Conceptualizando el Maze

1. En la página 2 del tutorial señalan que los sprites tienen las siguientes propiedades: width = 32 y height = 32. ¿Qué son esos 32?

Los 32 son el número de pixeles que tiene la imagen, en este caso tanto de altura como de ancho.

2. ¿De qué tamaño es la máscara del osito?

La máscara del oso automáticamente se pone en 28px x 29px. Sin embargo, se puede modificar para que sea exactamente de la forma del oso.

3. ¿Qué es la máscara de colisión y para qué la utiliza gamemaker (el motor)

La máscara de colisión es una figura que recubre al sprite la cual utiliza el gamemaker para calcular donde se colisionan dos objetos, o donde no se colisionan con respecto a la imagen.

4. ¿Qué es el origen de un sprite? ¿Dónde está ubicado?

El origen es el lugar desde donde el sprite se dibuja cuando se coloca como objeto en una habitación. Usualmente está ubicado en la esquina superior izquierda, más se modifica al centro para varios casos.

5. Al crear el objeto wall, la propiedad solid se hace true. ¿Qué significa esto?

Eso significa que el objeto es sólido, es decir, ningún otro objeto debería de ser capaz de atravesarlo.

6. ¿Por qué no es necesario adicionar eventos al objeto wall?

Porque de por sí el wall no realiza ninguna acción, y solo es un objeto contra el cual otros objetos no deben de pasar.

7. ¿Qué pasa si no se adicionan las acciones del objeto obj_goal?

Nada ocurre. Cuando el jugador lo toca nada pasa y el juego es incapaz de terminar.

8. En el osito u objeto person qué quiere decir la speed, cuáles son las unidades.

La speed es la velocidad a la que se empieza a mover el objeto en una dirección. Está en “steps” (pasos. Usualmente gamemaker da 30 pasos por segundo, pero eso depende del room)

9. ¿Cuáles son las dimensiones de cada una de las celdas que componen la grid en el laberinto?

Cada celda es de 32x32, para adecuarse al tamaño de los sprites.

10. ¿Por qué es necesario mantener alineado el objeto person con la grid del laberinto (maze)?

Porque de otra forma el jugador tendría problemas para jugar el juego, le sería difícil moverse y entrar por los cruces.

11. Personalizar: haga que los movimientos del objeto person no estén alineados en la grid del laberinto. ¿Cuál es el comportamiento?

El objeto person es capaz de moverse más libremente por el mapa. Pero esto es un problema. Se choca contra los muros en todo momento y es difícil voltear por esta misma razón. Además, se puede mover verticalmente incluso cuando no hay espacio para hacerlo.

12. ¿Por qué ocurre el comportamiento visto en el punto anterior?

Porque la máscara del objeto no es exactamente 32x32. Y porque ya no está restringido de moverse de un cuadro a otro, sino que solo va a una velocidad determinada cuando se presiona la tecla de dirección. Como no existe esta restricción, es fácil golpearse contra un muro sin querer.

13. El objeto obj_door tiene un evento denominado Step. ¿En qué consiste este evento? ¿Cuándo ocurre?

El evento consiste en revisar unos parámetros (en este caso la cantidad de diamantes en el mapa para abrir la puerta) cada vez que sucede un paso en el juego. Ocurre desde que se crea el objeto y se genera el primer Step, repitiéndose cada vez que pasa un Step.

14. El objeto obj_door ejecuta la acción Destroy the instance. ¿En qué consiste esa acción?

Consiste en destruir la instancia a la cual se le aplica dicha acción, eliminado por completo a ese objeto del juego.

15. Personalizar: luego de la acción Destroy the instance, adicione otras acciones. ¿Qué ocurre? Explique

Solo las acciones que se realizan por si solas ocurren. Todas las demás que sean relacionadas con el objeto no lo hacen. Es decir, acciones como sonidos o reiniciar el juego ocurren después de que se destruye, pero mover el objeto, cambiarle el sprite o cosas por el estilo se cancelan, pues el objeto como tal ya no existe.

16. ¿Qué quiere decir que un objeto sea el Parent de otro objeto?

Eso quiere decir que las acciones y eventos que se apliquen al objeto Parent, se aplicarán a todos sus objetos child (los que estén asociados con el primer objeto) sin requerir copiar el código una vez más.

17. ¿Cómo se puede utilizar la idea del punto anterior si quiero hacer un juego con 6 monsters que se comporten igual, todos matan al personaje, pero que cada monster se vean diferente?

Se puede crear el primer monster, y añadirle todas las acciones que se requieran. Después, se le pone sprites diferentes a los otros 5 monsters y se les añade que su parent sea el primer monster, sin necesidad de poner más código.

18. ¿Qué quiere decir que los eventos del hijo pueden sobreescribir (override) los eventos del padre? Indique un ejemplo de lo anterior.

Cuando un child tiene al menos un evento con acciones en él, estas acciones se van a realizar primero, o en vez de las del padre en el mismo evento. Por ejemplo, si el parent dice que los monsters se mueven de derecha a izquierda, pero uno de sus childs dice que se mueve de arriba hacia abajo, el monster que tenga estas acciones se moverá de manera diferente a todos los demás monsters (de arriba hacia abajo).

19. ¿Qué es un objeto controlador. ¿Para qué se utilizó este tipo de objeto en el juego?

Un objeto controlador, o controller, es un objeto que no aparece como tal en el juego, más guía las acciones de otros objetos, el cambio de ciertas variables o el comportamiento del juego. En este juego se utilizó para dibujar la pantalla inicial, y para que cuando el jugador oprimiera una tecla, pudiera seguir hasta la próxima pantalla.

20. **Personalizar**: adicione un objeto cualquiera al laberinto que tenga un sprite (cualquiera). Ejecute el juego. ¿Se observa el sprite? Ahora adicione el evento Draw a ese objeto. Ejecute de nuevo el juego. ¿Qué ocurre? Explique

Es posible ver el sprite si se adiciona el objeto por sí solo, mas nada se puede hacer con él. Si se le adiciona el evento Draw, nada cambia, pues este evento no posee ninguna acción y se elimina automáticamente. Si se le adiciona una acción al evento, el sprite se cambia por lo que sea que pusimos a dibujar, sea otro sprite, o una acción.

21. ¿Qué es la propiedad depth de un objeto?

Es la profundidad, según en qué momento lo dibuje el gamemaker (antes o después de otros objetos).

22. Personalizar (room4): ¿Qué pasa cuando un monster pasa a través de una bomba? Cambie el depth de la bomba de 10 a -10. ¿Qué pasa ahora cuando el monster pasa a través de la bomba?

El monster pasa por encima de la bomba con depth 10. Con depth -10, el monster pasa por debajo de la bomba.

23. El objeto obj_trigger tiene una acción llamada Change instance into obj_explosion. Al abrir dicha acción aparece Applies to. ¿Qué significa esto?

Significa “Aplica a”, es decir, a qué objeto es al que se le va a aplicar dicha acción (que puede ser el mismo objeto que la hace, el otro objeto que toca al primero, o un objeto específico del room.

24. En el punto anterior ¿Qué significa perform events?

Esto quiere decir que cuando una instancia se cambiar por otra, esta puede realizar las acciones que tiene inscritas en su objeto, o solo hacer el cambio, según se indique.

25. Explique qué significa la opción Relative en las acciones de obj_explosion

La opción relative indica que las acciones se realizan según el punto de origen del obj_explosion y no en un punto específico del room.

26. En la sección de Blocks and holes del tutorial, explique porqué es necesario probar la posición relativa (a quién) 8*other.hspeed, 8*other.vspeed. ¿Qué quiere decir other, hspeed, vspeed?

Es necesario probar la posición relativa de esos puntos pues son la posición a la que la roca va a moverse cuando el jugador colisione con ella. Si se encuentra con una pared, debe de dejar de moverse. Si se encuentra con el hueco, debe de taparlo. De lo contrario, cuando el jugador toque la roca esta no va a hacer nada, o se va a mover a una velocidad menor que el jugador, o va a atravesar las paredes, o no va a tapar el hueco, entre otras posibilidades.

Other quiere decir que se revisa la variable siguiente del objeto con el cual está colisionando.
hspeed es la velocidad horizontal que lleva el objeto (en este caso el obj_person)
vspeed es la velocidad vertical que lleva el objeto (en este caso el obj_person)