

Trabajo Práctico 3
Pensamiento Computacional
I100 Moreno

Olivia Browne Corbacho
ocorbacho@udesa.edu.ar

Maria Mora Vigo Malusardi
vigomalusardim@udesa.edu.ar

Objetivos del trabajo

El objetivo de este trabajo era crear un juego interactivo con el usuario del estilo del juego "Rogue". Consiste en que por la consola se imprima un mapa en el que se encuentra un personaje, y el usuario inserte una tecla (sea 'w', 's', 'd', 'a', ' ') que hace que ese personaje realice distintas acciones. A su vez, también en el mapa debía aparecer un segundo personaje, llamado Gnome, que sería el enemigo del personaje que mueve el usuario. El personaje debía moverse a través de los distintos niveles para alcanzar el tesoro, y luego salir de allí con el tesoro para volver al nivel inicial. El resto del juego era libre, es decir que quedaba a criterio nuestro que queríamos hacer con ello.

Diseño del programa

Comenzamos con archivos previamente editados y configurados, y en base a ellos comenzamos a completar las funciones que se encontraban incompletas. Desde un inicio decidimos que íbamos a trabajar separándonos pequeños problemas concretos para que cada uno resolviera por su cuenta. Cuando estos asuntos se resolvían, nos compartimos los archivos con las correcciones y continuábamos trabajando.

El trabajo consiste de distintos archivos que se importan entre sí: actions, game, gnome, human, items, keys, mapping y player. La mayoría de estos archivos contienen la definición de diferentes clases que luego son llamados e implementados en distintas funciones.

El archivo principal es game, este es el archivo que al ejecutarse imprime el mapa donde se realiza el juego. En este se agregan al mapa los ítems que después se ven en cada nivel del mapa, y se crean también los personajes 'player' y las distintas versiones de 'gnome'. A su vez es en este archivo donde se ejecuta el comando para que se imprima el mapa y reciba un input denominado key que hace que se mueva el player.

El movimiento del player se puede hacer sin la necesidad de apretar un enter después de cada input del usuario gracias al uso de la función read_single_key_press que se encuentra en el archivo 'keys'.

En el archivo mapping se encuentran tres clases principales a las cuales se les definen distintos atributos que luego son utilizados para crear el mapa del juego. La primera clase creada es 'Tile', que se usa para definir las distintas casillas del mapa. La segunda clase creada es 'Level'. Esta tiene definida los distintos atributos relacionados con las características de los niveles. En ella encontramos los atributos que permiten identificar si un casillero es accesible para el jugador, que se le agreguen diferentes ítems al mapa (como escaleras y armas), entre otras cosas. La tercera clase creada es 'Dungeon' y en ella se crean atributos muy similares a los de la clase 'Level' porque también son implementados en distintas funciones para obtener la misma información.

En el archivo player se crea la clase Player, que pasa a ser la clase padre de las clases Gnome y Human. Human y Gnome están creados en dos archivos separados, llamados human y gnome respectivamente. En ellos se definen los elementos básicos de los jugadores, como el icono que tienen, su ubicación y sus atributos.

El archivo actions es uno en el cual se encuentran distintas funciones definidas que representan distintos movimientos que puede realizar el player y son llamadas en base a las teclas que utiliza el usuario. Por ejemplo, si el usuario inserta como input la w,d,s,a, el jugador se mueve para arriba, para la derecha, para abajo y para la izquierda, correspondientemente.

De esta forma, consideramos que quedaba más claro y fácil para encontrar errores el uso de diferentes archivos en los cuales se creaban funciones o clases específicas.

Problemas y soluciones

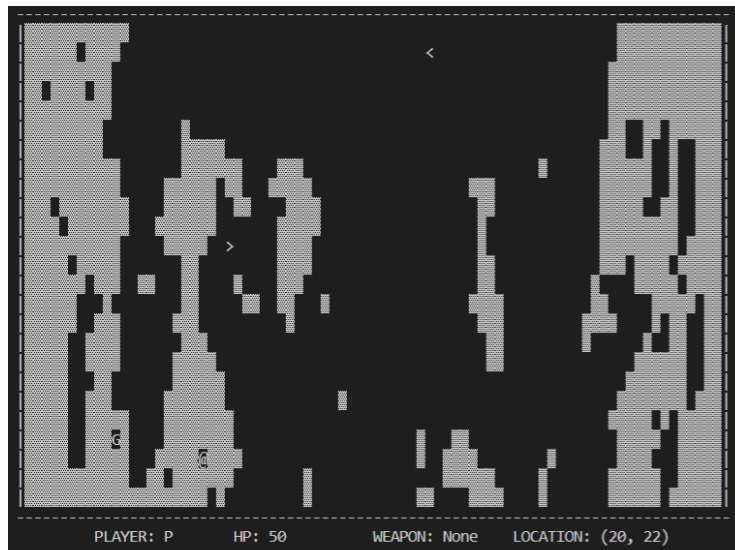
El mayor desafío de este trabajo fue poder escribir código en base a otro código que nos dieron. Al principio era medio confuso darnos cuenta cual era el propósito de cada función y cómo podíamos darle uso a aquellas que estaban definidas pero no implementadas. Pero, rápidamente decidimos que la mejor forma de poder entender el código era que cada uno lea los archivos y empiece a probar que cosas podíamos hacer con las funciones dadas. La primera prueba que hicimos fue crear al jugador y hacer que se imprimiera el mapa, y una vez que fuimos experimentando comenzamos a comprender cada vez más el código.

Un problema en particular que tuvimos era que en ciertas ocasiones el jugador se nos creaba encerrado entre cuatro celdas, y como en el jugador no tiene la capacidad de romper las paredes hasta que adquiere

el pico, quedaba atrapado sin poder moverse como se ve en la imagen. Esto lo solucionamos creando una función recursiva que encuentre si el jugador y el pico (o cualquier otro ítem que se represente en el mapa y que el jugador necesite obtener) se encuentre en una posición accesible. Es decir, que el pico y el jugador están conectados por celdas libres. Esta función se encuentra en el archivo mapping, bajo el nombre `are_connected`.

Otro problema que tuvimos fue relacionado con esta función que

implementamos. Llegaba un punto que por como estaba implementada la función, aparecía un `RecursionError`, porque se llamaba a esa función más veces de lo permitido por el `Recursion Depth Limit`. Esto sucedía porque esta función estaba implementada al momento de crear la ubicación de los ítems del mapa, que se les daba un lugar aleatorio libre en el mapa y luego se evaluaba si esa ubicación estaba conectada con la ubicación del jugador. Si no estaban conectados, se volvía a elegir una ubicación para el ítem y se evaluaba nuevamente si estaba conectado o no con el jugador. Para solucionar este error, averiguamos que se puede modificar el `Recursion Depth Limit`, entonces en el archivo mapping importamos la función `sys.setrecursionlimit()` y modificamos el límite para que en vez de que sea mil, sea diez mil. En la imagen adjunta se puede ver el error que aparecía.



```

PS C:\Users\Olivia\Desktop\Pensamiento_Computacional\TPS\TP3> & C:/Users/Olivia/AppData/Local/Microsoft/Windows/TPS/TP3/game.py
What is your name? oli
Traceback (most recent call last):
  File "c:\Users\Olivia\Desktop\Pensamiento_Computacional\TPS\TP3\game.py", line 45, in <module>
    dungeon.add_item(player, sword, 2)
  File "c:\Users\Olivia\Desktop\Pensamiento_Computacional\TPS\TP3\mapping.py", line 305, in add_item
    self.dungeon[level - 1].add_item(player, item, xy)
  File "c:\Users\Olivia\Desktop\Pensamiento_Computacional\TPS\TP3\mapping.py", line 116, in add_item
    if self.are_connected(player.loc(), position):
  File "c:\Users\Olivia\Desktop\Pensamiento_Computacional\TPS\TP3\mapping.py", line 235, in are_connected
    if self.are_connected(position, end):
  File "c:\Users\Olivia\Desktop\Pensamiento_Computacional\TPS\TP3\mapping.py", line 235, in are_connected
    if self.are_connected(position, end):
  File "c:\Users\Olivia\Desktop\Pensamiento_Computacional\TPS\TP3\mapping.py", line 235, in are_connected
    if self.are_connected(position, end):
[Previous line repeated 992 more times]
  File "c:\Users\Olivia\Desktop\Pensamiento_Computacional\TPS\TP3\mapping.py", line 225, in are_connected
    if initial == end:
RecursionError: maximum recursion depth exceeded in comparison
PS C:\Users\Olivia\Desktop\Pensamiento_Computacional\TPS\TP3>

```

Un tercer problema fue principalmente con el uso de GitHub. Decidimos que para poder enviarnos nuestros archivos con mayor prolijidad y para evitar errores, íbamos a utilizar GitHub. El problema que teníamos con esto era que habíamos decidido trabajar en distintas ramas de un mismo repositorio, pero notamos que estábamos teniendo dificultades en el merge de ambas ramas para darle lugar al main y que no estábamos pudiendo guardar correctamente nuestro trabajo. La solución de esto fue crear una nueva repo en la que trabajamos en conjunto en el mismo main. Notamos que para que podamos utilizarlo, una debía hacer el push, y luego la otra el pull antes de poder hacer otro push. Con esta metodología logramos enviarnos de forma eficiente los cambios que íbamos realizando en el archivo a medida que íbamos trabajando.

Indicaciones

Descargar todos los archivos en la carpeta comprimida.

Abrir todos los archivos

Hacer correr el archivo game.

Ir completando con los inputs solicitados hasta que el juego finalice.

Resultados

```

PS C:\Users\Olivia\Desktop\Pensamiento_Computacional\TPS\TP3\src\templates\oop\game.py

HELLO
WELCOME TO THE DUNGEON!
What is your name? Oli
Select an avatar!
  1. 🐱
  2. 🐼
  3. 🐶
  4. 🐹
  5. 🐭
  >> 6
Please insert a valid input.
Select an avatar!
  1. 🐱
  2. 🐼
  3. 🐶
  4. 🐹
  5. 🐭
  >> 5

```

A continuación se verán imágenes de la ejecución del código.

Al hacer correr el archivo de game.py se ve por la consola el siguiente mensaje. Se le solicita al usuario que ingrese su nombre y luego se le pide que seleccione un avatar que va a representar el carácter para su personaje.

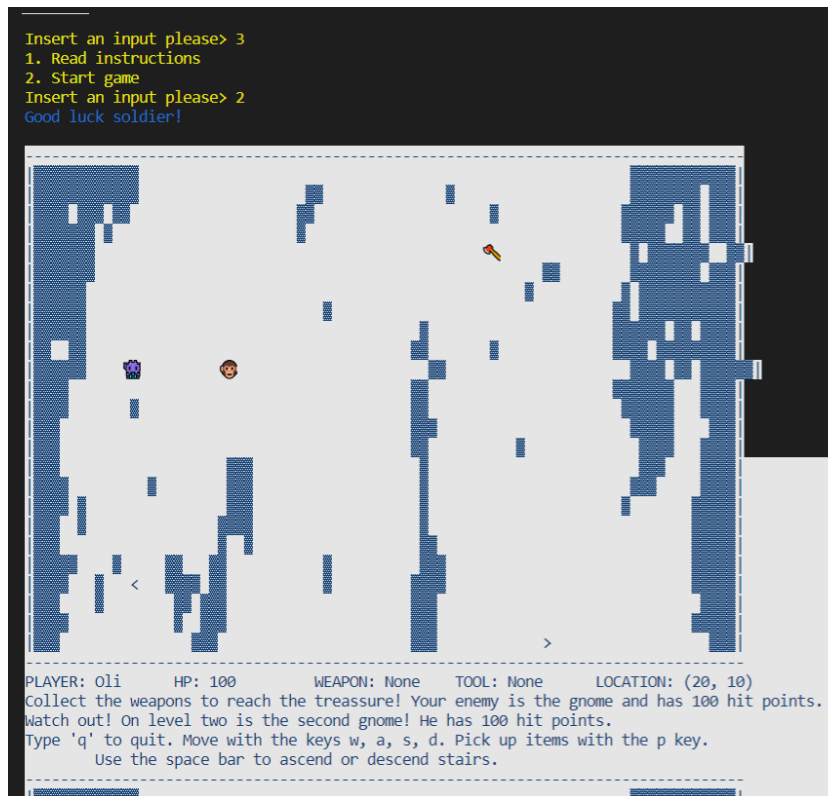
Si se ingresa un valor invalido, se imprime un mensaje que aclara que el input es erróneo y se vuelve a solicitar un input para seleccionar el avatar.

```

1. Read instructions
2. Start game
Insert an input please> 1
THE GAME IS SIMPLE.
The goal is to move your character through the map in order to reach the third level where you will have
to collect the treasure. Use the keys w, a, s, d to move.
You can also use the p to pick up items and the space bar to ascend or descend stairs. Make sure you collect the pick before
you descend the stairs. If you don't, you won't be able to go to the next level.
BE CAREFUL! There are gnomes you will have to fight in each level. Collect the pickaxe before you fight them.
Win the game by getting back to level 0 with the treasure and by having collected any and every other object found ('(', '/', 'x', '').
GOOD LUCK!!
1. Read instructions
2. Start game
Insert an input please> 3
1. Read instructions
2. Start game
Insert an input please>

```

Al seleccionar el avatar, se pide otro input para elegir leer las instrucciones o jugar al juego. Si se elige la opción 1 se imprimen por la consola las instrucciones. Si se ingresa un input erróneo se vuelve a hacer la pregunta. Al seleccionar la opción 2, se da inicio al juego.

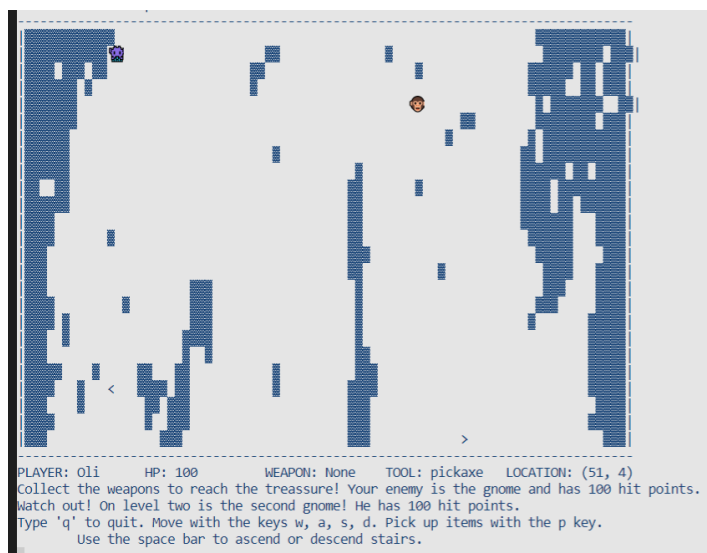


Al iniciar el juego, la consola se ve como la imagen a la izquierda.

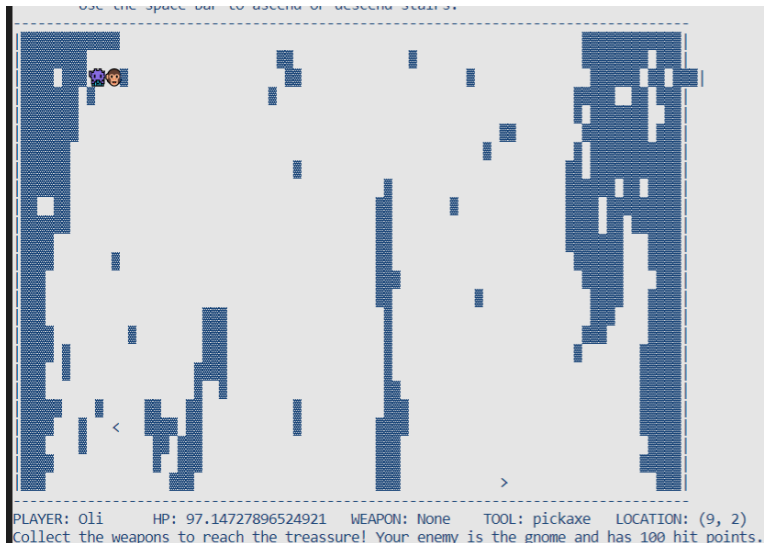
Se imprime el mapa con los ítems y jugadores creados en color azul, y los ítems como emojis.

Bajo el mapa se ve la información general del juego y pequeños recordatorios sobre cómo jugar.

A medida que se ingresen distintas teclas, el jugador se ira moviendo en distintas posiciones, con el objetivo de llegar al pico.



Al agarrar el pico, la sección TOOL se actualiza y muestran que el pico fue agarrado. Esto permite que el jugador ahora atraviese las paredes.

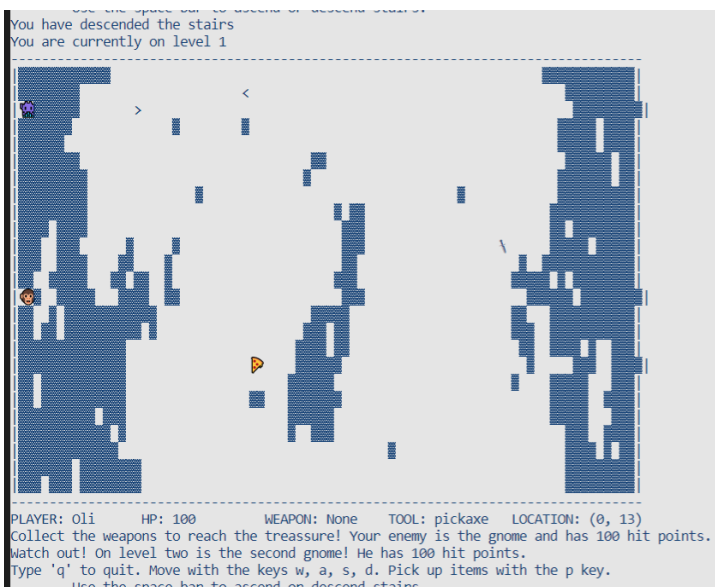


Si el jugador en esta instancia se acerca al gnomo, es probable de que pierda parte de sus vidas, dado a que no tiene un arma para pelear contra su enemigo.

El jugador debe tener cuidado de no chocarse con el gnomo demasiado porque corre el riesgo de perder su vida.

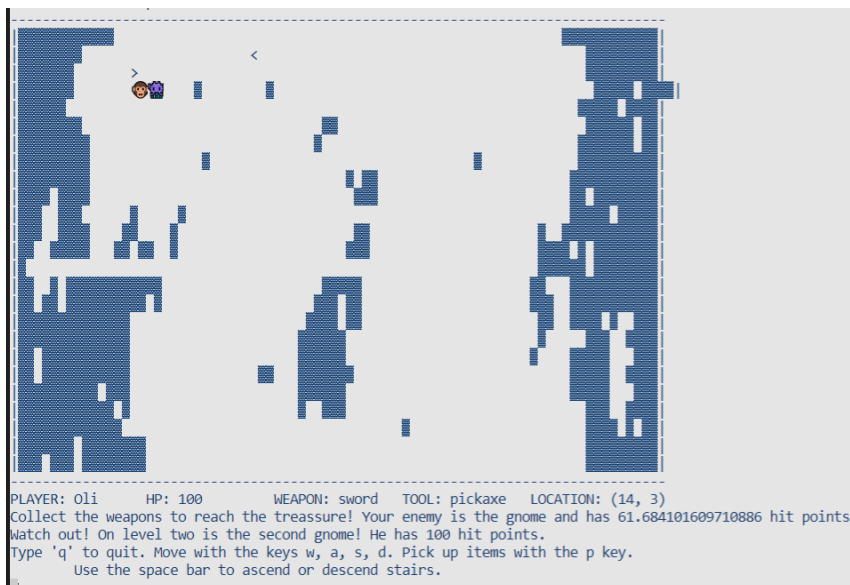


En este caso podemos ver que el jugador perdió el juego porque perdió todas sus vidas, ya que se enfrentó con el gnomo muchas veces y no tenía ningún arma para defenderse.

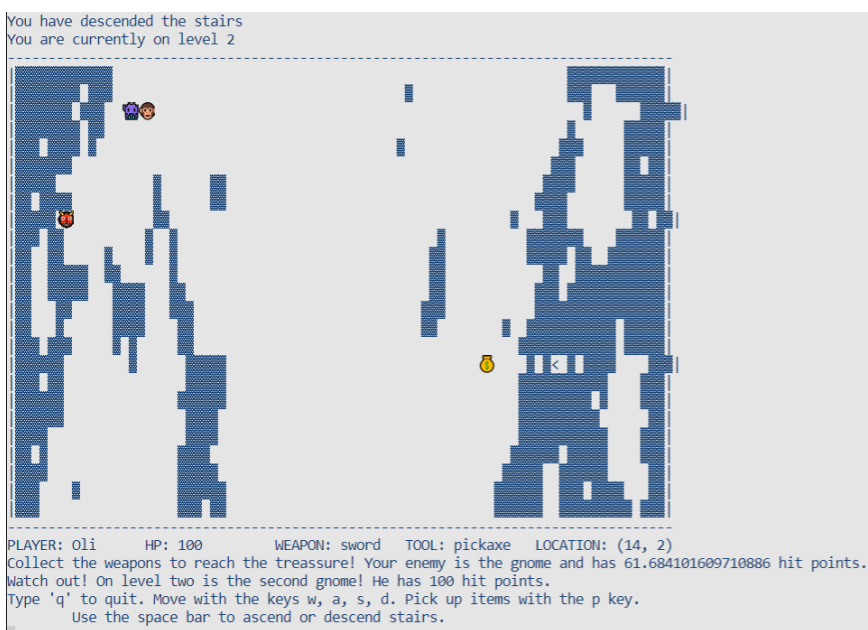


Si esto no sucede, y el jugador sigue el juego, debería buscar avanzar al siguiente nivel usando las escaleras. Una vez que se posiciona sobre el icono de las escaleras y selecciona el espacio, se regenera un mapa nuevo del siguiente nivel.

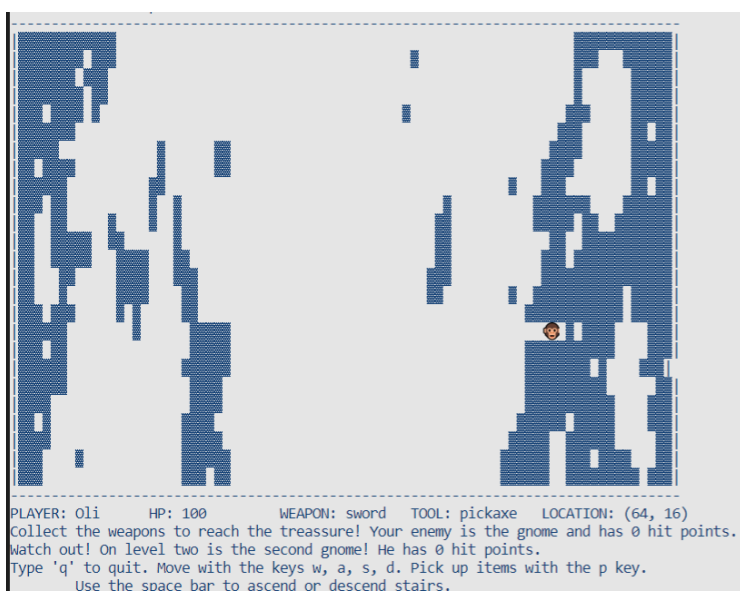
Aquí, el jugador debe recoger la espada y la pizza con el desafío de escapar del gnomo que está programado para seguirlo. Al juntar la pizza, el jugador recarga 10 hit points, si es que sus hitpoints son menores a 100.



Una vez que se recolectaron los elementos, si el jugador se acerca lo suficiente al gnomo, logra bajarle los hit points ya que tiene un arma para producirle daño. Se ve cuanta vida le queda al gnomo en la consola bajo el mapa. Si el gnomo no es eliminado en este nivel, en el próximo nivel va a seguir apareciendo.



Una vez que se llegó al segundo nivel, el jugador debe enfrentar al segundo gnomo y a su vez recolectar el tesoro para poder regresar al nivel inicial y ganar la partida.



Habiendo recolectado todos los ítems, y posiblemente matado a los dos gnomos, el jugador debería subir las escaleras para volver al inicio.

```
You have climbed the stairs
You are currently on level 1
0

[Map view showing a cave with a treasure chest icon on the right side.]

PLAYER: Oli      HP: 100      WEAPON: sword  TOOL: pickaxe  LOCATION: (65, 16)
Collect the weapons to reach the treasure! Your enemy is the gnome and has 0 hit points.
Watch out! On level two is the second gnome! He has 0 hit points.
Type 'q' to quit. Move with the keys w, a, s, d. Pick up items with the p key.
Use the space bar to ascend or descend stairs.
```

Hace esto para cada nivel, atravesando el plano hasta llegar a las escaleras que suben en cada nivel.

```
[Map view showing the player at the stairs.]

PLAYER: Oli      HP: 100      WEAPON: sword  TOOL: pickaxe  LOCATION: (29, 1)
Collect the weapons to reach the treasure! Your enemy is the gnome and has 0 hit points.
Watch out! On level two is the second gnome! He has 0 hit points.
Type 'q' to quit. Move with the keys w, a, s, d. Pick up items with the p key.
Use the space bar to ascend or descend stairs.
You have climbed the stairs
You are currently on level 0

You have picked up the treasure and made it back to the start!
CONGRATULATIONS, YOU WON!
You have completed the game in 406 moves!

WINNER

PS C:\Users\Olivia\Desktop\Pensamiento_Computacional\TPS\TP3>
```

Cuando se llega al nivel inicial, habiendo recolectado todos los ítems, se gana el juego.

Se imprime por consola un mensaje de ganador, y se finaliza el juego.

```
[Map view showing the player at the stairs.]

PLAYER: o      HP: 100      WEAPON: None  TOOL: None  LOCATION: (21, 11)
Collect the weapons to reach the treasure! Your enemy is the gnome and has 100 hit points.
Watch out! On level two is the second gnome! He has 100 hit points.
Type 'q' to quit. Move with the keys w, a, s, d. Pick up items with the p key.
Use the space bar to ascend or descend stairs.

Thanks for playing! See you next time :)
PS C:\Users\Olivia\Desktop\Pensamiento_Computacional\TPS\TP3>
```

En cualquier momento del juego si se ingresa la letra 'q' como input, se da por finalizado el juego porque el usuario decidió renunciar.

Bibliografía (citado en APA)

W3 Resources, . "Python: Recursion Limit." *w3resource*, 2022, <https://www.w3resource.com/python-exercises/python-basic-exercise-80.php>.

RecursosPython, . (2022, April 1). *Colorama - texto y Fondo Coloreados En La Consola*. Recursos Python. Retrieved June 3, 2022, from <https://recursospython.com/guias-y-manuales/colorama-texto-fondo-coloreados-la-consola/>