Santiago Calle Rodas - 000290117

Conceptualizando primer juego.

1. Un sprite es una representación visual del objeto creado en el juego, es solo una imagen visual, pero cuando se crean varias y se superponen pueden generar una animación (Movimiento). En cambio el objeto es una entidad que tiene un comportamiento (evento) donde se le puede poner un sprite como representación grafica del objeto, estas entidades son la mayoría del juego, personajes, muros, paredes bolas, pero para que se vean como tales se les representa con un sprite.
2. Si, el objeto es una entidad que puede ser representada por múltiples sprites, ejemplo, un objeto que representa un personaje, el personaje lo vamos a poner a caminar, entonces hay que hacer varios sprite para que el personaje simule el movimiento de caminar, el personaje también muere, otro sprite con el personaje muerto, si ponemos a saltar el personaje hay que hacer otro sprite con el personaje saltando, todos estos sprites están en el objeto pero mediante los eventos van cambiando.
3. El asset es los archivos que tiene el juego que no son sprite, es decir, los assets pueden ser las imágenes de fondo, los sonidos del juego, efectos, niveles etc. Es cualquier dato que se pueda ingresar en el motor de juego en este caso game maker.
4. Los eventos son esos momentos del juego que pasan de acuerdo a cómo los hayas programado en los objetos. Estos comienzan desde el primer momento del juego hasta el último, y el programador puede elegir también en qué momento quiere que ocurran.
5. Create: Este evento es la primera cosa que ocurre dentro de una instancia que se da lugar en un “room”, y puede ser usado para cosas que generalmente pasan solo una vez o cuando una instancia aparece primero en el juego.

Collision: Es muy importante saber cuándo dos o más instancias de un objeto tienen que colisionar, y es un evento el cual se suelta en un objeto y se debe especificar con cuál otro objeto va a colisionar.

Mouse: El Mouse event está determinado por una serie de sub-eventos los cuales dan más precisión y facilitan a la hora de estar creando el juego, ya sea manteniendo presionado algún botón o haciendo un clic…

1. Las acciones pueden determinarse por comandos y funciones que facilitan la manipulación del juego, y vienen en conjuntos los cuales se pueden arrastrar y soltar en un evento del objeto para crear algún tipo de comportamiento en este. A esto se le llama comúnmente DnD (drag’n’drop).
2. Se podría decir que los eventos son como algo más general que las acciones, ya que las acciones se pueden arrastrar hasta los eventos para generar algunos comportamientos en un objeto. Un ejemplo de ello podría ser que el evento sea presionar la tecla izquierda del mouse, y la acción, sería que el objeto se comenzara a mover en cualquier dirección.
3. Cuando se tiene un objeto, con eventos definidos, ya se puede colocar en el “room”, y al ponerlo allí, se crea como una copia de este objeto, a lo cual se le llama instancia; y se puede poner más de una allí, y editar la posición, o el color de alguna si se quiere.
4. La diferencia es que un objeto es creado con anterioridad, al cual se le añaden los eventos; y por consiguiente, al crear una instancia, estamos obviamente utilizando un objeto, pero la instancia es la que va puesta ya en el “room”; y se pueden crear varias instancias las cuales se pueden editar por separado pero no los eventos, que fue lo que se hizo con anterioridad al crear el objeto, sino posición, fondo, entre otros.
5. …
6. Son las acciones que se realizan por segundo, el número de pasos por segundo se configura en cada room del juego, las configuraciones más comunes son 30 o 60 pasos por segundo. Si queremos que unos aviones salgan en determinado tiempo tenemos que crear un evento steps para que salgan en el tiempo propuesto.

Conceptualizando Plataforma.

1. No ocurre nada, la bola no se puede mover.
2. Si se cambia a una forma rectangular y se pone en esta posición, la figura puede moverse normal.
3. Que si un objeto está posicionado en una instancia, la gravedad se hará cero; y de ocurrir lo contario, el objeto bajará según la gravedad puesta.
4. Porque al poner esto, se le está dando un límite a la velocidad vertical, y no seguirá incrementando excesivamente.
5. Porque con esa acción se le está diciendo que al tocar objetos sólidos, la velocidad se haga cero en el momento de la colisión.
6. Al objeto saltar en un bloque, se queda quieto ya que se anula la velocidad, y hay que ponerle cero para que al colisionar con este objeto se pueda mover.
7. Se le tiene que asignar una tecla y se le debe decir que al momento de estar tocando alguna plataforma, se haga efectiva la velocidad negativa que se le va a asignar a la acción “speed vertical”.
8. Para que el objeto vuelva a bajar, se le está diciendo que cuando esté en el aire y no esté colisionando con ningún objeto sólido, se active la gravedad puesta anteriormente.
9. - El objeto deja de moverse porque se le está diciendo que solo se mueva si no está colisionando con algo en y=1, refiriéndose a un objeto sólido que esté bajo de la bola.

* En la acción “move to contact in direction direction”, se está diciendo que el objeto se pueda mover en la dirección que se le especifique cuando esté en una plataforma.

1. Esto significa que puede ocurrir que cuando la imagen tiene un sprite animado, la máscara de colisión puede cambiar en cada paso que se da, así que se debe asegurar que el personaje tenga solamente una máscara de colisión para que no ocurra esto.
2. Es la misma que la 9.
3. Significa que el límite de las “cajas” de los sprites se tienen que poner igual en todos para evitar que el personaje se mueva extraño, o que al momento de colisionar con algo, ocurra algún error.
4. Se debe poner objetos encima de los tiles y poner que no sean visibles, y a estos son los que se les asignan los eventos.