

Ejercicio Integrador Zelda

Se pide ayudar a terminar de programar el Zelda hecho en Gobstones, hay que poder programar para resolver cómo combatir a los enemigos.

Se trata de un personaje llamado Link que tiene que combatir enemigos en diferentes mapas. Link cuenta con 2 armas de ataque que son el Arco y la Espada, y 2 de defensa que son el Escudo y la Persuasión. El Arco y el Escudo pueden ser usadas para atacar y defenderse respectivamente de enemigos que se encuentran hacia una dirección a cualquier distancia. La espada y la Persuasión solo pueden usarse para atacar y defender cuando el enemigo se encuentra lindante en cualquiera de las direcciones.

Las Enemistades de Link pueden ser de diferentes tamaños y colores. Estas características hacen que sus energías de ataque y defensa sean diferentes. Las más grandes son más débiles que las más pequeñas y los de color más oscuro tienen menos poder de ataque que las de color más claro.

También se debe tener en cuenta que los mapas contienen diferentes objetos. Las Rocas y los Árboles que interfieren en la visión entra Link y sus enemigos, y los Arbustos y Yuyos que no interfieren para que estos puedan verse y eventualmente atacarse.

Primitivas:

MoverLinkHacia_(dir)

PROPÓSITO: Mover una ubicación a Link en la dirección dada
PRECONDICIÓN: Debe estar el cabezal sobre Link
Debe existir al menos una ubicación en la dirección

hayObstáculo()

PROPÓSITO: Indica si hay un obstáculo en la celda actual.
PRECONDICIÓN: Ninguna
TIPO: Booleano

hayEnemigoDeLink()

PROPÓSITO: Indica si hay un enemigo en la celda actual.
PRECONDICIÓN: Ninguna
TIPO: Booleano

MatarEnemigoHacia_ConElArco(dir)

PROPÓSITO: Elimina al enemigo que se encuentra hacia la dirección dada
PRECONDICIÓN: Debe haber un enemigo a una distancia mayor a 3 ubicaciones en la dirección dada.
Debe estar el cabezal sobre Link

MatarEnemigoHacia_ConLaEspada(dir)

PROPÓSITO: Elimina al enemigo que se encuentra hacia la dirección dada
PRECONDICIÓN: Debe haber un enemigo a en la celda lindante en la dirección dada.
Debe estar el cabezal sobre Link

Ejercicio 1)

Se pide implementar la función **hayEnemigoEnLaVisionDeLinkHacia_(dirección)** que teniendo a link en la posición actual asumiendo el cabezal sobre el mismo, y dada una dirección indique si puede ver a su enemigo y no hay ningún obstáculo en el camino.

Ejercicio 2)

Se pide implementar la función **distanciaEntreLinkYSuEnemigo()** describe la distancia que hay entre link y su enemigo sabiendo que existe algún enemigo visible sin obstáculos en alguna dirección.

Ejercicio 3)

Implementar el procedimiento **EliminarEnemigoVisible()** donde Link mata a un enemigo que se encuentra en alguna ubicación visible para LINK. Para seleccionar de qué forma matar el enemigo debe evaluar si lo mata cuerpo a cuerpo o de un flechazo. Si el enemigo está a más de 3 ubicaciones de distancia va a usar su arco, sino se acerca al enemigo y lo mata con una espada.