1. variable es algo que utilizamos para almacenar un valor para su uso en una o más operaciones o funciones construidas en las variables:
2. son variables especiales que se "construyen en" los objetos y las habitaciones en el mundo del juego y que puede ser única instancia o de alcance mundial (pero no local)
3. instancia: La categoría más común. Las variables son definidas dentro de la instancia. Son únicas y propias de su instancia y se les puede usar en cualquier evento y cualquier función dentro de esa instancia.

local: Estas variables se declaran mediante la función "var". Una variable local sólo es válida dentro del evento o script en el cual se crea.

global: Una variable global pertenece a todo el entorno de juego, y no a una instancia en específico (a pesar que se declare dentro de una instancia). En un principio, tiene que declararse como global, pero después de esto, cualquier instancia puede leer o cambiar su valor. El valor de la variable siempre reflejará la última operación de la que fue objeto, sin importar qué instancia desarrolló la operación.

1. Variables internas: Son variables especiales inherentes a los objetos y habitaciones en el entorno. Están ahí desde que una habitación u objeto es creado. Pueden tener alcance de instancia o global, pero nunca local. Hay una gran cantidad de estas variables y cada una tiene usos muy específicos, se les describe con detalle a lo largo del manual en las secciones correspondientes.
2. Variables de Instancia: El alcance de una variable de instancia está limitado a la instancia donde se crea. Una variable de instancia se crea dentro dela instancia de un objeto y es considerada única para esa instancia

Variables locales: Una variable local se declara, se usa y después se descarta. Las variables locales solamente se crean para eventos específicos, cuando dicho evento finalice, la variable se descarta

Variables globales: En términos simples, una variable global es un tipo de variable que una vez declarada no pertenece a ninguna instancia, pero puede ser leída por todas ellas. Las variables globales deben ser declaradas, tal como sucede con las variables locales, pero a diferencia de éstas, una variable global permanece en memoria hasta el final del juego

1. La variable instancia ya que permite usarla en cualquier evento y cualquier función dentro de esa instancia.

La variable local ya que sólo es válida dentro del evento o script en el cual se crea.

La variable global ya que esta, en si no pertenece a ninguna instancia

1. Vidas, contador de balas, bonus, puntajes y salud
2. Flechas, Durabilidad, Fuerza, Salud, Pociones, Mana, Cambio de arma, accesorios
3. No, ya que game maker reconoce a las mayúsculas como un carácter diferente de las minúsculas
4. En y = -16
5. Que dispara dependiendo de la posición del avión
6. Porque esto las coloca por debajo de otros sprites u objetos generando la sensación de profundidad
7. Selecciona una posición aleatoria en la horizontal
8. Porque esta función empieza a aplicarse una vez que las islas desaparezcan de la pantalla la primera vez
9. Esta función establece la semilla a un valor aleatorio. Si usted necesita para mantener un valor constante durante un número de carreras de un juego, usted debe utilizar random\_set\_seed (). Tenga en cuenta que cuando se utilizan las funciones de números aleatorios en GameMaker: Estudio de la semilla inicial es siempre el mismo, ya que esto hace que los errores de rastreo y depuración mucho más fácil. Si desea probar con el verdadero azar, usted debe llamar a esta función en el inicio de su juego.
10. Porque esto facilita el rastreo de errores
11. Después de iniciar el juego
12. El anterior código cuestiona si hay un obstáculo en el camino del objeto en la posición x=40 y luego si no halla ningún obstáculo desplaza al objeto a la posición x=-4
13. Esta no se coloca en el room ya que es generada por el objeto avión relativamente en la posición en la que este se encuentre
14. Esta variable evalúa si ya la bala supero la vista del room para poder destruirla y así ahorrar memoria
15. Esto depende del evento step que se ejecuta cada 30mpasos o segundo
16. Disminuyendo la cantidad de pasos
17. Primeramente al chocar se activa el evento de colisión que cambia el sprite del avión por la animación de la explosión y luego de que esta termina la destruye y reinicia el juego
18. Este objeto invisible controla la cantidad de enemigos que surgen y su posición
19. Funciona como un temporizador que le indica a game maker realizar una acción cada cierta cantidad de tiempo
20. Prueba Probabilidad Esta acción tiene el resultado de una oportunidad "uno en X", donde " X "es el número que ingresa a prueba.
21. Se crea con una probabilidad de 1/30 es decir de cada 30 el chance es de 1
22. Esta variable chequea la instancia y luego hace que el objeto bala persiga al objeto avión con cierta velocidad vertical
23. Lo primero es un set variable que confirma si el objeto avión puede disparar en caso afirmativo inicia un bloque en el cual le permite crear 3 objetos bala dependiendo de su score, cuando terina ese bloque le permite crear al objeto avión un último objeto bala y luego vuelve a evaluar la capacidad del objeto avión para disparar e inicia una e inicia una alarma de tiempo para que pueda volver a disparar el objeto avión de nuevo
24. Ellos son, básicamente, pequeñas ventanas en su mundo de juego que le permiten mostrar las partes jugador de una habitación, ya sea reducido o 1: 1, y como tales son esenciales cuando su sala de juegos es más grande que el tamaño de la pantalla.
25. La vista se crea cundo uno quiere centrarse en un espacio específico del room o este es más grande que la pantalla. Para que se quede estático y sea la vista la que se mueva a la velocidad que una mismo le asigne
26. Que su movimiento es instantáneo en el momento en el que se le da la acción de moverse
27. La vista se puede hacer mover asignándole que persiga a un objeto generalmente este es el personaje del juego o con el cual se está jugando. La velocidad de la room puede ser ajustada al gusto pero al darle un valor negativo esta se movería de forma instantánea con el objeto asignado a seguir
28. Con una acción de set variable que confirme en todo momento la posición del avión y cuando este esté cerca del borde de la room, lo envíe con una acción de salto hacia atrás
29. Con una set variable que confirme la posición en todo momento del objeto bala y cuando este esté fuera de la room, inicie la acción de destrucción
30. Con una velocidad vertical prediseñada
31. La vista siempre se define como un área rectangular en la sala, donde se especifica la posición de la esquina superior izquierda, el ancho y la altura de esta zona. Luego debe especificar donde se muestra esta área en la ventana en la pantalla mediante la definición del puerto de vista, donde de nuevo se especifica la posición de la esquina superior izquierda y el tamaño.
32. Si entre más “Ticks” la bala se tardara más en salir debemos de dividir el número actual por 3: (15/3 = 5), al cambiar el número de ticks, 3 balas saldrán por segundo
33. Simplemente no se dispararía nada, pues no hay ningún valor que pueda tomar el “obj\_bullet2”