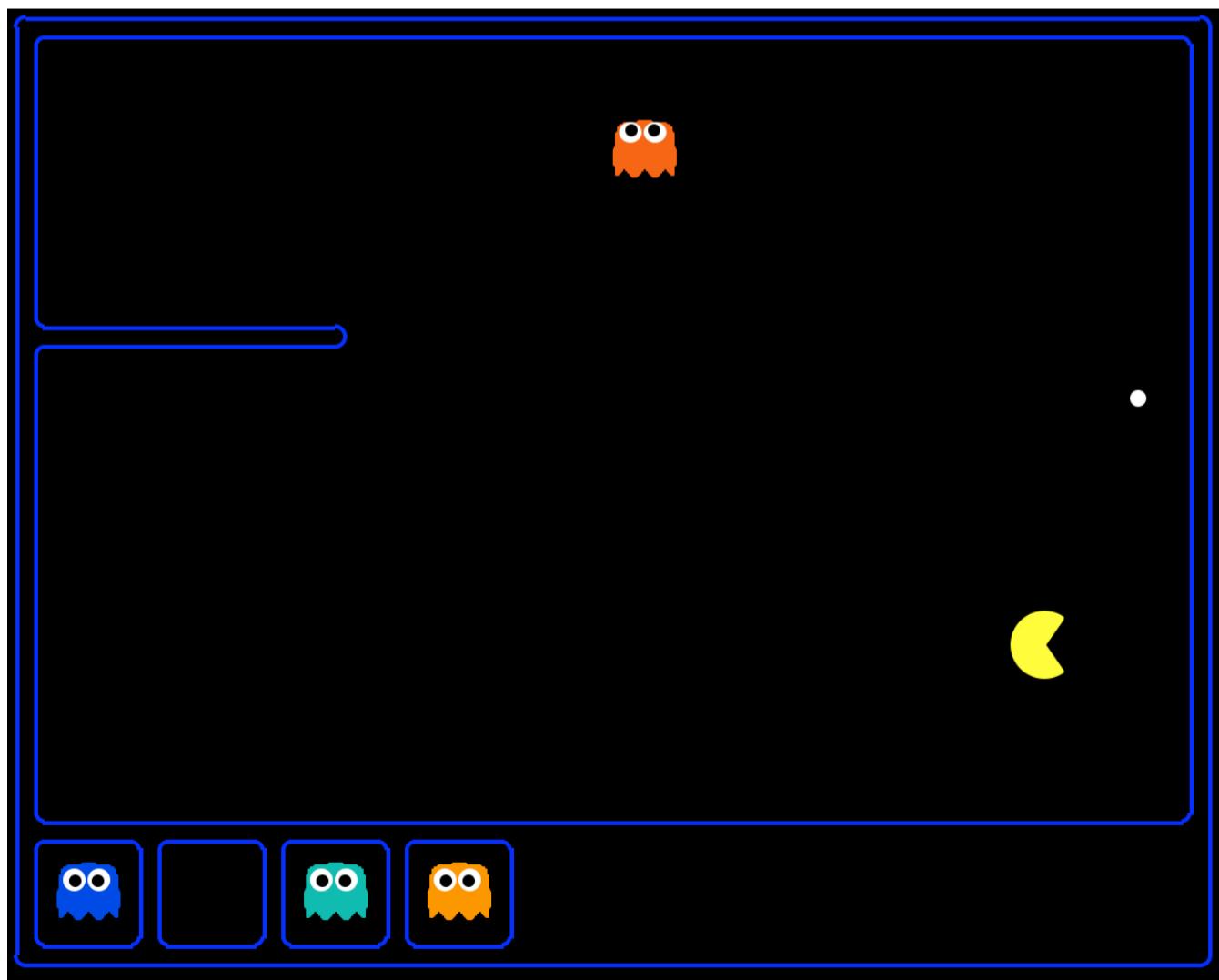
- **Puntuación:** Que muestra la diferencia entre el tiempo transcurrido y los puntos obtenidos, y puede ser negativa.

SCORE: 122

- **Fantasma comido:** Cajas con los fantasmas que han sido comidos y no se muestran en el mapa.



- **Mapa:** Visualización de fantasmas, pasman, comida y límites de movimiento (vallas) del mapa.



APRENDIZAJE AUTOMÁTICO

- **Posicionamiento de fantasmas:** Celdas de distancia de los fantasmas a Pacman. En caso de ya estar comidos, muestra 1. El color de cada número identifica el fantasma.

1	9	1	1
---	---	---	---

Por la consola se muestra:

- **Distancer:** Trazas de log cuando el fantasma se saldría del mapa, redirige a otro punto para continuar en los límites del mismo.
- **REMOVE:** Cuando finaliza un juego.
- **Average Score:** Puntuación media entre todas las partidas disputadas.
- **Scores:** Listado de puntuaciones de las partidas de modo individual.
- **Win Rate:** Porcentaje de victoria de partidas.
- **Record:** Listado con victorias/derrotas (rachas).

```
pacman -- zsh -- 80x24
File "/Users/andreigarcia/Documents/Formación/UC3M/2021-2022/C2/Aprendizaje automático/pacman/graphicsUtils.py", line 229, in edit
    _canvas.itemconfigure(id, **dict(args))
File "/Users/andreigarcia/opt/miniconda3/envs/pacman/lib/python2.7/lib-tk/Tkinter.py", line 2414, in itemconfigure
    return self._configure(('itemconfigure', tagOrId), cnf, kw)
File "/Users/andreigarcia/opt/miniconda3/envs/pacman/lib/python2.7/lib-tk/Tkinter.py", line 1327, in _configure
    self.tk.call(_flatten((self._w, cmd)) + self._options(cnf))
_tkinter.TclError: invalid command name ".140462879393344"
[pacman] andreigarcia@MacBook-Pro-de-Andrei pacman % python busters.py -n 3 -l o
neHunt -p RandomPAgent -g RandomGhost -k 4 -t 0.2
[Distancer]: Switching to maze distances
[]
(pacman) andreigarcia@MacBook-Pro-de-Andrei pacman %
[(pacman) andreigarcia@MacBook-Pro-de-Andrei pacman % python busters.py -n 3 -l o
neHunt -p RandomPAgent -g RandomGhost -k 4 -t 0.05
[Distancer]: Switching to maze distances
REMOVE
Average Score: 89.6666666667
Scores:      -142, 265, 146
Win Rate:    3/3 (1.00)
Record:      Win, Win, Win
(pacman) andreigarcia@MacBook-Pro-de-Andrei pacman %
```

APRENDIZAJE AUTOMÁTICO

Pacman se encuentra inicialmente en la posición $x = 19$; $y = 9$ y orientación sur. Se ha obtenido la siguiente información con el debugger de PyCharm.

The screenshot shows the PyCharm debugger interface with a hierarchical tree view of variable states. At the top, 'agentIndex' is set to 0. Below it, 'state' is expanded to show 'data', which contains 'agentStates'. The first item in 'agentStates' is '0', representing Pacman, with its position listed as '(x,y)=(12, 9), South'. Other items in 'agentStates' represent three ghosts at positions (5, 6), (5, 12), and (13, 6), all in 'Stop' mode. The bottom of the window shows tabs for 'Debug' (selected), 'Terminal', 'Python Packages', and 'Python Console'.

```
01 agentIndex = {int} 0
\-\ state = {GameState} %%%%%%%%
\-\ \-\ data = {GameStateData} %%%%%%%%
\-\ \-\ \-\ agentStates = {list: 5} [<game.AgentState object at 0x7fe57823c8d0>, <game.Ag
\-\ \-\ \-\ \-\ 0 = {AgentState} Pacman: (x,y)=(12, 9), South
\-\ \-\ \-\ \-\ \-\ configuration = {Configuration} (x,y)=(12, 9), South
\-\ \-\ \-\ \-\ \-\ 01 isPacman = {bool} True
\-\ \-\ \-\ \-\ \-\ 01 numCarrying = {int} 0
\-\ \-\ \-\ \-\ \-\ 01 numReturned = {int} 0
\-\ \-\ \-\ \-\ \-\ 01 scaredTimer = {int} 0
\-\ \-\ \-\ \-\ \-\ start = {Configuration} (x,y)=(12, 10), Stop
\-\ \-\ \-\ \-\ \-\ 1 = {AgentState} Ghost: (x,y)=(5, 6), Stop
\-\ \-\ \-\ \-\ \-\ 2 = {AgentState} Ghost: (x,y)=(5, 12), Stop
\-\ \-\ \-\ \-\ \-\ 3 = {AgentState} Ghost: (x,y)=(13, 6), Stop
```

Nota: Para identificar la posición inicial de Pacman se utiliza la letra “P” en el mapa. Ver pregunta 3.3.

3.2. SEGÚN TU OPINIÓN, ¿QUÉ DATOS PODRÍAN SER ÚTILES PARA DECIDIR LO QUE TIENE QUE HACER PACMAN EN CADA MOMENTO?

1. La posición actual.
2. La distancia al fantasma más cercano.
3. La distancia a las paredes (a fin de atrapar fantasmas).
4. Si se encuentra en modo “inmune” (que se da después de comer el punto blanco).
5. La cantidad de fantasmas pendientes de matar.
6. La velocidad de Pacman.
7. La velocidad de los fantasmas.
8. El tiempo de inmunidad restante.
9. El tamaño del tablero.

3.3. REVISA LA CARPETA LAYOUTS. ¿CÓMO ESTÁN DEFINIDOS LOS LABERINTOS EN ESTOS FICHEROS? DISEÑA UN LABERINTO NUEVO, GUÁRDALO Y EJECÚTALO EN EL JUEGO. INCLUYE UNA CAPTURA DE PANTALLA DEL MISMO EN LA MEMORIA

Los laberintos están definidos por un nombre (que será el necesario para el argumento -l para inicializar el juego) con extensión .lay (que será necesario para que el programa principal de python lo detecte). Es sensible a mayúsculas y minúsculas.

Para los mapas se utiliza un lenguaje nemotécnico, que utiliza un lenguaje sustitutivo con el siguiente significado (sensible a mayúsculas y minúsculas):

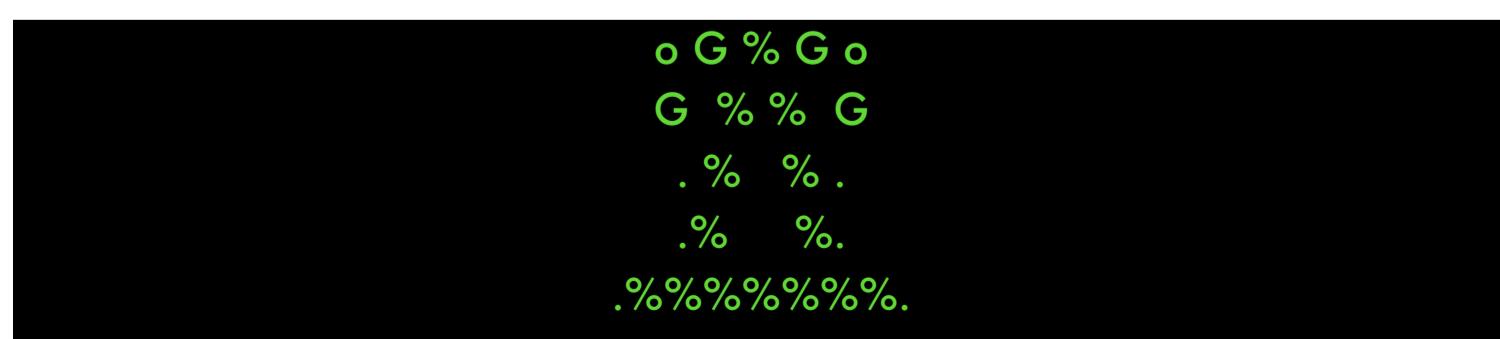
- G: Posición inicial de fantasma.
- P: Posición inicial Pacman.
- %: Pared (muro) no transitable.
- o: Posición de comida de inmunidad (asusta a los fantasmas).
- .: Posición de bola de puntuación.
- _ (*espacio en blanco*): Casilla transitable vacía.

Para ello, lee el archivo y sustituye los caracteres con el elemento y su significado para la casilla dada (cada casilla es un carácter).

El mapa debe comenzar con un relleno de % en su primera línea y debe finalizar con los mismos caracteres. Se debe agregar una línea de relleno para mostrar los fantasmas y una línea de relleno final. El modelo resultante para crear un nuevo mapa se puede consultar en el fichero newMap.lay.

Todas las líneas deben comenzar con una pared.

Se ha diseñado el siguiente mapa, el cual es una A de Andrei simétrica:



o	G	%	G	o
G	%	%	G	
.	%	%	.	
.	%	%	.	
.	%	%	.	
.	%	%	%	.
.	%	%	%	.
.	%	P	%	.

.% %.
.% P %.

Para ejecutarlo, podemos hacerlo con el siguiente comando:

```
python busters.py -n 3 -l andrei -p RandomPAgent -g RandomGhost -k 4 -t 0.05
```

Y obtenemos como resultado en la primera ejecución:

```
agent.registerInitialState(self.state.deepCopy())
File "/Users/andreigarcia/Documents/Formacion/UC3M/2021-2022/C2/Aprendizaje automatico/pacman/bustersAgents.py", line 132, in registerInitialState
    self.distancer = Distancer(gameState.data.layout, False)
File "/Users/andreigarcia/Documents/Formacion/UC3M/2021-2022/C2/Aprendizaje automatico/pacman/distanceCalculator.py", line 59, in __init__
    dc.run()
File "/Users/andreigarcia/Documents/Formacion/UC3M/2021-2022/C2/Aprendizaje automatico/pacman/distanceCalculator.py", line 137, in run
    distances = computeDistances(self.layout)
File "/Users/andreigarcia/Documents/Formacion/UC3M/2021-2022/C2/Aprendizaje automatico/pacman/distanceCalculator.py", line 168, in computeDistances
    if not layout.isWall((x,y+1)):
File "/Users/andreigarcia/Documents/Formacion/UC3M/2021-2022/C2/Aprendizaje automatico/pacman/layout.py", line 70, in isWall
    return self.walls[x][col]
IndexError: list index out of range
(pacman) andreigarcia@MacBook-Pro-de-Andrei pacman %
```

Esto se ocasiona debido a que Pacman no se puede mover. Por ende, reemplazamos las paredes por comida y verificamos la sintaxis. Finalmente obtenemos el siguiente código:

o G . G o	o G . G o
G .. G	G .. G
% . . %	% . . %
%. . %	%. . %
%.....%	%.....%
%. . %	%. . %
%. P .%	%. P .%

APRENDIZAJE AUTOMÁTICO

Pantallazo final del mapa creado:



3.4. EJECUTA EL AGENTE BASICAGENT AA TAL Y COMO SE INDICA A CONTINUACIÓN Y DESCRIBE QUÉ INFORMACIÓN SE MUESTRA POR PANTALLA SOBRE EL ESTADO DEL JUEGO EN CADA TURNO. DE ESTA INFORMACIÓN, ¿CUÁL CREEES QUE PODRÍA SER MÁS ÚTIL PARA DECIDIR AUTOMÁTICAMENTE LA SIGUIENTE ACCIÓN DE PAC-MAN?

```
python busters.py -p BasicAgentAA
```

Traza de información obtenida:

```
————— TICK 518 —————
Width: 20 Height: 16
Pacman position: (11, 13)
Legal actions: ['West', 'Stop', 'East', 'North', 'South']
Pacman direction: South
Number of ghosts: 4
Living ghosts: [False, True, False, False]
Ghosts positions: [(5, 6), (3, 1), (5, 1), (7, 1)]
Ghosts directions: ['Stop', 'Stop', 'Stop', 'Stop']
Ghosts distances: [13, None, None, None]
Pac dots: 1
Distance nearest pac dots: 11
Map:
TTTTTTTTTTTTTTTTTT
TFFFFFFF FFFFFFFFT
TFFFFFFF FFFFFFFFT
TFFFFFFF FFFFFFFFT
TFFFFFFF FFFFFFFFT
TTTTTT FFFFFFFFT
TFFFFFFF FFFFFFFFT
TTTTTTTTTTTTTTTT
```

TFTFTFTTTTTTTTT
TTTTTTTTTTTTTTTT
Score: 83

De la información anterior, denotamos que en primer lugar se muestra el “tick” que viene a ser el número de ronda. Seguidamente, nos muestra la información del alto y ancho del mapa. Esta información no es la más necesaria en el agente, ya que con saber dónde se puede mover es suficiente.

Prosigue con la posición del agente (en este caso pacman) con coordenadas x e y. Las acciones legales están asociadas al punto anterior, ya que en caso de estar colindando con una pared no permitirá desplazarse en dicha dirección. La dirección del tick anterior es la que se muestra en la siguiente línea. De este párrafo la información más útil es la posición cartesiana.

Las líneas venideras nos otorgan información sobre los fantasmas: sus posiciones, sus direcciones y cuáles de ellos están vivos. De dicha sección lo más útil es la distancia a (el/los) fantasma(/s) que estén a menos de un umbral de “seguridad”, dado que no deseamos tenerlos cerca (salvo que tengamos el momento de inmunidad activo).

Las últimas líneas de traza nos otorgan los puntos que ya se han comido, la distancia al punto más cercano y las paredes del mapa. Para finalizar, nos muestra nuestra puntuación actual. De este contenido nos resulta útil tan sólo la distancia al punto más cercano (estrategia eager “pájaro en mano”), dado que las paredes se calculan en el movimiento albergado en el párrafo dos de esta sección y no necesitamos más información relativa a la posición, pero sí requerimos la información relativa al punto más cercano para priorizar el siguiente movimiento en función de la estrategia del agente.

3.5. PROGRAMA UNA FUNCIÓN DE NOMBRE PRINTLINEDATA() DENTRO DEL AGENTE BASICAGENT AA DEL FICHERO BUSTERAGENTS.PY. ESTA FUNCIÓN DEBE DEVOLVER UNA CADENA DE CARACTERES CON LA INFORMACIÓN QUE SE CONSIDERE RELEVANTE DEL ESTADO DEL PAC-MAN. DICHA FUNCIÓN SERÁ LLAMADA DESDE EL BUCLE PRINCIPAL DEL JUEGO EN GAME.PY PARA QUE SE ESCRIBA EN UN FICHERO. ESTE PASO ES DE GRAN IMPORTANCIA YA QUE SERVIRÁ COMO PRIMERA VERSIÓN DE LA FASE DE EXTRACCIÓN DE CARACTERÍSTICAS Y SERÁ IMPRESCINDIBLE PARA LAS SIGUIENTES PRÁCTICAS. POR CADA TURNO DE JUEGO, SE DEBE GUARDAR UNA LÍNEA CON TODOS LOS DATOS CONCATENADOS DEL ESTADO DEL JUEGO QUE SE CALCULAN POR OMISIÓN, SEPARADOS POR EL CARÁCTER COMA (,). ADEMÁS, CADA VEZ QUE SE INICIE UNA NUEVA PARTIDA O SE ABRA EL JUEGO DE NUEVO, LAS NUEVAS LÍNEAS DEBEN GUARDARSE DEBAJO DE LAS ANTIGUAS. NO SE DEBE REINICIAR EL FICHERO DE TEXTO AL EMPEZAR UNA NUEVA PARTIDA, POR LO QUE EL FICHERO DE TEXTO RESULTANTE TENDRÁ TANTAS LÍNEAS COMO TURNOS SE HAYAN JUGADO EN TODAS LAS PARTIDAS.

Se denota el umbral de seguridad como 5 celdas, si un fantasma se encuentra más cerca de esto, se mostrará su posición (si no, no es necesario ya que podemos ignorarlo).

Función printLineData:

```
def printLineData(self, gameState):
    securityDistance = 5 # Hardcoded security distance

    posStr = str(gameState.getPacmanPosition()[0]) + ";" + str(gameState.getPacmanPosition()[1])
    unsafeGhostsStr = ""
    unsafeGhostsNearby = False
    for idx, distance in enumerate(gameState.data.ghostDistances):
        if distance <= securityDistance and distance is not None:
            unsafeGhostsStr += str(idx) + ";"
            unsafeGhostsNearby = True
    if unsafeGhostsNearby: unsafeGhostsStr = unsafeGhostsStr[:-1] # remove last ;
    nearestDotDistanceStr = str(gameState.getDistanceNearestFood())
    # positionX;positionY,[ghostsNearby],nearestDotDistance
```

```
# [ghostsNearby] -> can be NULL or ; separated idx of the ghosts
return posStr + "," + unsafeGhostsStr + "," + nearestDotDistanceStr
```

Función printLineData: (desde el archivo game.py, guarda la traza al archivo traces.csv)

```
while not self.gameOver:
    # Fetch the next agent
    agent = self.agents[agentIndex]
    move_time = 0
    skip_action = False
    if hasattr(agent, 'printLineData'):
        f = open("traces.csv", "a")
        f.write(agent.printLineData(self.state) + "\n")
        f.close()
```

Resultado de ejecución básica:

```
pacman - traces.csv
pacman
Project - Documenti (Mac OS X) [CJM2021-2022]
pacman
|- pacman
|   |- agents
|   |   |- 20-hunt.ley
|   |   |- andrei.ley
|   |   |- andrey.ley
|   |   |- capsuleClassic.ley
|   |   |- classic.ley
|   |   |- contestClassic.ley
|   |   |- minmaxClassic.ley
|   |   |- minimaxClassic.ley
|   |   |- oneHunt.ley
|   |   |- openHunt.ley
|   |   |- openOneHunt.ley
|   |   |- openTwoHunt.ley
|   |   |- stochastic.ley
|   |   |- smallClassic.ley
|   |   |- smartClassic.ley
|   |   |- testClassic.ley
|   |   |- trapdoorClassic.ley
|   |   |- trickyClassic.ley
|   |   |- bustersAgents.py
|   |   |- bustersGhostAgents.py
|   |   |- graphicsCalculator.py
|   |   |- game.py
|   |   |- ghostAgents.py
|   |   |- grading.py
|   |   |- graphicsDisplay.py
|   |   |- graphicsUtils.py
|   |   |- inference.py
|   |   |- keyboardAgents.py
|   |   |- layout.py
|   |   |- pacman.py
|   |   |- projectAgents.py
|   |   |- randomAgents.py
|   |   |- textDisplay.py
|   |- traces.csv
|- External Libraries
|- Scratches and Consoles

Terminal - Local - + ~
self.moveToPacman(pos, self.getDirection(pacman), image)
File "/Users/andregarcia/Documents/Proyecto/IA/UCM2021-2022/C2/AprendizajeAutomaTico/pacman/graphicsDisplay.py", line 336, in movePacman
    moveCircleImage([0], self.mousePosition, r, endpoint=True)
File "/Users/andregarcia/Documents/Proyecto/IA/UCM2021-2022/C2/AprendizajeAutomaTico/pacman/graphicsUtils.py", line 225, in moveCircle
    edit((i, ('start', e0[i]), ('end', e1[i])), ('extent', e0[i]) + (e0[i],))
File "/Users/andregarcia/Documents/Proyecto/IA/UCM2021-2022/C2/AprendizajeAutomaTico/pacman/graphicsUtils.py", line 229, in edit
    _canvas.itemconfigure(id, **dict(args))
File "/Users/andregarcia/Documents/Proyecto/IA/UCM2021-2022/C2/AprendizajeAutomaTico/pacman/graphicsUtils.py", line 2414, in itemconfigure
    return _tkinter._configure(_id, _options, conf, kw)
File "/Users/andregarcia/Desktop/miniconda/anaconda/lib/python2.7/lib-tk/_tkinter.py", line 1327, in _configure
    self.tk.call(_flatten([self._w, cmd])) + self._options(conf)
_tkinter.TclError: invalid command name ".140269746702288"
self.tk.call(_flatten([self._w, cmd])) + self._options(conf)
andregarcia@MacBook-Pro-de-Andrei: ~ ]
```

12;10,2;3,7
12;11,3,8
11;11,3,9
11;10,3,8
11;9,2;3,7
12;9,2;3,6
12;10,2;3,7
12;9,2;3,6
11;9,2;3,7
10;9,,8
10;8,2,9
11;8,2,9
12;8,2,9
11;9,2;3,7
11;9,2;3,7

3.6. PROGRAMA UN COMPORTAMIENTO “INTELIGENTE” PARA EL PAC-MAN. PARA ELLO, EN LA CLASE BASICAGENT AA, SE PIDE MODIFICAR EL MÉTODO CHOOSEACTION() QUE HASTA AHORA PRESENTABA UN COMPORTAMIENTO ALEATORIO. PAC-MAN DEBE PERSEGUIR Y COMERSE A TODOS LOS FANTASMAS DE LA PANTALLA. COMPARA LOS RESULTADOS EJECUTANDO EL JUEGO CON LOS FANTASMAS ESTÁTICOS Y EN MOVIMIENTO ALEATORIO (-G RANDOMGHOST). HAZ VARIAS EJECUCIONES Y DETERMINA CUÁNTOS TURNOS DE MEDIA SUELE TARDAR EN FINALIZAR.

Código:

```
def chooseAction(self, gameState):
    self.countActions = self.countActions + 1
    self.printInfo(gameState)
    move = Directions.STOP
    legal = gameState.getLegalActions(0) ##Legal position from the pacman
    pacmanPos = gameState.getPacmanPosition()

    closestGhostDistance = None
    closestGhostPosition = None

    for idx, alive in enumerate(gameState.getLivingGhosts()):
        distance = None
        if alive:
            distance = gameState.data.ghostDistances[idx - 1]
        if distance is not None and (closestGhostDistance is None or distance <
closestGhostDistance):
            closestGhostDistance = distance
            closestGhostPosition = gameState.getGhostPositions()[idx - 1]

    # Check what do we have to prioritize: in case we moved X axis, try to move Y
    # before now, and viceversa

    pacmanDirection = gameState.data.agentStates[0].getDirection()
    print(pacmanDirection)
    # Last move was Y axis, prioritize axis X (default)
    if (pacmanDirection is None or pacmanDirection is Directions.STOP) or (
        pacmanDirection is Directions.SOUTH or pacmanDirection is Directions.NORTH):
        if closestGhostPosition[0] < pacmanPos[0]:
            move = Directions.WEST
        elif closestGhostPosition[0] > pacmanPos[0]:
            move = Directions.EAST
        elif closestGhostPosition[1] > pacmanPos[1]:
            move = Directions.NORTH
```

APRENDIZAJE AUTOMÁTICO

```
elif closestGhostPosition[1] < pacmanPos[1]:
    move = Directions.SOUTH

if pacmanDirection is Directions.EAST or pacmanDirection is Directions.WEST:
    if closestGhostPosition[1] > pacmanPos[1]:
        move = Directions.NORTH
    elif closestGhostPosition[1] < pacmanPos[1]:
        move = Directions.SOUTH
    elif closestGhostPosition[0] < pacmanPos[0]:
        move = Directions.WEST
    elif closestGhostPosition[0] > pacmanPos[0]:
        move = Directions.EAST

# Return move only if that's valid. If not, stop!
if move in legal:
    return move

# Prevent stuck stuff! Illegal movement detected. Alternate X / Y axis
if closestGhostPosition[1] > pacmanPos[1]:
    move = Directions.NORTH
elif closestGhostPosition[1] < pacmanPos[1]:
    move = Directions.SOUTH
elif closestGhostPosition[0] < pacmanPos[0]:
    move = Directions.WEST
elif closestGhostPosition[0] > pacmanPos[0]:
    move = Directions.EAST

# Return move only if that's valid. If not, stop!
if move in legal:
    return move

# Stuck stuff -> try to move to another axis (note that it's not circular or
# bijective).
# EAST -> NORTH; NORTH -> EAST; WEST -> SOUTH; SOUTH -> WEST
if pacmanDirection is Directions.EAST:
    move = Directions.NORTH
elif pacmanDirection is Directions.WEST:
    move = Directions.SOUTH
elif pacmanDirection is Directions.NORTH:
    move = Directions.EAST
elif pacmanDirection is Directions.SOUTH:
    move = Directions.WEST

# Only legal moves ahead our stochastic movements.
if move in legal:
    return move

# In case we can't move left, move right. In case we can't move up, move down.
# If this does not work, map is wrong, so we just STOP. We may just wait.
move = Directions.REVERSE[move]
if move in legal:
    return move
```

APRENDIZAJE AUTOMÁTICO

As a desperation movement. Rand it! Break the walls!! This happens when we have tons of walls nearby at the beginning.

```
move_random = random.randint(0, 3)
if (move_random == 0) and Directions.WEST in legal: return Directions.WEST
if (move_random == 1) and Directions.EAST in legal: return Directions.EAST
if (move_random == 2) and Directions.NORTH in legal: return Directions.NORTH
if (move_random == 3) and Directions.SOUTH in legal: return Directions.SOUTH

# No way. Just stop. Wait for a beer
return Directions.STOP
```

APRENDIZAJE AUTOMÁTICO

Ejecución con fantasmas estáticos: 9 turnos (las 10 ejecuciones devuelven el mismo output ya que el algoritmo es estadístico y a mismo input, mismo output). Esto cambiaría con Aprendizaje Automático, que tendría fitting sobre el caso actual.

Comando:

```
python busters.py -p BasicAgentAA -n 10
```

Output:

```
Average Score: 775.0
Scores:    775, 775, 775, 775, 775, 775, 775, 775, 775, 775
Win Rate:   10/10 (1.00)
Record:    Win, Win, Win, Win, Win, Win, Win, Win, Win
```

Ejecución con fantasmas aleatorios: los turnos varían ya que el movimiento de los fantasmas es aleatorio, no obstante aquí el aprendizaje automático cobraría valor real ya que tendría casos constantes de aprendizaje y podría estipular predicciones reales. Actualmente tan sólo se varía la posición que más se acerque al fantasmas para que no se quede atascado en ninguna posición, variando los ejes cartesianos. Lógicamente la puntuación es inferior respecto a la ejecución anterior ya que los movimientos son mayores e impredecibles. No obstante, debido a la optimización del algoritmo, se obtienen puntuaciones ligeramente inferiores.

```
python busters.py -p BasicAgentAA -n 10 -g RandomGhost
```

```
Average Score: 764.2
Scores:    764, 772, 767, 770, 769, 762, 774, 767, 727, 770
Win Rate:   10/10 (1.00)
Record:    Win, Win, Win, Win, Win, Win, Win, Win, Win
```

3.7. EL AGENTE PROGRAMADO EN EL EJERCICIO ANTERIOR NO UTILIZA NINGUNA TÉCNICA DE APRENDIZAJE AUTOMÁTICO. ¿QUÉ VENTAJAS CREEES QUE PUEDE TENER EL APRENDIZAJE AUTOMÁTICO PARA CONTROLAR A PAC-MAN?

El agente anterior se basa en estadística y posicionamiento dinámico basado en los datos de cada partida. Con aprendizaje automático tendremos la capacidad de obtener datos de muchas partidas y, de esta manera, tener la capacidad predictiva basada en la experiencia previa sobre un caso similar. Gracias a esto las acciones serán mucho más efectivas que simplemente perseguir el fantasma más cercano por la pantalla, ya que podremos predecir determinados movimientos basados en patrones gracias a los datos previamente obtenidos.

Del mismo modo, el aprendizaje automático puede generar nuevas estrategias dinámicamente para capturar los fantasmas de modo óptimo (si está en el suroeste: ¿debe moverse al sur o al oeste primero? Esta respuesta sería dada por el Aprendizaje automático).

4. CONCLUSIONES

La práctica ha resultado entretenida y ha sido de agrado realizarla. Pacman es un juego divertido y sobre el lenguaje Python será un gran entretenimiento realizar las prácticas.

Ha sido ligeramente complicado la implementación del algoritmo para la caza de fantasmas. Se ha pensado una y otra vez cada mejora implementada sobre él para optimizar el funcionamiento del mismo.

Gracias al conocimiento en algoritmos se denota que las funciones matemáticas siempre prevalecen en eficiencia frente a la computación, y gracias esto se alimentó la idea de intercambiar ejes cartesianos en caso de que el movimiento anterior fuese del mismo eje para prevenir que se quede pillado nuestro querido marciano y por ende que la salida sea predictiva pero variable y no fija.

El mapa es una A de Andrei (o de victoriA).

He tenido que hacer la práctica solo porque no conozco mucha gente de clase y mis compañeros cercanos ya tienen pareja. Soy un soltero de oro para esta asignatura.

Funciona en todos los mapas, pero es verdad que no es precisamente inteligente cuando tiene más paredes que terreno para recorrer. Necesita ayuda de la potente inteligencia del Aprendizaje Automático.