Cuestionario de 1945

* Las variables son propiedades de un objeto o una instancia. Son algo que utilizamos para guardar un valor que se puede usar en uno o más operaciones o funciones
* El scope podría decirse que es el área de operación o alcance de una variable
* Variables de Instancias, Variables Locales y Variables Globales
* El scope en las variables de instancias significa que es único para esa instancia y que puede ser usado en cualquier evento o función dentro de la misma instancia. En las locales es solo valido para determinado evento así que game maker la usa y cuando finaliza el evento es olvidada. Las variables globales aplican para todo en el juego como tal.
* Variable de instancia
* Variable local
* variable global
* 5 variables globales en game maker: score, Health, lives, mouse\_x, mouse\_y, room\_speed.
* 8 variables de instancia en game maker: x, Y, Hspeed, vspeed, direction, speed, visible, solid.
* No son iguales puesto que las variables son sensibles a las mayúsculas, es decir, ponerlo en mayúscula significa que pueda ser de pronto una variable diferente.
* Se crea justo debajo del objeto my\_plane, relative indica que bullet siempre va a ser creado bajo el avión, así este cambie de posición.
* La islas tienen un dept de 10000 porque así serán dibujadas primero y no habrá riesgo de que estas se crucen con otros objetos, entonces cualquier nuevo objeto se hace con menos profundidad y así su dibujo se verá superpuesto al de las islas.
* función [random](http://docs.yoyogames.com/source/dadiospice/002_reference/maths/real%20valued%20functions/random.html) (room\_width): es utilizada para ubicar el objeto en una posición cualquiera, desde 0 hasta (en este caso) el ancho del cuarto.
* Porque la función random escogerá automáticamente siempre el mismo valor, cada vez que el juego sea iniciado otra vez
* la función [randomize](http://docs.yoyogames.com/source/dadiospice/002_reference/maths/real%20valued%20functions/randomize.html), hace que cada vez que el juego que inicie el valor que escoja sea verdaderamente al azar.
* No debería utilizarse ya que es más fácil rastrear fallos en el juego si siempre se empieza de la misma manera.
* Se puede utilizar una vez esté terminado el juego y que no sea necesario depurarlo más.
* El código indica que al presionar la tecla izquierda, activa probar variable: si es mayor de 40 entonces se desplaza -4 pixeles a la izquierda. Probar variable se pone para que el avión solo se pueda mover cuando no sobrepase el límite izquierdo del cuarto.
* La posición seria relativa al avión, y se coloca en -16 en y para que los ejes x coincidan y la bala no se vea cuando está bajo el avión.
* Se utiliza cuando la bala sobrepase el borde superior esta se autodestruya
* Un segundo en gamemaker son 30 steps, y en este caso se está disparando una bala cada 15 steps, es decir, dos balas por segundo. Primero cuando se crea el avión se pone una variable que dice que tiene una balita, cuando presiona space se prueba la variable de si puede disparar o no, si es verdadera entonces dispara una balita y se pone una alarma para que después de 15 steps se recargue su munición a uno otra vez.
* Cuando el avión choca contra un enemigo, primero se la salud barrita de salud baja 25, a continuación el enemigo es reemplazado por una nueva instancia que es la animación de la explosión, el enemigo no es destruido si no que cambia de posición para que parezca que un nuevo enemigo a aparecido, por ultimo suena el sonido de la explosión.
* el objeto controller\_enemy, administra la creación de aviones enemigos
* la función del evento Alarm es controlar en que cantidad de steps cada nuevo avión enemigo va a aparecer
* el parámetro subimage, significa que el sprite está compuesto de varias imágenes. El -1 indica la subimagen que es dibujada.
* Esta línea de código juega con las probabilidades, entonces es como si lanzaras en este caso un dado de 30 caras para saber cuándo va a salir la siguiente balita enemiga.
* la balita siempre se dirigirá a la posición donde se encuentre el avión principal, el problema es que puede salir un error en el momento en que este avión desaparece momentáneamente cuando pierde una vida, entonces primero se comprueba que la instancia del avión exista, si sí existe la balita se dirigirá a esta posición, si no, se moverá en línea recta.
* Outside room nos permite destruir la balita una vez salga del cuarto, si no se coloca, destruir la balita será una tarea casi imposible, ya que la posición en la que salga es muy impredecible.
* Son power ups para el avión, cuando el jugador pasa de los 400 puntos y luego de los 1000, indican que tendrías más munición disponible.
* [View](http://docs.yoyogames.com/source/dadiospice/002_reference/windows%20and%20views/views/index.html)s: Nos dan un mecanismo para dibujar diferentes partes de tu cuarto en diferentes puntos de la pantalla o para dibujar solo una parte de tu cuarto y que cubra toda la pantalla.
* Porque ahora el jugador solo estará viendo una porción del cuarto y será el jugador quien se desplace no el fondo.
* Es para asegurarse de que el avión continuara moviéndose a cierta velocidad así el jugador no haga nada.
* La vista se moverá a la misma velocidad que el avión, es decir a cada paso se acercara al borde superior del cuarto 2 pixeles
* Cuando movemos arriba o abajo el avión probábamos antes si el avión sobrepasaba el límite inferior o superior, ahora en lugar del borde ponemos variable view\_yview, así se quedara dentro de la vista
* En probar variable reemplazamos la variable y por view\_yview
* Primero probamos si se encuentran en la vista, si sí lo están les damos movimiento
* La vista es considerada una parte del cuarto que vamos a mostrar en pantalla, el puerto es una parte de lo que mostramos en pantalla donde vamos a dibujar la vista.