**Clase 4**

1. Un objeto es donde se fundamenta todo lo queremos que nuestros personajes hagan,

Mientras que un sprite solo es la máscara visual de nuestro objeto, es decir cómo se va a ver gráficamente.

2. Un objeto solo puede tener asociado un único sprite

3. Un asset, son recursos proporcionados para que la creación se mas fácil, como el sprite del jugador o los tiles del nivel.

4. Los eventos es donde entran todas las acciones del jugador

5.

Create: Acciones que ocurren cuando los objetos se crean.

Collision: Acciones que ocurren cuando el objeto colisiona con otro recurso.

Mouse: Acciones que ocurren cuando usas el mouse.

6. Las acciones son comandos programados por el jugador para que un objeto funcione.

7. En los eventos es donde todas las acciones del jugador toman vida y se funcionan.

8. Acciones para destruir o crear un objeto

9. En la instancia en que momento especifico se crea un objeto o se destruye

11. Los steps con eventos en los cuales se agregan comandos que son afectados por el número de steps que tanga la room

**Clase 5**

1.32 pixeles de alto y ancho

2. 32x32

3. La máscara de colisión

4. Es donde el sprite se puede mover con respecto a ese punto, aparece ubicado a la esquina superior derecha del sprite

5. Que ahora es un objeto sólido y todo lo que colisione con el choca.

6. Porque ya como es un sólido cumple su trabajo

7. No se podría avanzar al siguiente nivel.

8. La velocidad del objeto, en fps.

9. 32x32

10. Para que no hayan problemas de colisión

13. Cuando se cumple una serie de objetivos este evento se acciona

14. En destruir la puerta para que el jugador pase

16. Que heredara los mismos eventos que su padre

17. Solamente creas un monstro padre con todos los eventos, y con eso como base creas diferentes sprites que no contengan nada para que el hereden los eventos de su padre.

18. Que si creas un eventos step o creative esto sobrescribirá lo que tengas en tu eventos de parent

19. Un objeto controlador se puede utilizar para varias cosas, como controlar la vida, el puntaje o hasta la cámara del jugador.

21. La profundidad del objeto con respecto a la room

22. Se vería por encima de la bomba.

23. A objeto que esta instancia aplica.

25. Que va relativamente con la speed del room

26. hspeed: Velocidad Horizontal

Vspeed: Velocidad Vertical

**Clase 6**

1. Una variable es algo que se usa para determinar el valor de algo. En el caso de game maker existen varios tipos de variables, de instancia, local, global y de construcción.

2. El límite que permite alcanzar el programa en términos de variable, en otras palabras el área de operación de la variable.

3. Instancia, local, global y de construcción.

4. Respectivamente a la anterior respuesta, son las más comunes definidas dentro de las funciones o eventos de la instancia. Son variables que son creadas y usadas dentro de un ámbito local y luego olvidadas, solo se permite bajo la escritura de var. Una variable que existe dentro de un ámbito global en absolutamente todas las instancias. Y variables que pertenecen respectivamente a un objeto o cuartos.

5. Local

6. Instancia

7. Global

8. Vidas, Barra de vida, Puntaje, Tiempo, Objetos adquiridos

9. Gravedad, fuerza, vida, nombre, armadura, altura, velocidad, posición.

10. Depende de la clasificación que se le haya dado a la variable, sin embargo bajo una clasificación de un mismo nombre game maker las va a confundir.

11. En el avión, y es relativo a la posición del avión.

12.☺

13. Para mostrar la diferencia de planos donde se encuentra el avión y las islas.

14. Crea un territorio al azar de términos de anchura.

15. Está programado para seguir una secuencia. Se cambia el tamaño no el orden.

16. Cambia aleatoriamente las posiciones de los objetos.

17. Cambia constantemente las variables que son llamadas al azar.

18. Al comienzo del juego.

19. Mueve el objeto a la izquierda 4 unidades.

20. Relativo al Avión

21. Evitar que se acumule memoria innecesaria con las balas.

22. Evento de Alarmas

23. Se cambia la alarma para permitir 3 balas antes de la retención

24. Activa animación de explosión del avión y reinicia el juego en posición original.

25. Es el objeto que aparece los enemigos es el que controla en que momento o con que probabilidad van a ir apareciendo los enemigos

26. Es un evento el cual se llama después de esperar un tiempo determinado, y cuando se llama hace las acciones dentro del evento.

27. Es la posición o subimagen que se mostrar del sprite ya que el sprite puede ser una animación y tener muchas subimagenes y -1 es para que corra la animación no se quede estático.

28. Es al azar en una probabilidad de 1 a 30 para disparar, es decir como un dado que tiene 30 caras.

29. Es para que las balas creadas se dirijan hacia el objeto del avión principal.

30. El enemigo se buguearia pues no ha confirmado la existencia del avión principal

31. Es para determinar que los objetos que salen del cuarto sean destruidos y no consuman más memoria del programa.

32. Determinar que tanto puede disparar en un límite de tiempo y en qué situación acorde a su puntaje.

33. Es lo que muestra la pantalla en referencia a la instancia creada con sus objetos.

34. Cuando la vista sigue al jugador es necesario que el fondo se quede quieto para generar la sensación de movimiento.

35. Que el avión va continuamente hacia arriba a 2 unidades p/s.

36. Se usa el comando follow en la opción de View y se le indica seguir al jugador, la velocidad de la cámara es relativa al avión en ese caso.

37. Bajo el comando Follow significa que va a seguir al avión.

38. Creando la bala bajo la condición de variable si Y de posición del objeto es mayor a la de la vista, destruir objeto.

39. Es igual a la condición anterior pero al referente del límite de la vista y a empezar movimiento.

40. View in room es del tamaño del room que parte va a ver el jugador, Port in room es de que tamaño en la pantalla del computador se va a ver ese view.

**Clase 7**

1. No se puede mover la pelota.

2. Ahora su máscara de colisión es mayor y se ve más separado

3. Se verifica si no hay un objeto en la posición específica.

4. Para evitar que la velocidad se incremente en exceso.

5. Para que la velocidad se haga 0

6. Mantiene su velocidad constante anterior y atraviesa los objetos a contacto, porque su velocidad no se convierte en 0

7. Una vez se detecté que el jugador esta sobre un plataforma, este salta y su velocidad vertical se pasa a negativo.

8. Se verifica que si está en el aire su velocidad vertical se sume positivamente.

10. Que para que no haya problemas con la colisión todas las mascara deben ser iguales en cada imagen del sprite animado.

11. Empieza a flotar a colisión con el otro objeto, pasa por la orden de cambio a posición Y 1.

12. Significa que las cajas que determinan el tamaño de colisión de los sprites tienen que estar bien seleccionados, tienen que tener el mismo tamaño a pesar de que cambie el sprite, básicamente que las máscaras sean la misma para que no cause problemas.

13. Se programan como un objeto normal solido al cual se le asigna una parte del tile set en su sprite.