

# Crear Juegos para

# Android

Por Erasmo Ortúño

Blocks

B Group

B Rule

B Timer

Conditions

C Attribute

C AutoRotation

C Collision

C Hibernation



When All of the following are happening  
receives a key press: up is: down

tap or drag in a condition

do

else

Reiniciar

scene.Background.Actor 2.position

ONE WAY

## En solo 18 Horas

¡Sin Conocimientos Previos!

# Crear Juegos para

# Android

Por Erasmo Ortúñoz

- ▼ Blocks
- Group
- Rule
- Timer
- ▼ Conditions
- Attribute
- AutoRotation
- Collision
- Hibernation

scene.Background.Actor 2.position

ONE WAY

Reiniciar

## En solo 18 Horas

¡Sin Conocimientos Previos!

# *Índice*

<u><a href="#">ACERCA DEL AUTOR</a></u>	.....
<u><a href="#">6 PREFACIO</a></u>	.....
<u><a href="#">7 CREAR UNA IDEA</a></u>	.....
<u><a href="#">8 OBTENIENDO GAMESALAD</a></u>	.....
<u><a href="#">9 Requerimientos del Sistema</a></u>	.....
<u><a href="#">9 Descargar Gamesalad</a></u>	.....
<u><a href="#">10 CONOCIENDO GAMESALAD</a></u>	.....
<u><a href="#">13 La Ventana Library</a></u>	.....
<u><a href="#">16 SCENE</a></u>	.....
<u><a href="#">16</a></u>	

Layers.....

20

Actors.....

21

Media.....

25

## Behaviors

28

1. Group

28

2. Rule

28

3. Timer

28

4.

Attribute.....

28

5. Autorotation

28

6.

Collision.....

28

7. Hibernate

28

8. Key

<u>29</u>	
<u>9. Mouse Button</u>	<u>29</u>
<u>10. Mouse Position</u>	<u>29</u>
<u>11. Touch</u>	
<u>29</u>	
<u>12. Accelerate</u>	
<u>29</u>	
<u>13. Accelerate</u>	
Toward.....	<u>29</u>
<u>14. Add/Remove</u>	
Row.....	<u>29</u>
<u>15. Animate</u>	
<u>29</u>	
<u>16. Change Attribute</u>	<u>29</u>
<u>17. Change Image</u>	<u>29</u>
<u>18. Change Scene</u>	<u>29</u>
<u>19. Change</u>	
Size.....	
<u>29</u>	
<u>20. Change Table Value</u>	
	<u>29</u>
<u>21. Change Velocity</u>	
	<u>30</u>
<u>22. Collide</u>	
<u>30</u>	
<u>23. Constrain attribute</u>	

<u>24. Control</u>	30
<u>Camera</u>	
30	
<u>25. Copy Table</u>	30
<u>26. Destroy</u>	
30	
<u>27. Display</u>	
<u>Text</u>	
30	
<u>28. Interpolate</u>	30
<u>29. Keyboard Input</u>	30
<u>30. Load</u>	
<u>Attribute</u>	
30	
<u>31. Log Debugging Statement</u>	30
<u>32. Move</u>	
<u>Move</u>	
30	
<u>33. Move</u>	
<u>To</u>	
30	
<u>34. Note</u>	
<u>35. Particles</u>	
<u>36. Pause Game</u>	31
<u>37. Pause Music</u>	

	.....	31
<u>38. Play</u>		
Music.....		
31		
<u>39. Play Sound</u>		31
	.....	
<u>40. Replicate</u>		
	.....	
31		
<u>41. Reset</u>		
Game.....		
31		
<u>42. Reset</u>		
Scene.....		
31		
<u>43. Rotate</u>		
	.....	
31		
<u>44. Rotate to</u>		
Angle.....		
31		
<u>45. Rotate to Position</u>		31
	.....	
<u>46. Save Attribute</u>		31
	.....	
<u>47. Save</u>		
Table.....		
31		
<u>48. Spawn</u>		
Actor.....		
31		
<u>49. Stop Music</u>		31
	.....	
<u>50. Unpause Game</u>		31
	.....	

<b>Tables.....</b>	
<b>32</b>	
<b>Stage</b>	
<hr/>	
<b>35</b>	
<b>Mover</b>	
<b>Actores.....</b>	
<b>39</b>	
<b><u>Animar Actores</u></b>	
<hr/>	
<b>46</b>	
<b><u>Hacer un Fondo para el juego</u></b>	
<hr/>	
<b>48</b>	
<b><u>Colocar una Cámara</u></b>	
<hr/>	
<b>52</b>	
<b><u>Colisionar y Rebotar</u></b>	
<b>Actores.....</b>	
<b>56</b>	
<b><u>Colisionar y Desaparecer Actores</u></b>	
	<b>59</b>
<b><u>Adherir Sonidos Al Juego</u></b>	
<hr/>	
<b>61</b>	
<b><u>Adherir Gravedad a un Actor</u></b>	
<hr/>	
<b>63</b>	
<b><u>Crear y Realizar Disparos</u></b>	
<hr/>	
<b>65</b>	

<b><u>Mostrando y creando un Score</u></b>	<b>69</b>
<hr/>	
<b><u>Crear Escenas</u></b>	
<hr/>	
<b>78</b>	
<b><u>Hacer que Enemigo Ataque el</u></b>	
<b>Jugador.....</b>	<b>84</b>
<b><u>Juego para Android</u></b>	
<hr/>	
<b>85</b>	
<b><u>Descargar el SDK para Android</u></b>	
<hr/>	
<b>85</b>	
<b><u>Crear una Keystore</u></b>	
<hr/>	
<b>88</b>	
<b><u>Instalando tu Telefono</u></b>	
<hr/>	
<b>89</b>	
<b><u>Instalando el Visor de</u></b>	
<b>Gamesalad.....</b>	
<hr/>	
<b>89</b>	
<b><u>Probando un dispositivo Android usando el</u></b>	
<b>visor de Gamesalad .....</b>	<b>89</b>
<b><u>Usar el Teléfono o Tableta vía USB</u></b>	
<hr/>	
<b>90</b>	

## Por medio de Wi-Fi

---

**90**

*Interactuar con el Acelerómetro*

91

*Incluir Botones en pantalla para Juegos*

*Android*..... 97

# **Crear Juegos para Android (En tan solo 18 horas)**

Derechos de Copia © 2013 thenacoosweb.com

Todos los Derechos Reservados, ninguna parte de este libro debe ser reproducida, almacenada o copiada en cualquier forma física o sistema informático, sin el previo aviso de su Autor.

Todos los esfuerzos vertidos en este libro se han preparado para asegurar toda la información presentada, sin embargo la información contenida en el mismo es vendida sin una garantía, thenacoosweb.com y el autor del libro no se hacen responsables por cualquier daño causado o alegado directamente o indirectamente por este libro.

Thenacoosweb.com es un sitio especializado en brindar información técnica y de conocimientos en el uso de software, sistemas informáticos y Ciencias exactas, para el auto aprendizaje.

Primera publicación: Marzo 2013

Publicado por thenacoosweb.com Manuel Bueno 121  
Paraiso Tabasco, México.

[www.thenacoosweb.com](http://www.thenacoosweb.com)

# ACERCA DEL AUTOR

**Erasmo Cesar Ortúñoz Morales** es un desarrollador de videojuegos y webmaster fundador del sitio web [www.thenacoosweb.com](http://www.thenacoosweb.com) y el canal thenacoos en youtube.com, es creador del juego Mastermaticas Pro, Mastermaticas Lite y Unit Converter STEP, las cuales se pueden encontrar en la tienda de apps para Teléfonos Móviles con sistema Operativo Android, Google Play.

Originalmente **Erasmo Cesar Ortúñoz Morales** conocido por sus familiares y amigos como **Eritos**, inicia en el área de la informática por curiosidad y hobby en el año 2000, debido a que adquiere una computadora Pentium III en un mercado de computadoras viejas en la ciudad de México, es en ese año que empieza su camino por el conocimiento de la programación web, base de datos y lenguajes de programación como autodidacta, en el año 2002 termina la Carrera en Relaciones Comerciales, una profesión no tan afín a la informática, a la postre Eritos se dedica a aprender todos los secretos de cómo crear videojuegos, los cuales eran muy bien salvaguardados por las casas desarrolladoras de juegos, en este largo camino aprende técnicas y adquiere conocimientos para desarrollar sus propios videojuegos.

En el año 2010 Eritos adquiere los conocimientos suficientes para crear y desarrollar videojuegos para dispositivos móviles, mas sin embargo su descubrimiento más oportuno fue encontrar una herramienta diseñada especialmente para esto, Gamesalad, de la cual Eritos se da cuenta que es una herramienta de fácil

aprendizaje y que permite crear videojuegos para varias plataformas como son, Windows, MacBook, iPhone, Tablets, Android.

# PREFACIO

Actualmente la mayoría de las personas están inmersas en el mundo tecnológico de la última década, este mundo tecnológico primero fue dominado por las computadoras científicas las cuales solamente tenían instituciones como la NASA o los Bancos, después de eso llegó la **PC (personal computer)**, La computadora personal cambio al mundo para siempre, todo mundo podía tener acceso al conocimiento y agilizar las tareas, pero el hombre siempre pensante, veía con reservas que este dispositivo tan poderoso tenía un pequeño impedimento, no se podía trasladar a cualquier parte, de esta idea nace la computadora Portátil mejor conocida como Laptop, pero en esos mismos años el boom del celular ya estaba en auge, y en alguna parte de los últimos años a alguna corporación o empresa se le ocurrió unir el poder de la laptop con el de un teléfono móvil, y esa corporación daría vida al teléfono inteligente mejor conocido como iPhone, si exactamente esa corporación o empresa es Apple, hoy en día la mayoría de los jóvenes y personas menores de 45 años cuentan con un SmartPhone (Teléfono inteligente), y estos teléfonos para funcionar, jugar con ellos, escuchar música, ver películas o usar un software. Necesitan unos pequeños programas que se llaman apps.

Las apps en estos momentos son los programas más solicitados y requeridos por todo aquel que cuenta con un Smartphone, pero especialmente existe un tipo de app que seguramente todos tienen en su teléfono inteligente, esta app seguramente será un videojuego, así es, las apps más

descargadas en iTunes o en Google Play, son juegos, y es por ende uno de los mercados más redituables para cualquier desarrollador, este libro abordara técnicamente el desarrollo de juegos para teléfonos inteligentes en un sentido práctico, sencillo y fácil de aprender, utilizando GAMESALAD.

# CREAR UNA IDEA

Bueno ya has decidido crear un videojuego para algún teléfono móvil, claro hasta ese momento parece una excelente idea, vas a crear seguramente algún juego con el nombre y la temática muy parecida al mega éxito de Angry Birds, o bien el próximo Plantas vs Zombies, pero te voy a hacer bajar de esa gran nube, en primer lugar crear un juego es algo serio. Y para ser honesto toda idea original siempre va a ser un éxito. ¿Por qué? Bueno si miras la historia del ser humano visualizaras con mucha razón que el primero de la fila, siempre es el que se lleva la mejor tajada de pastel y eso en el desarrollo de juegos para teléfonos móviles también es un hecho.

A mi parecer y si miras el comportamiento de todos los desarrolladores exitosos, es que nunca andan imitando, se apegan a un simple plan, este es el de inventar una idea, innovar una idea, o mejorar una idea, pero nunca a imitar, porque saben que sería un fracaso, el plan de inventar, innovar o mejorar, no es parte de algún libro de informática o de algún científico desarrollador de sistemas espías, sino que es parte del plan de empresas, y te digo con seguridad que un excelente empresario sabe que esta es una verdad absoluta, y por mis estudios en Relaciones Comerciales, es que he podido relacionar esta pequeña sintaxis que aplican los programadores de apps hoy en día, para tener juegos exitosos y con ganancias millonarias. Así que si quieres crear un juego que solamente tú vas a jugar no necesitas pensar una buena idea, pero si quieres crear un juego excelente y de calidad y que además agrade a miles de personas, siempre se ORIGINAL.

# OBteniendo GameSalad

Ha llegado el gran momento y seguramente ya quieras empezar a desarrollar juegos para iPhone y Android, claro que ya vamos a empezar y déjame recordarte que para caminar hay que gatear primero, de igual forma es importante que leas esta parte del libro, ya que te librara de muchas horas perdidas en la instalación y configuración de GameSalad.

# **Requerimientos del Sistema**

Para hacer funcionar Gamesalad necesitas los siguientes requerimientos de sistemas:

## **Mac:**

Mac OSX Lion o superior. Xcode 4.3 con Herramientas Línea de Comandos e iOS. Android SDK para soporte Android.  
Procesador Core 2 Duo o Superior. 2GB de memoria RAM como mínimo.

## **Windows:**

Windows Vista, Windows 7 o Windows 8. Android SDK para Android  
Procesador Core 2 Duo o Superior. 2GB de memoria RAM como mínimo.

Si tu Computadora de escritorio, laptop o Mac, cuenta con los requerimientos mínimos mencionados arriba, entonces felicidades, y si no los tienes, no podrás hacer correr Gamesalad.

## **Descargar Gamesalad**

Para obtener una versión Gratuita de Gamesalad deberás dirigirte a la siguiente dirección web [www.gamesalad.com](http://www.gamesalad.com) y presionar en el botón “Download Creator for Free” como se muestra en la imagen a continuación.

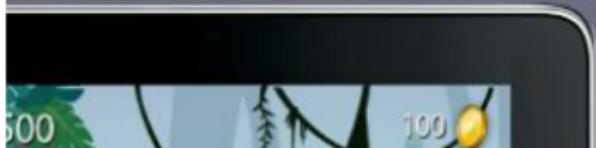
Create Games Rapidly  
Drag & Drop, No Code

Download Creator for Free

Develop on and publish to all major platforms

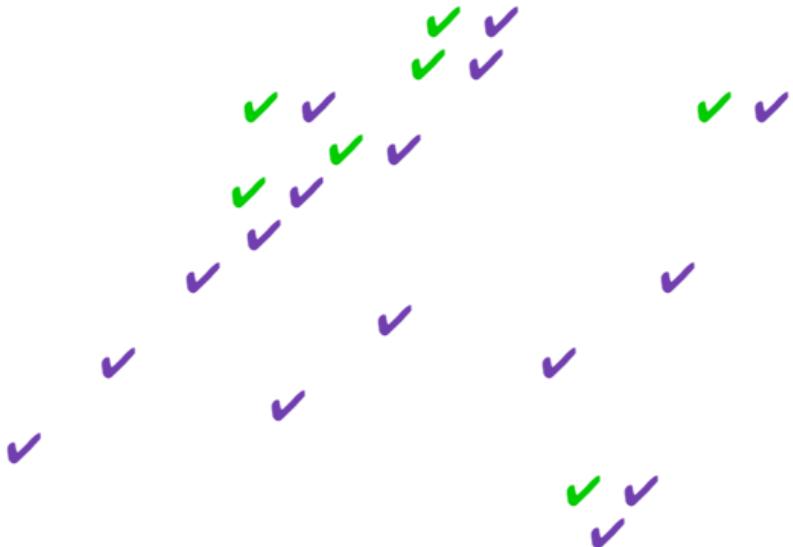


Cuando se esté descargando Gamesalad, crea una cuenta con tu correo electrónico en [www.gamesalad.com](http://www.gamesalad.com) en la imagen debajo se muestra la opción para crear la cuenta.

[Publish Games ▾](#)

Existen dos versiones de gameSalad, la versión gratuita la cual te he sugerido y la versión de pago, veremos a continuación una tabla con las prestaciones y las diferencias entre las dos.

#### **Prestaciones Version Basic o Gratis Version PRO**

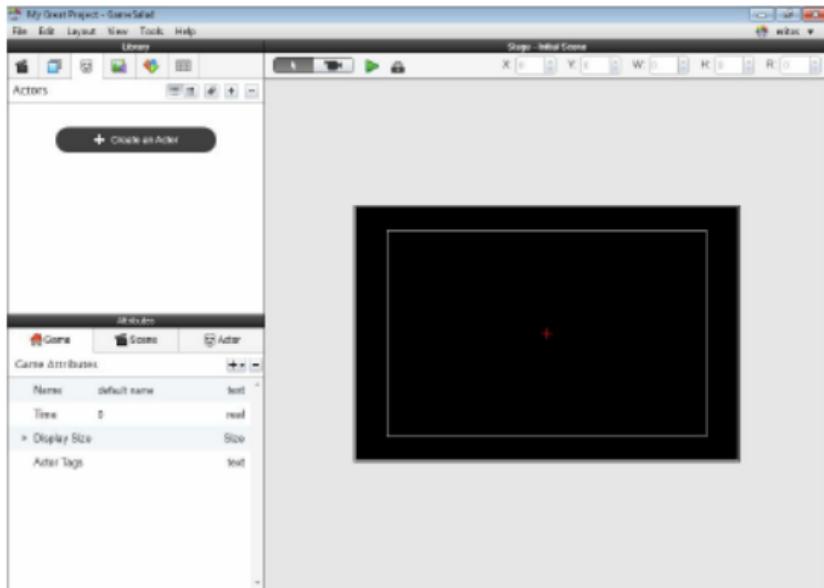


## Gratis \$299 USD

Como puedes ver la versión Basic se encuentra muy limitada y no nos dejara crear juegos para Android o Windows 8, y te recomiendo finalizar el aprendizaje de este libro con la versión Basic, cuando hayas finalizado la lectura, puedes adquirir la versión PRO la cual te dejara publicar tu juego para sistema móviles, en este momento lo más importante es que aprendas a utilizar Gamesalad, recuerda que Gamesalad es un Motor para videojuegos para exportar un solo trabajo a diferentes plataformas. Así que no te preocupes, descarga la versión Basic.

# CONOCIENDO GAMESALAD

Al iniciar Gamesalad veremos a continuación esta pantalla llamada el **Scene Editor** que es en si el programa completo de gamesalad y en donde pasaremos la mayor parte de nuestro tiempo, desarrollando el videojuego.



En esta pantalla tenemos las siguientes prestaciones:

Las típicas opciones de un software, como **File**, **Edit**, **Layout**, **View**, **Tools**, **Help**. En **File** nos encontramos con una serie de Sub Opciones como **New** que sirve para crear un nuevo proyecto.

**Open** nos sirve para abrir un proyecto ya existente.

**Open Recent** abrirá el proyecto más reciente con el que hayamos estado trabajando.

**Open Dashboard** abre una ventana parecida a un escritorio en el cual podemos las opciones de Crear un nuevo proyecto y abrir el proyecto más reciente, de igual forma contamos con un

icono de correo electrónico en donde nos llegaran noticias de Gamesalad, también existen tres iconos más abajo, en el icono del libro son preguntas de problemas que encuentran los desarrolladores y son respondidas por el staff de Gamesalad, El icono del Carrito de Compras es para adquirir de la tienda de gamesalad plantillas de juegos, código, imágenes, fotogramas, audio, etc. Todo lo que necesita un juego lo puedes encontrar en el Marketplace de Gamesalad, y el último icono que tiene un portafolio, es en donde administraras los juegos que vayas creando.



**Close** es para cerrar el proyecto en Gamesalad.

**Save As** es el equivalente a Salvar Como (los archivos de Gamesalad se guardan con la extensión de archivo .gsproj)

**Publish** nos permitirá publicar nuestro juego (primero te tienes que loguear para hacerlo)

**Export** es para exportar nuestro proyecto.

**Preferences** es para administrar las preferencias del software.

**Quit Gamesalad** es para salir del Software.

El Botón **Edit** es el equivalente a Editar y cuenta con las siguientes opciones:

**Undo** es para regresar al estado anterior cuando así lo deseemos. **Redo** es para adelantar hacia el estado corregido. **Cut** es para cortar algún objeto o imagen en Gamesalad.

**Copy** es para copiar algún objeto, elemento o imagen. **Paste** es para pegar el objeto, elemento o imagen copiada. **Delete** es para borrar el objeto, elemento o imagen seleccionada.

**Select All** Selecciona todos los elementos en la escena.

Botón **Layout** es para colocar las imágenes en primera, segunda u otra posición en la escena de Gamesalad, algo muy importante a la hora de colocar las imágenes u objetos, imaginemos una naranja, La Cascara de la Naranja ocupa la primera posición, La segunda posición la ocupa la cascara de la fruta y la tercera posición el fruto en si, en donde se encuentra el jugo. Pues bien lo mismo ocurre en un juego, vamos a ver las opciones de este botón.

**Bring Forward** equivalente a acercar.

**Bring to Front** equivalente a traer al Frente de la escena.

**Send Backward** enviaría al objeto o imagen hacia atrás.

**Send To Back** enviar hacia el fondo.

Botón **View** que significa Vista y no tiene muchas opciones, la opción **Debugger** es para depurar el juego.

De igual Forma el botón **Tools** solamente cuenta con una opción que es **APK Signer** y nos servirá para firmar nuestra aplicación cuando deseemos subirla al Marketplace de Google.

**Help** nos servirá para obtener ayuda acerca de Gamesalad y cuenta con las siguientes opciones:

**About** que es Acerca y nos muestra la siguiente ventana con información acerca de la versión de Gamesalad.

**Gamesalad Help** nos lleva hacia la página de ayuda de Gamesalad.

**Check for Updates** es para verificar si existen actualizaciones de Gamesalad, si existiera alguna actualización el programa pasara a la versión más reciente.

**Report Bug...** Sirve para reportar a los desarrolladores de Gamesalad si existieran errores en el software.

## La Ventana Library

En Library es en donde colocaremos **Scenes** (Escenas), **Layers** (Capas), **Actors** (Actores), **Media** (es en donde se adhieren imágenes y Audio), **Behaviors** (Comportamientos), **Tables** (Se pueden Crear Tablas para almacenar datos).

## Library



Actors



**+ Create an Actor**

# SCENE

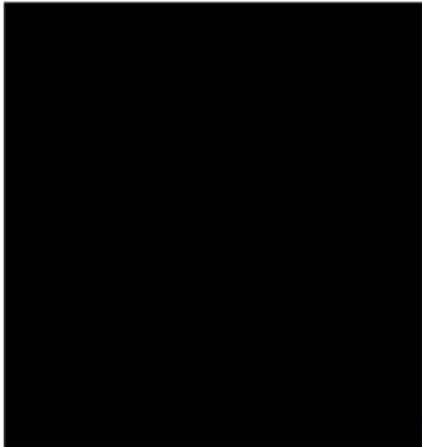
Vamos a ver el primer módulo que es Scene, en este módulo es en donde vamos a crear todas las escenas de nuestro videojuego, como pueden ser, el Splash de entrada, el menú de botones o inicio, una escena de inicio del juego, escena de fin del juego, escena en donde se desarrolla el videojuego, niveles, backgrounds, etc.

## El botón con los cuadros

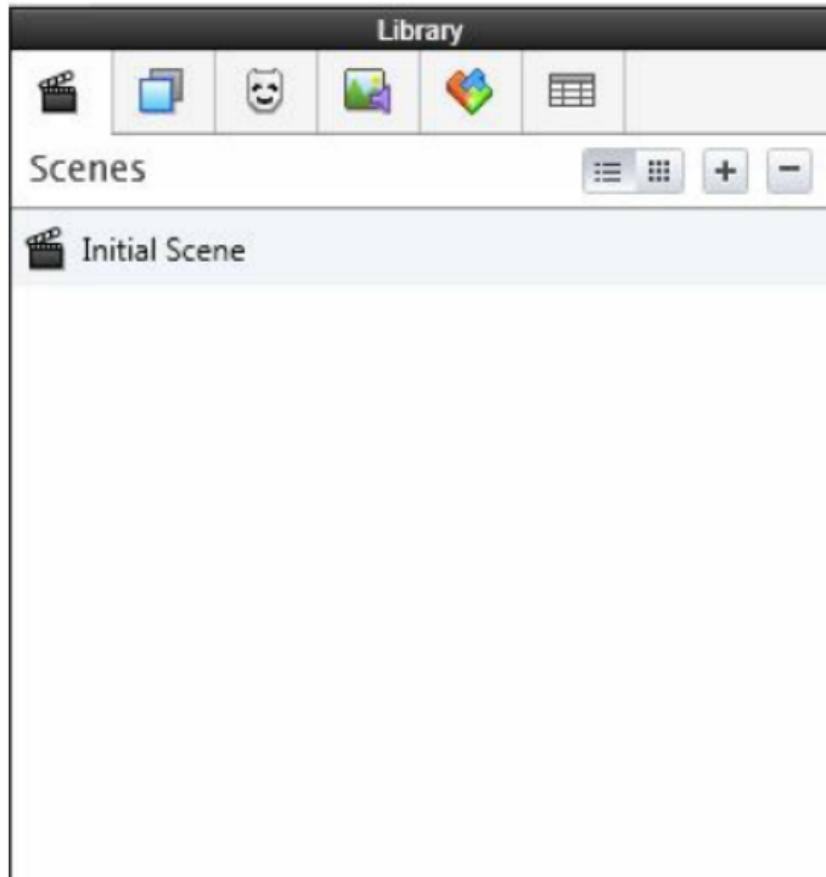
Como puedes ver en Scene acontece todo, ahí mismo deberemos adherir nuestros personajes, y configurar el tamaño de la escena del juego en cuestión. **El botón con el símbolo + sirve SCENE** luce como en la imagen.



para adherir una Escena al juego



El botón con el símbolo  
- sirve para eliminar una Escena al juego  
Este botón mostrara las  
Escenas en forma de lista



En la imagen puedes ver que existen cuatro botones en el top los cuales nos servirán para administrar las escenas. De igual forma cuando iniciamos Gamesalad por defecto nos aparece una escena con el nombre "Initial Scene".

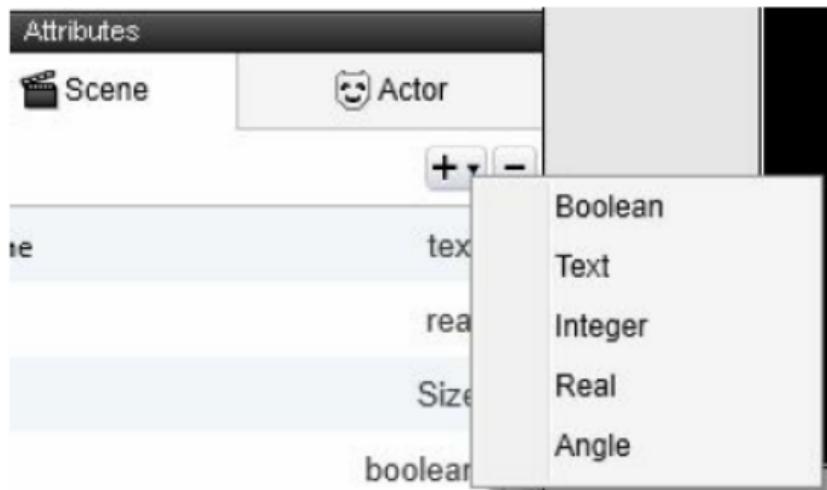
Las Escenas en Gamesalad cuentan con una serie de

propiedades que puedes encontrar en la ventana “Attributes”

The screenshot shows the Unity Editor's "Attributes" window. At the top, there are three tabs: "Game" (selected), "Scene" (disabled), and "Actor". Below the tabs, the title "Initial Scene" is displayed, followed by a set of buttons for adding (+) or removing (-) attributes. The main area contains a list of attributes:

Name	Initial Scene	Type
Time	0	real
► Size		Size
Wrap X	<input type="checkbox"/>	boolean
Wrap Y	<input type="checkbox"/>	boolean
► Gravity		point
► Color	<span style="background-color: black; color: black;">#000000</span>	Color
► Camera		rect
► Autorotate		attributes

En la ventana **Attributes** pestaña **Scene** podemos ver que existen una serie opciones, como puedes ver en la parte derecha hay una serie de textos que dicen text,real,Size,boolean,point,Color,Etc. Y es que en realidad se trata de variables, estas variables pueden ser del tipo boolean,text,integer,Real,Angle. Y puedes crear las tuyas con el botón que se ve en la imagen.



Es muy importante que distingas y conozcas las variables a crear, a continuación la traducción de cada uno de los significados.

Boolean es booleano y los únicos valores que puede tener es verdadero y falso. Text es texto y puede contener caracteres como alfabeto y números. Integer es Entero y solamente puede contener números enteros.

Real es para número Reales.

Angle es para ángulos.

Regresando a las opciones que vienen por defecto encontramos lo siguiente: Name es en donde le pondremos algún nombre a la escena.

Time es para ponerle un tiempo a la escena.

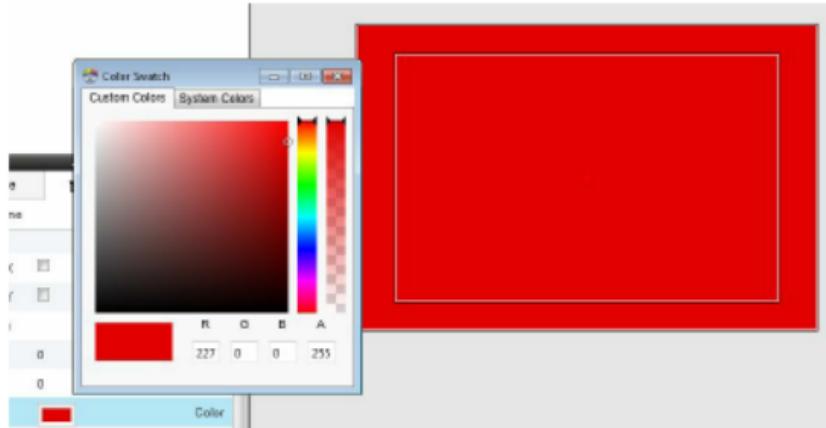
Size es para otorgarle un tamaño a la escena, una escena puede tener el tamaño

del dispositivo o bien un tamaño opcional, esto dependerá del tipo de juegos que se vayan a desarrollar. Cuenta con las Variables Wrap X, Wrap Y, y sus valores son del tipo booleano, o sea Verdadero o Falso, al poner verdadero en estas casillas, nuestro objeto o personaje no se perderá en el infinito, como veremos más adelante.

Gravity se refiere a adherir gravedad al juego, así es para darle realismo al juego este debe tener gravedad y en gamesalad no se han olvidado de este opción, como puedes ver cuenta con dos variables del tipo real una en X y la otra en Y, están hechas con tipo de variable real, si aplicamos un valor ya sea negativo o positivo el objeto o personaje se moverá en los ejes aplicando un tipo de fuerza gravitacional.

▼ Gravity		point
X	0	real
Y	0	real

Color es para cambiar el color de fondo en nuestra escena, si presionas en el rectángulo de color negro aparece una pequeña ventana en donde puedes elegir el color del fondo. Si desplegamos Color se muestra una serie de opciones RGB y Alpha para administrar la transparencia de un objeto.



Camera esta opción como ya te has imaginado es para administrar la Cámara, cuenta con las opciones Origin, Size y Tracking Area. Origin es Origen y es ahí en donde especificaremos en las coordenadas X y Y en qué posición estará la cámara, Size es el tamaño de la Cámara, por defecto cubre toda la pantalla pero con los valores Width específicas el ancho de la cámara, y con Height el Alto, Tracking Area que el área de seguimiento de la Cámara, puedes especificar el tamaño del área a seguir en el videojuego. Cuenta con valores en Width, Height y Rotation, este último es para rotar la cámara de 0 a 360 grados.

Attributes		
Game	Scene	Actor
Initial Scene	+ -	
▼ Camera	rect	
▼ Origin	point	
X	0	real
Y	0	real
▼ Size	Size	
Width	480	real
Height	320	real
▼ Tracking Area	Size	
Width	400	real

**Autorotate** es el comportamiento que tendrá el juego al poner el teléfono móvil o celular en posición vertical u horizontal, cuenta con 4 opciones Landscape Left que es horizontal hacia la izquierda, Portrait en forma de porta retrato que es en la forma vertical, Landscape Right significa Horizontal hacia la derecha y

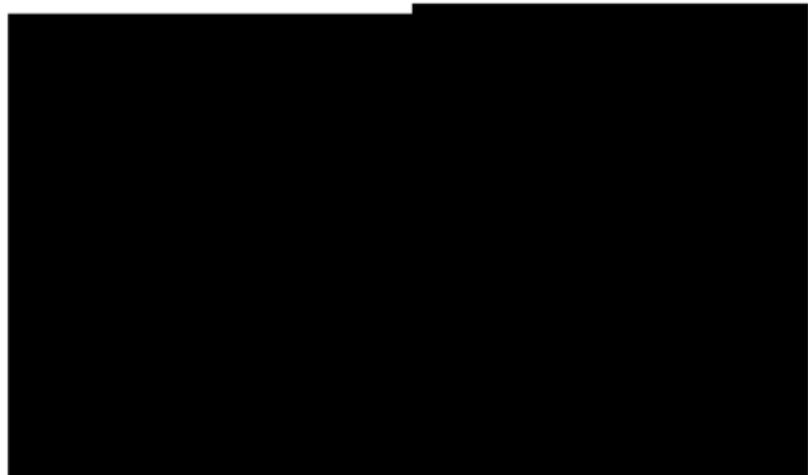
Portrait Upside Down significa en forma de Porta retrato con la parte de superior hacia abajo.

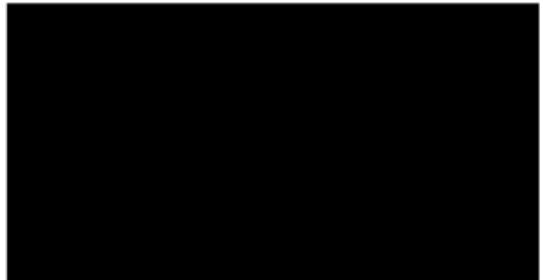
▼ Autorotate		attributes
Landscape Left	<input checked="" type="checkbox"/>	boolean
Portrait	<input type="checkbox"/>	boolean
Landscape Right	<input type="checkbox"/>	boolean
Portrait Upside Down	<input type="checkbox"/>	boolean

Aquí finalizamos las opciones con las que cuenta la pestaña **Scene**, a continuación la Pestaña **Layers**.

## **Layers**

Cuenta con las mismas opciones que la pestaña Scene pero con una pequeña diferencia. Que cuenta con dos botones para administrar las posiciones de las Layers, antes de que inicie a explicar para que sirve esta pestaña, te voy a recordar que las Layers (capas) son importantísimas, porque es ahí en donde vas a colocar los objetos o personajes del videojuego, siempre en posiciones de capas, por ejemplo vamos a suponer que existen tres objetos en el juego, un personaje, enemigo y un fondo, el personaje puede ocupar la primera posición, el enemigo la segunda capa y el fondo la última posición.

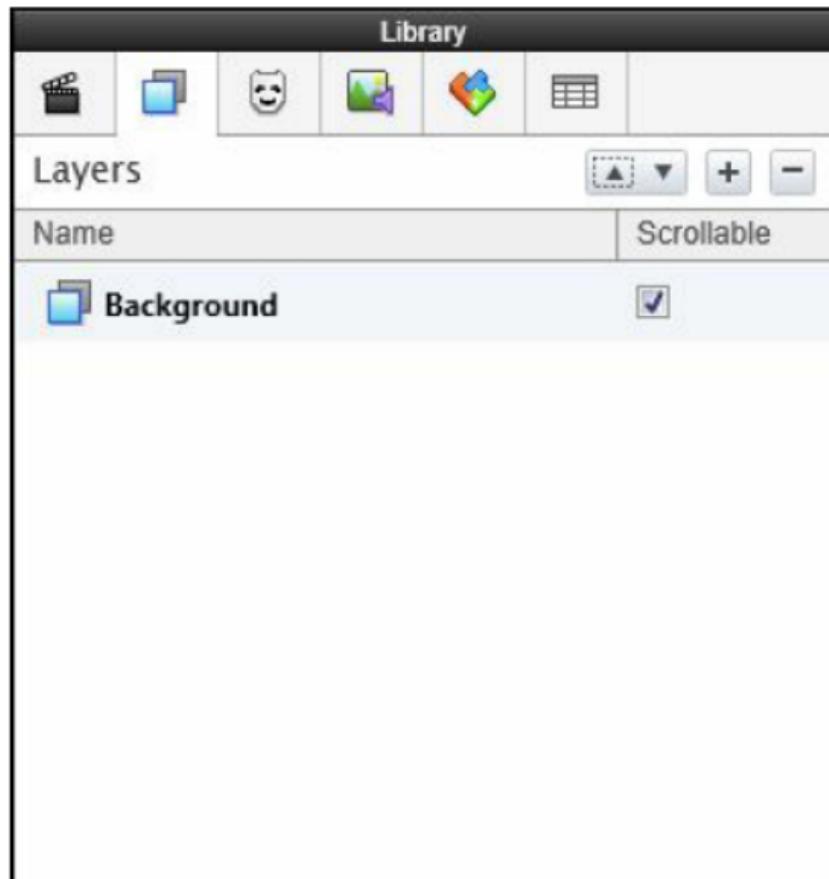




El botón con el símbolo + sirve para eliminar una capa existente.

El botón con el símbolo + sirve para adherir una Layer Nueva.

Estos dos botones sirven para posicionar las capas. Arriba o abajo.



En la pestaña de las Capas o Layers existe una opción del tipo Booleano que se llama Scrollable, esta opción es muy importante que la aprendas de memoria, cuando deseleccionamos esta opción en la capa el objeto no se moverá para nada y se moverá acorde la cámara, ¿sabes que objetos nunca se mueven en un videojuego y deben estar fijos? ¿y a lo

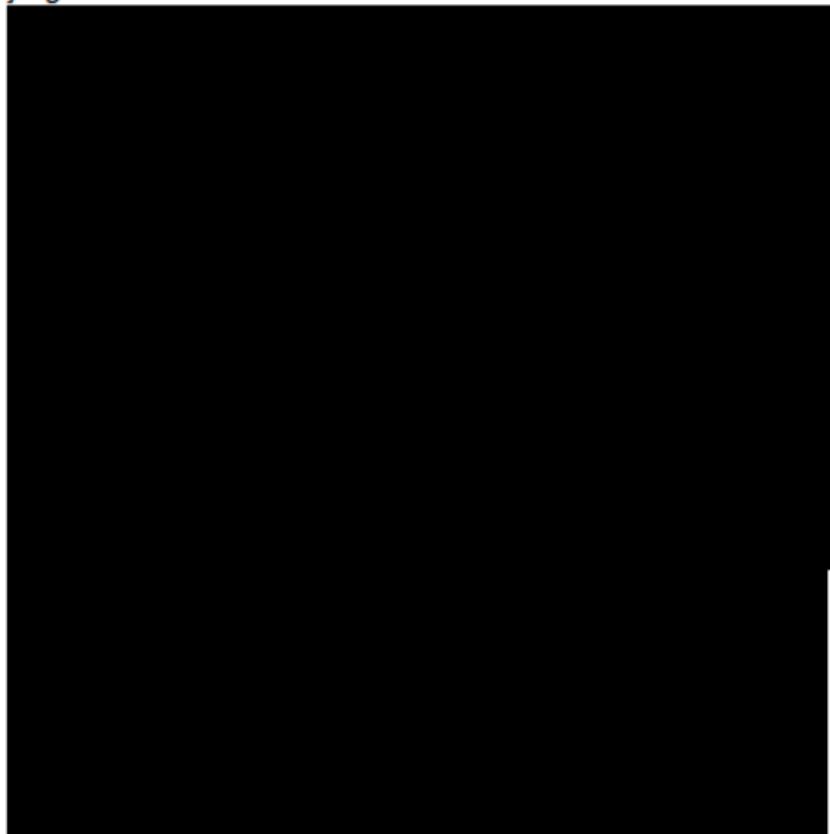
pensaste? Bueno pues por si no te has dado cuenta, son los botones de movimiento, disparo, opciones de menú, pausa y demás botones que deben estar disponibles, así que cuando vayas a hacer algunos de estos objetos, recuerda deseleccionar Scrollable.

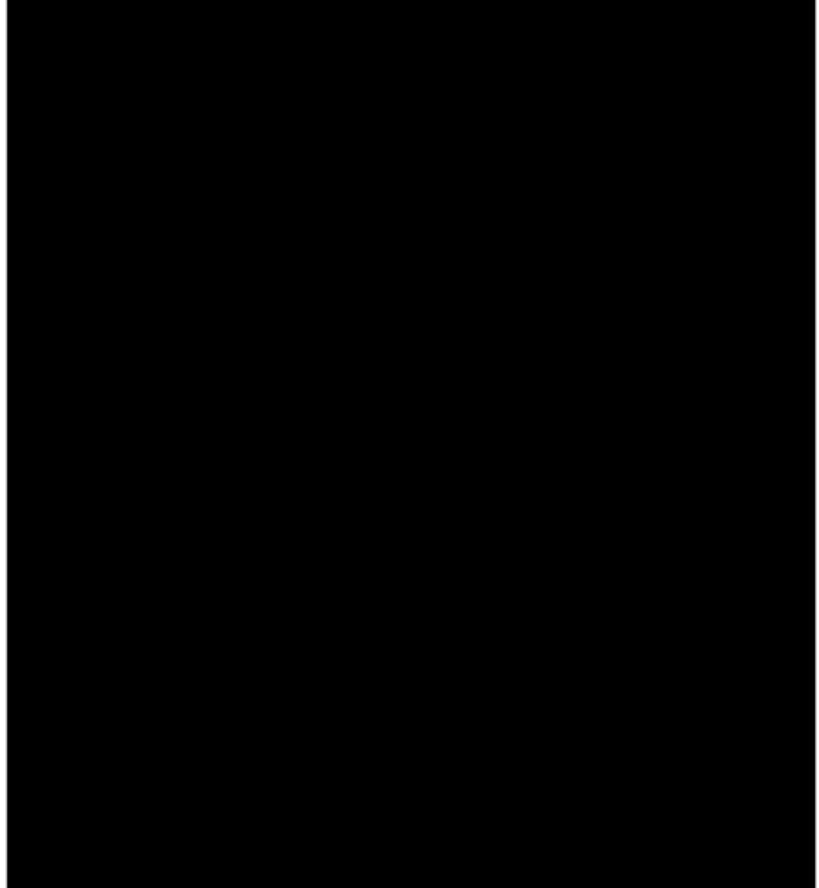
Layers					
Name		Scammable			
 Background		<input checked="" type="checkbox"/>			

Veremos a continuación la pestaña Actors.

## Actors

La pestaña actors es muy importante porque como su nombre lo dice, ahí es donde iremos colocando nuestros actores en el juego.





**El botón con símbolo + adhiere nuevos actores y  
el botón – elimina un actor.**

**Estos dos botones son para**

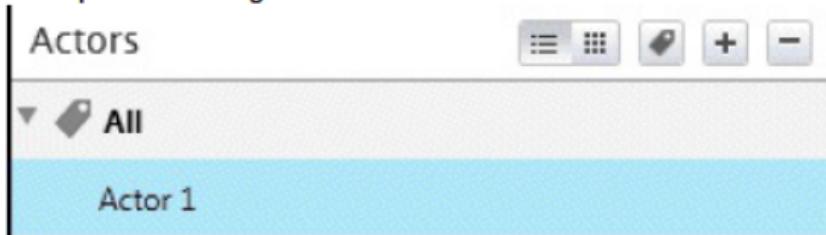
mostrar los actores en forma de  
lista o iconos.



Este botón se llama Tags y significa

**Etiquetas en nuestro idioma, con esta opción podemos etiquetar los actores, y más adelante veremos porque es importante.**

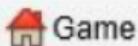
Para adherir un actor solamente presionamos en el botón + y nos aparecerá lo siguiente.



The screenshot shows a software window titled "Actors". At the top right is a toolbar with five icons: a list view, a grid view, a pencil for edit, a plus sign for add, and a minus sign for delete. Below the toolbar, there is a dropdown menu with a tag icon and the word "All". Underneath this is a list item "Actor 1", which is highlighted with a blue background. To the left of "Actor 1" is a vertical line and a downward arrow.

Vemos que aparece una opción de etiqueta en donde dice ALL y si la desplegamos otra opción que dice Actor 1. Haciendo doble clic sobre Actor 1, podemos cambiar el nombre. Si presionamos sobre Actor 1, podemos ir a la pestaña Actor en Attributes y nos aparecen las opciones que podemos poner, modificar o eliminar el Actor 1.

## Attributes



Game



Scene



Actor

Actor 1



Drag an Image Here

Name	Actor 1	text
------	---------	------

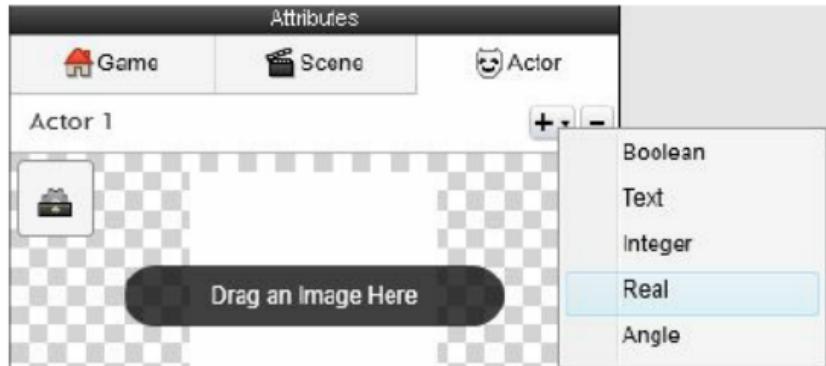
Time	0	real
------	---	------

▶ Position	point
------------	-------

▶ Size	Size
--------	------

Si creas un Actor, pedirá poner una imagen aquí, cuando carguemos imágenes desde la pestaña Media podemos arrastrar el elemento y dejarlo caer en esta opción.

Las opciones de Actor también nos permiten colocar Variables como lo hicimos con Scene



Y de igual forma son del tipo Boolean, Text, Integer, Real, Angle, modificar, adherir o borrar variables nos permitirá mejorar el juego en cuestión.

Las Variables que traen predefinidas el actor son:

Name que sería ponerle un nombre al Actor.

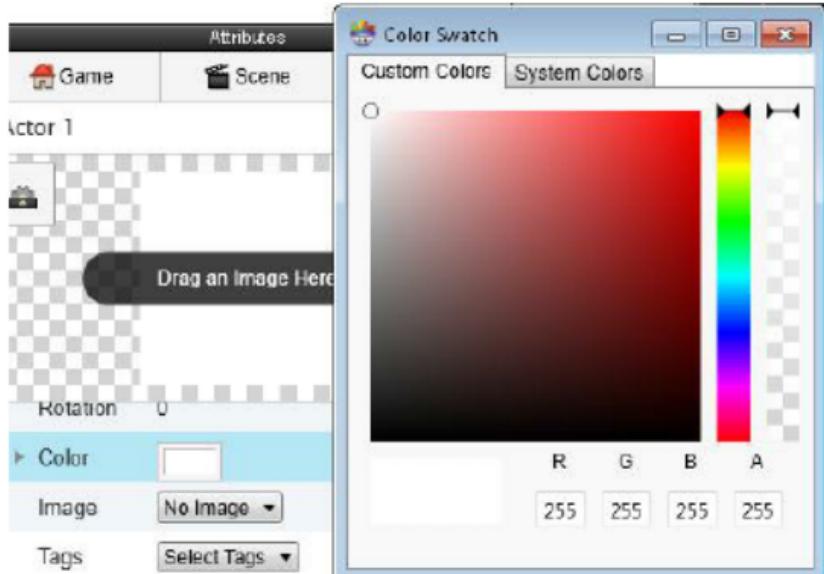
Time es colocar un tiempo al actor.

Position es para colocar el actor en una posición, utilizando siempre X y Y. Size es para especificar el tamaño del actor en pantalla, cuenta con los valores

Width y Height. Por defecto trae los valores 120.

**Rotation** es para colocar una rotación al Actor de 0 a 360. Color en realidad esta casilla carece de mucha importancia una vez que asignas

una imagen al actor, pero si deseas crear bloques, presionando el rectángulo con el color blanco por defecto, puedes cambiar el color de tu actor.



**Image**, se usa para seleccionar que hayamos adherido a Gamesalad Tags son Etiquetas y recuerda que ya lo habíamos visto en la opción de layers, ahora te voy a explicar para que sirven, supongamos que tenemos un bloque que hace la función del suelo, si quisieramos que el actor saltara entre plataformas podríamos usar otros bloques para que salte de plataforma a plataforma, para esto usamos las etiquetas, todos

los bloques se vayan creando los colocamos en una etiqueta llamada previamente en Layers "bloques para suelo" y de esta forma podemos ahorrar tiempo.

**Preload Art** es Precargar el arte, todo lo que sean imágenes seleccionadas se precargaran en Gamesalad.

**Graphics** es para los gráficos, y cuenta con varias opciones, Visible es para mostrar u ocultar el Grafico, Blending Mode es para fusionar el grafico cuenta con sub opciones como Normal, Opaque, Additive, Screen y Multiply.

**Horizontal Wrap** es como poner una envoltura al grafico pero horizontalmente.

**Vertical Wrap** es parecida a la opción anterior pero la envoltura es verticalmente.

**Horizontal Anchor** es anclar horizontalmente la imagen.

**Vertical Anchor** anclar verticalmente la imagen.

**Flip Horizontally** es voltear la imagen horizontalmente.

**Flip Vertically** equivale a voltear la imagen verticalmente.

**Tile Width** es elegir el ancho de la imagen.

**Tile Height** es elegir la altura de la imagen.

**Motion** es en donde configuraremos como se va a mover el actor en el juego, cuenta con las opciones Linear Velocity, Angular Velocity, Max Speed, Apply Max Speed. **Linear**

**Velocity** en esta opción aplicaremos una velocidad lineal ya sea en X o Y. **Angular Velocity** se refiere a cambiar la velocidad angular.

**Max Speed** como se puede ver es una variable del tipo real, si deseamos aplicar

una velocidad máxima al actor cambiamos el valor de 0 a otro de mayor nivel, y tenemos la velocidad máxima que deberá

alcanzar el actor, ojo, la velocidad está implícita en todos los videojuegos y no precisamente en los de carreras.

**Apply Max Speed** si ponemos un valor en Max Speed debemos poner verdadero este valor para que tenga efecto.

**Physics** es la opción en donde le pondremos algo de física al juego, sus atributos son, Density, Friction, Bounciness, Fixed Rotation, Mov able, Collision Shape, Drag, Angular Drag.

**Density** se refiere a la cantidad de masa contenida en un cuerpo   **Friction** es la resistencia que se aplica el objeto ante otro cuerpo   **Bounciness** es la capacidad de elasticidad o rebote del cuerpo del actor.   **Fixed Rotation** se refiere a un giro o rotación arreglada.

**Movable** se puede entender como la capacidad del objeto para moverse.   **Collision Shape** el forma de colisión que tendrá la forma, existen dos

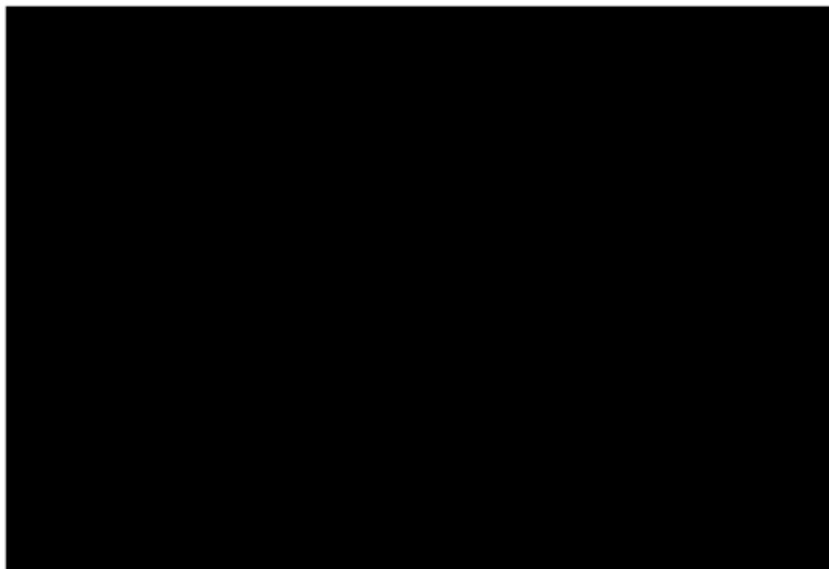
formas rectangular y circular.

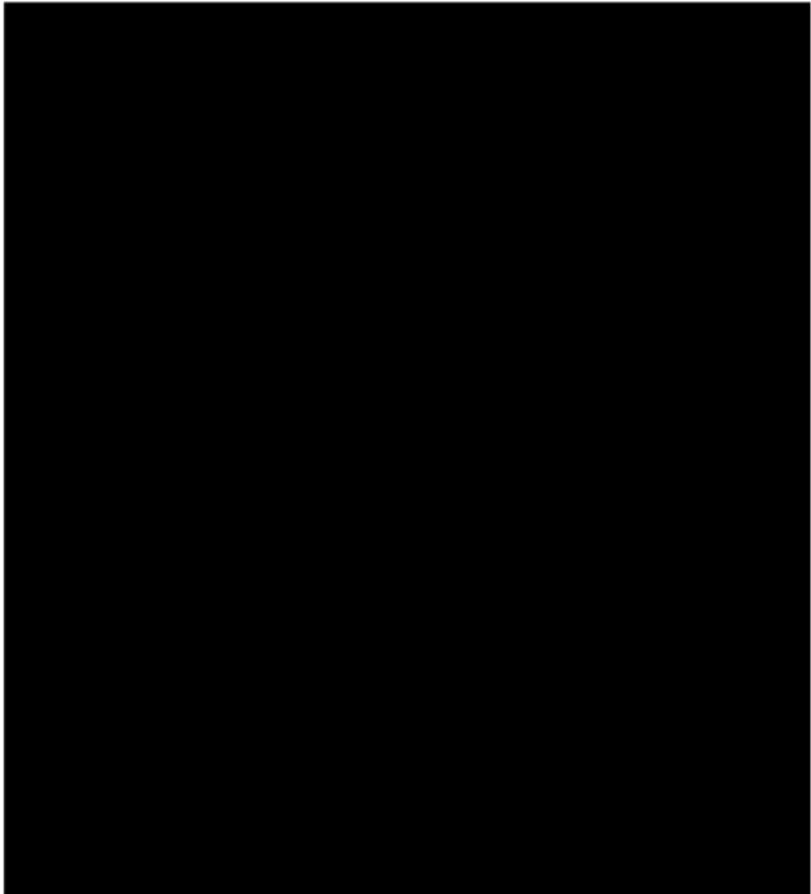
**Drag** es para poder tomar el objeto con alguna Behavior.

**Angular Drag** es para tomar el objeto en forma angular.

## Media

En la pestaña Media es donde agregaremos imágenes y audio, como podemos ver media también cuenta los mismos botones de +, -, lista y en forma de iconos, pero hay uno muy especial que tiene forma de carrito de compras y precisamente sirve para adquirir plantillas, imágenes, caracteres, etc. De la tienda de Gamesalad.

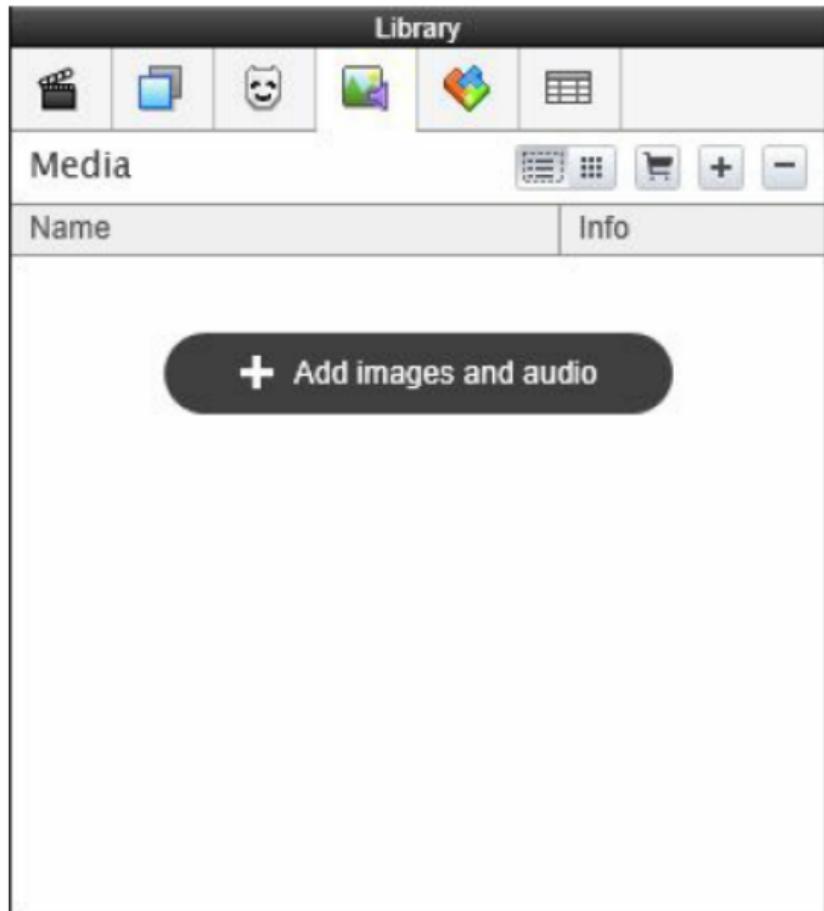






El botón con símbolo + adhiere nuevas imágenes o audio y el botón – elimina las imágenes o audio que se encuentren en la lista.

Estos dos botones son para mostrar las imágenes y audio en



**Botón de Carrito de compras que forma de lista o iconos.** sirve para adquirir, plantillas de juegos, players, imágenes, fondos,

iconos, etc. Desde la tienda de Gamesalad.

Para adherir nuevas imágenes o audio a esta pestaña tan solo tenemos que presionar en el botón con el símbolo +, nos aparecerá la siguiente ventana que se llama Select Media.



Cuando aparece esta ventana ya podemos adherir imágenes o audio a la lista de la pestaña media.

El tipo de imágenes que pueden adherir son:

Png formato grafico basado en un algoritmo de compresión sin perdida    Jpeg archivo con un algoritmo de perdida.

Gif formato grafico en la web tanto para imágenes o para animaciones.    Tiff formato de imagen por etiquetas.

Bmp formato de imagen que viene con Windows, puede guardar imágenes de 24

bits.

El tipo de audio que podemos adherir son:

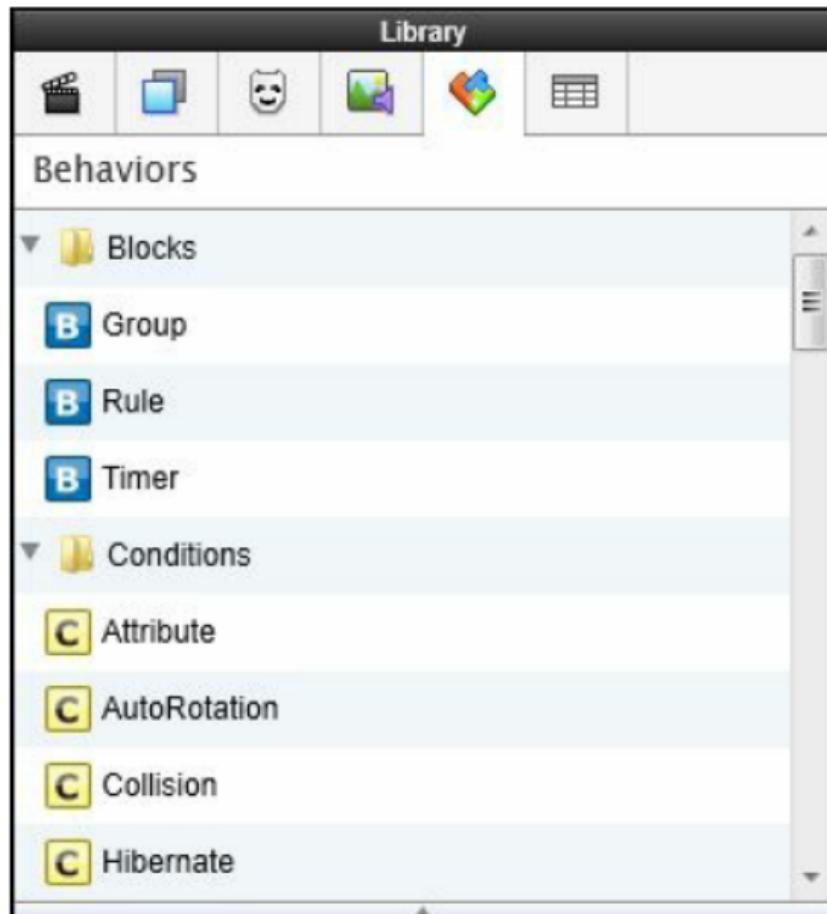
Ogg formato de multimedia que es usado como para audio o videos.

**Wav** formato de audio digital normalmente sin compresión de datos, desarrollado por Microsoft.

**M4a** archivo de contenidos audiovisuales, Apple empezó a usar la extensión de archivo .M4a para distinguir los archivos M4V.

## Behaviors

Los behaviors no son otra cosa que pequeños scripts en forma gráfica que nos ayudaran para ordenarle a gamesalad como queremos que se ejecute x acción. En los behaviors no tenemos botones como en las anteriores pestañas, en esta pestaña se encuentran divididos en tres conceptos Blocks que significa bloques, Conditions que es Condiciones y Behaviors que significa comportamientos. La pestaña Behaviors cuenta con 50 tipos de opciones.



A continuación la explicación de cada uno de los Behaviors (comportamientos).

1. **Group** (Block) Crea un nuevo grupo para contener varios behaviors y reglas.

2. **Rule** (Block) crea una condición o una serie de condiciones para verificar el jugador, entrada o cambio de atributo.
3. **Timer** (Block) permite ejecutar behaviors o reglas en determinados intervalos. Estos intervalos están definidos como: después de ciertos números de segundos, cada par de segundos, o para ciertos números de segundos.
4. **Attribute** (Condition) Un estado condicional que hace la función de apagador y encendedor de comportamientos dependiendo en el valor de una atributo.
5. **Autorotation** (Condition) una instrucción condicional que hace de activación de comportamientos dependientes de si o no un actor ha comenzado o terminado un auto-rotativo.
6. **Collision** (Condition) una instrucción condicional que hace de activación de comportamientos dependientes en donde un actor colisiona con otro actor.
7. **Hibernate** (Condition) una instrucción condicional que hace de activación de comportamientos dependientes en donde el sistema hiberna.
8. **Key** (Condition) una instrucción condicional que hace de activación de comportamientos dependientes al presionar una tecla.
9. **Mouse Button** (Condition) una instrucción condicional que hace de activación de comportamientos dependientes al presionar un botón de un Mouse.
10. **Mouse Position** (Condition) una instrucción condicional que hace de activación de comportamientos dependientes en la posición del mouse.
11. **Touch** (Condition) una instrucción condicional que hace de

activación de comportamientos dependientes en acciones basadas en toques.

12. **Accelerate** (Persistent) Especifica la velocidad y dirección (ángulo) de aceleración de un actor. May ormente utilizado en una regla que verifica un evento del teclado. Nota: si el comportamiento arrastrar no es aplicado a un actor, la aceleración se aplicara continuamente, aumentando mayor y mayor velocidad al actor hasta que alcance su máxima velocidad previamente definida, Puedes ver también el comportamiento **Accelerate Toward**.
13. **Accelerate Toward** (Persistent) Especifica de forma precisa la posición X, Y del actor. Y de igual forma especifica la aceleración de forma definida.
14. **Add/Remove Row** (Action) este comportamiento le permite al usuario seleccionar una tabla, adherir o remover filas desde la tabla seleccionada. Las acciones pueden ser dos Add Row/Remove Row, Beginning/at End, si un Index (Índice) es seleccionado entonces la expresión del campo puede ser mostrada la cual permite al usuario cual index (índice) utilizar.
15. **Animate** (Persistent) arrastrar y soltar una secuencia de imágenes que han sido importadas en el proyecto a este Behavior, Una vez adheridas, las imágenes también pueden ser reordenadas en una nueva. Se especifica de igual forma la velocidad de los Fotogramas en la animación, (Máximo 30 fotogramas por segundo), y si la animación es un bucle, se detiene al anterior fotograma, y/o regresa a la última imagen antes del comportamiento Animate haya iniciado.
16. **Change Attribute** (Action) Esta acción permite configurar, cambiar o incrementar el atributo de un actor, por

ejemplo, adherir 250 puntos al nivel de un juego en un atributo llamado score, eligiendo el incremento por 250 en el campo de incremento, o bien remover una vida del jugador cambiando en el nivel un atributo llamado vidas especificando -1 en el campo de incremento, o cambiar un color del actor.

17. **Change Image** (Action) Especifica una imagen para reemplazar la imagen actual en un actor. La imagen debe ser importada previamente en el proyecto del videojuego y después usar este comportamiento.
18. **Change Scene** (Action) Ir a la escena especificada en el juego desde una lista desplegable, bien puede ser pasar a un nuevo nivel, ir a una escena de créditos, o una escena de Game Over.
19. **Change Size** (Action) Hacer crecer un actor por un factor de escala (se hace pequeño con un numero negativo).
20. **Change Table Value** (Action) Cambia el valor de una celda en una tabla.
21. **Change Velocity** (Action) Especifica el movimiento en un particular ángulo relativo al actor o a la escena.
22. **Collide** (Persistent) Utiliza este comportamiento cuando deseas controlar que actor o grupos de actores pueden chocar o colisionar entre sí. Puedes especificar estos actores o un grupo de actores con una etiqueta.
23. **Constrain attribute** (Persistent) Attribute, este comportamiento atributo al de otro. Para una instancia.
24. **Control Camera** (Persistent) Adhiere este comportamiento a un actor y la cámara de la escena seguirá al actor para mantenerlo en la vista. El área de seguimiento

alrededor del actor es definida en el editor de escenas usando el modo Camera Edit. Notas: solamente un actor por escena puede tener este comportamiento adherido. Actors que no tengan activada la opción Scrollable en las capas no pueden usar este comportamiento.

25. **Copy Table** (Action) Copia los datos de una tabla a otra. Las tablas deben tener la misma disposición.
26. **Destroy** (Action) Inmediatamente remueve el actor de la escena. Mejor resulta cuando se usa con una regla.
27. **Display Text** (Persistent) Especifica el color, alineación, fuente, tamaño y texto (incluyendo juego, escenas, o atributos del actor, como un score o vidas) mostradas de un actor. El texto puede ser opcionalmente envuelto en los límites del actor. Mejor aun cuando el actor tiene su propio canal alfa puesto en 0 en sus atributos.
28. **Interpolate** (Persistent) Interpol a un atributo desde un valor a otro.
29. **Keyboard Input** (Action) Pon texto desde el teclado.
30. **Load Attribute** (Action) Carga el valor almacenado por una llave aleatoria desde una almacenamiento persistente en un atributo. Ver también la acción Save Attribute.
31. **Log Debugging Statement** (Action) Registra una declaración en la ventana de depuración.
32. **Move** (Persistent) Especifica el movimiento en un ángulo relativo al actor o a la escena. Ver también el comportamiento Move To.
33. **Move To** (Persistent) Especifica movimiento hacia una coordenada específica en X o Y relativo al actor o a la escena.

Al menos controlando condiciones que no son ya válidas y Correr hasta su terminación si no es seleccionada, una vez que el actor llega a las coordenadas específicas, este comportamiento se detiene. Ver también el comportamiento Move.

34. **Note** (Persistent) Escribe una nota para ti mismo o para otros creadores acerca del comportamiento, regla o grupo.
35. **Particles** (Persistent) Crea una explosión radial o fuente de partículas. Este comportamiento tiene la habilidad de colocar una imagen, RGB, color, tiempo de vida de las partículas, y otros parámetros.  
A diferencia del comportamiento Change continuamente actualiza los valores de un
36. **Pause Game** (Action) Elige una escena específica en el juego de una lista desplegable. El juego tomará esta escena como una pantalla de pausa, pausa la escena actual y muestra la pantalla de pausa por encima de la escena actual. Utiliza Des pausar juego para iniciar nuevamente.
37. **Pause Music** (Action) Esta acción pausa la música actual que se esté reproduciendo.
38. **Play Music** (Action) Esta acción activa un archivo de música para reproducir. Configura la música para reproducirse una vez o repetidamente en un bucle.
39. **Play Sound** (Action) Este comportamiento activa un archivo de sonido para reproducirlo. Configura el sonido para reproducirse una vez o repetidamente en bucle.
40. **Replicate** (Persistent) Usa este comportamiento para crear copias de un actor basado en una atributo o un entero. Por

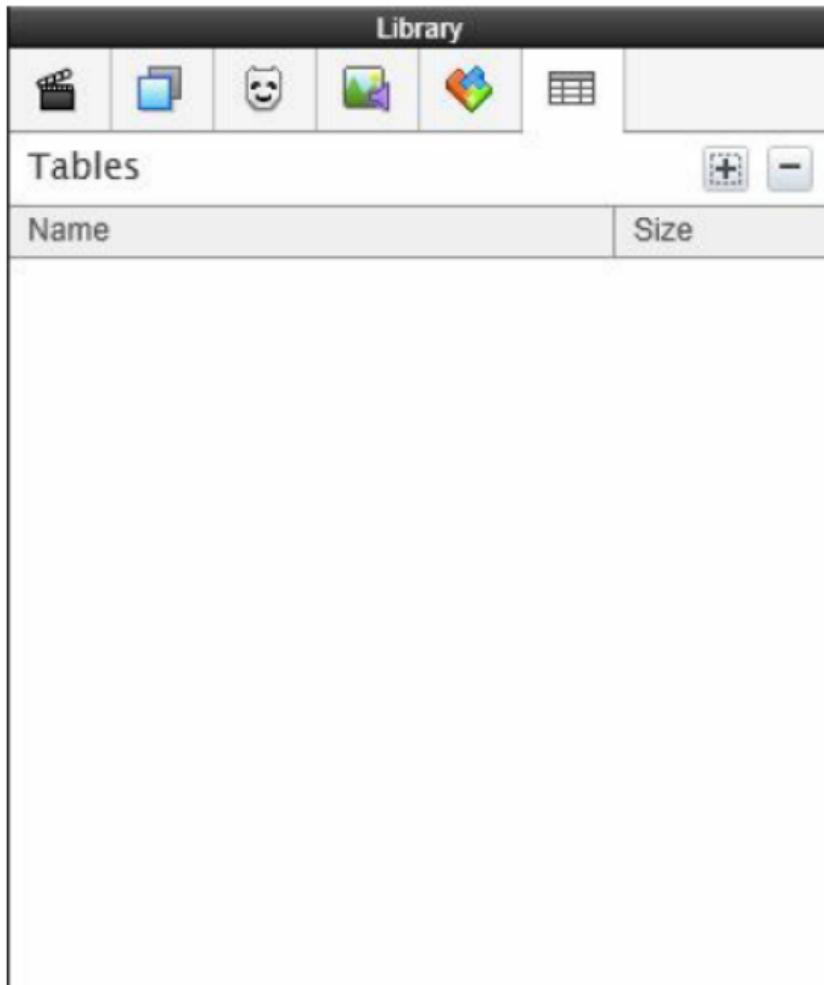
ejemplo, usa Replicate para mostrar el número de vidas de un jugador. Para hacer esto pon un actor en la escena con este comportamiento. Configúralo para repetirse basado en una variable llamada Vidas.

41. **Reset Game** (Action) Reinicia el juego y todas las escenas en este. Es mejor cuando lo usas con una regla.
42. **Reset Scene** (Action) Reinicia la escena actual y todos los actores en esta. Es mejor cuando se usa con una regla.
43. **Rotate** (Persistent) Usa este comportamiento para ti mismo para crear una constante del tipo manecillas del reloj de un actor. También puede ser usado en conjunción con una regla para rotar solamente cuando un evento ocurre, como cuando es presionado un botón desde el teclado de la computadora.
44. **Rotate to Angle** (Persistent) Usa este comportamiento para crear una rotación a un ángulo específico en un actor, relativo a cualquiera de los dos, a un actor o a una escena. También puede ser usada en conjunción con una regla para rotar solamente cuando un evento ocurre, como cuando el botón del teclado en la computadora es presionado.
45. **Rotate to Position** (Persistent) Usa este comportamiento para rotar específicamente en coordenadas X o Y, en la pantalla o relativamente al actor. También pueden ser usada en conjunción con una regla para rotar solamente cuando un evento ocurra, como cuando es presionado un botón en el teclado de la computadora.
46. **Save Attribute** (Action) Salva el valor de un atributo en almacenamiento persistente con una llave aleatoria. Ver también acción Load Attribute.

47. **Save Table** (Action) Salva los datos de una Tabla.
48. **Spawn Actor** (Action) Bueno para disparar proyectiles, soltar items, o poner huevos, este comportamiento crea la instancia de un nuevo actor en la escena.
49. **Stop Music** (Action) Esta acción detiene la música que se esté reproduciendo.
50. **Unpause Game** (Action) Si el juego está usando una escena como una pantalla de Pausa, este comportamiento remueve la pantalla de pausa, restaura el juego a la escena previa, y el juego se empieza a reproducir.  
Como puedes ver son 50 comportamientos y lo que te recomiendo es que practiques con cada uno para que veas cómo funcionan y qué comportamiento realizan dentro del juego.

## Tables

La última pestaña es Tables o tablas, esta pestaña nos servirá para adherir datos en una tabla para el juego en cuestión y luce de la siguiente forma.



Para Adherir una Tabla presionamos en el Botón + y para eliminarla el botón con el símbolo -, al crear una tabla nos aparece lo siguiente.

The screenshot shows the Microsoft Word ribbon with the 'Library' tab selected. Below the ribbon is a table structure. The table has one row and one column, labeled 'Table 1' and '1x1' respectively. There are also '+' and '-' buttons for managing the table.

Name	Size
Table 1	1x1

Vemos que tenemos una Tabla con el nombre "Table 1" y con un tamaño de 1x1, y bien ahora te preguntaras, ya tengo la tabla y ahora como la administro. Bueno si mueves tu vista hacia la parte Derecha inferior de tu pantalla veras lo siguiente.

The screenshot shows the Microsoft Word backstage view with the 'Table Tools' ribbon selected. The table is identified as 'Table 1'. The 'Rows' field shows '1' and the 'Columns' field shows '1'. A single cell containing the word 'Text' is visible.

En esta parte es en donde administraremos la tabla recién Creada, puedes Ver el nombre de la Tabla en la parte superior izquierda que es "Table 1", de igual forma dos campos con un numero 1 en el lado izquierdo, en medio "Rows" y a la derecha otro campo con valor 1, y después de este un texto que Dice "Columns" en esto valores es en donde especificamos cuantas Filas queremos en el valor Rows y Cuantas Columnas en el valor Columns.

En la parte Derecha de esta misma ventana, podemos apreciar

tres botones mas.

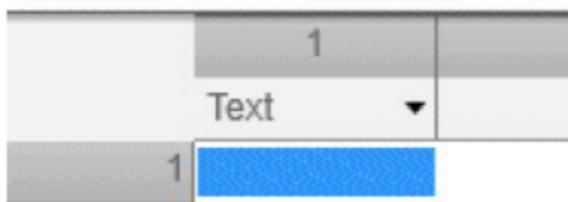


Si presionamos en el primero de estos tres botones (De izquierda a Derecha) vemos lo siguiente.



Import CSV y Export CSV, estas opciones nos sirven para importar o exportar datos de formato abierto, que sirve para representar datos en forma de tabla. En las que las columnas se separan por comas y las filas por saltos de línea.

Para el siguiente botón debemos primero seleccionar la columna para poder habilitarlo, como se muestra en la imagen.



Una vez que este seleccionado el campo, entonces las opciones del botón “+ desplegable” del lado derecho de esta opción, habilitara todas las opciones. Como se muestra en la imagen.



En donde Insert Row Above significa “Insertar una Fila arriba”, Insert Row Below “Insertar una Fila Debajo”, Insert Column Before “Insertar una Columna Antes”, Insert Column After “Insertar Columna Despues”.

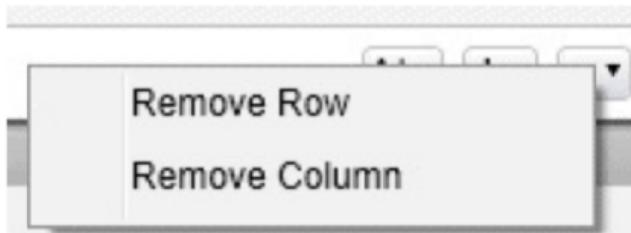
Vamos a presionar la opción Inserte Row Above y se colocara una Fila por encima de la celda seleccionada. Y como puedes ver ahora contamos con dos Filas, como se muestra en la imagen a continuación.

The screenshot shows a user interface for creating a table. At the top, there are two input fields: 'Rows' set to 2 and 'Columns' set to 1. Below these are two dropdown menus with up and down arrows. A table structure is displayed with three rows. The first row has a single cell containing the number 1. The second row has two cells: the first cell contains the word 'Text' and the second cell is highlighted with a blue dashed border. The third row has two cells, both containing the number 1. The entire table is contained within a light gray box.

	1	
Text		
1		

Como puedes ver ahora tenemos 2 Rows y 1 Columns, esto es lo que hemos hecho, si sabes usar Excel, no encontraras mayor complicaciones al crear una tabla.

El botón con el Símbolo -, nos sirve para remover filas o columnas.



Lo que Debes aprender para almacenar datos correctamente en una tabla, es identificar el tipo de datos que deseas almacenar, las tablas en Gamesalad nos permiten almacenar los siguientes tipos de datos.

Table 1

Rows 1 Columns

	1	
	Text	
1	Text	
2	Integer	

Text que

es Texto, Integer que son Enteros, Real es números Reales, Boolean que es booleanos (Verdadero o Falso), Angle que es Angulo.

Ojo, no puedes almacenar por ejemplo un valor de Dato de Texto en un valor Integer o Real, mas sin embargo si puedes almacenar valor numéricos en uno de texto. Lo ideal es que aprendas estos valores para saber cómo usar tablas. Te comento que este es un procedimiento estándar, que bien puede ser así en Gamesalad, pero de igual forma puedes aprenderlo por ejemplo en el aprendizaje de Access para Crear Tablas. Pero de igual forma no te preocupes que en este libro aprenderemos a crear las tablas, y lo verás en capítulos más adelante.

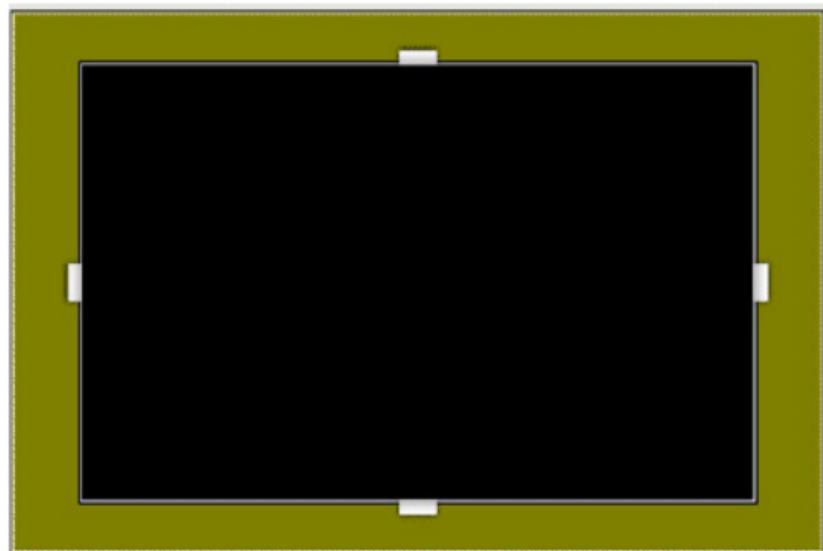
## Stage

El stage es en donde tenemos la escena de gamesalad, y es en donde gráficamente administraremos el juego, en Stage en la parte superior tenemos esta barra de opciones.



El botón con la Flecha del mouse nos permitirá elegir los actores u objetos que hayamos colocado en la escena.

Si presionamos el botón con el ícono de la Cámara en la Escena se muestra así.



Nos sirve para mostrar el ancho y el alto de la cámara, los cuadritos que se aprecian en los bordes de los medios, sirven para ampliar o reducir el aspecto de la cámara.

El siguiente botón del ícono Verde



nos sirve para reproducir el juego y para ver una previa de cómo va quedando, por el momento no lo usaremos hasta ir viendo cada uno de los ejemplos.

El botón Continuo al botón verde



nos sirve para mostrar u ocultar el backstage, que aparece en la parte inferior. Como se muestra en la imagen.



Finalmente Si nos damos cuenta en la parte Derecha de la barra contamos con una serie de opciones como X, Y, W, H y R.



Pues bien si seleccionamos un actor u objeto de la pantalla Scene, nos mostrara la Posición X o Y en donde se encuentra, Así como en W su Anchura, y en H su altura, en R su Rotación.

Vamos a hacer una prueba, vamos a ubicarnos en la pestaña Actors

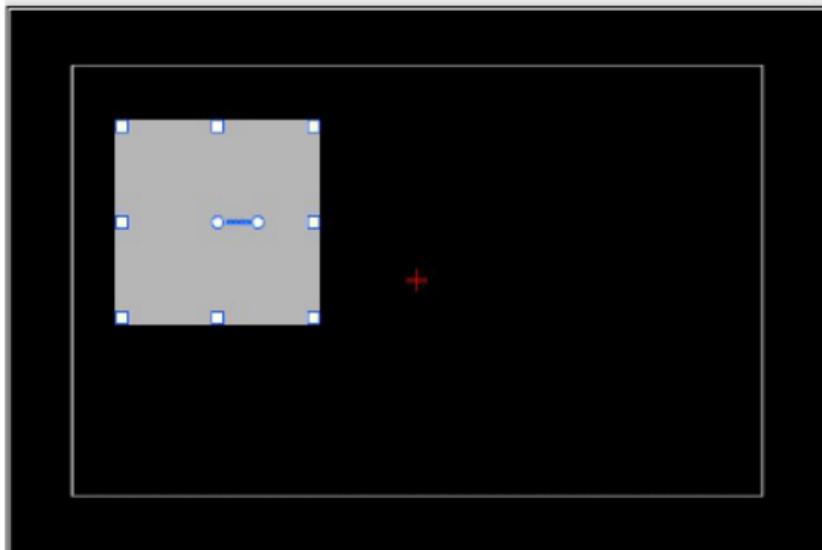


y creamos un nuevo actor.

Nos aparecerá el nuevo actor en esta forma.

The screenshot shows a software interface titled 'Actors'. A dropdown menu is open, showing 'All' selected. Below it, there is a list with one item: 'Actor 1'. To the right of the list are two small icons: a grid icon and a list icon.

Seleccionamos el Actor y lo arrastramos a esta ventana sin soltarlo hasta donde quede posicionado en el espacio negro.



Si nos vamos nuevamente a la barra para ver las propiedades

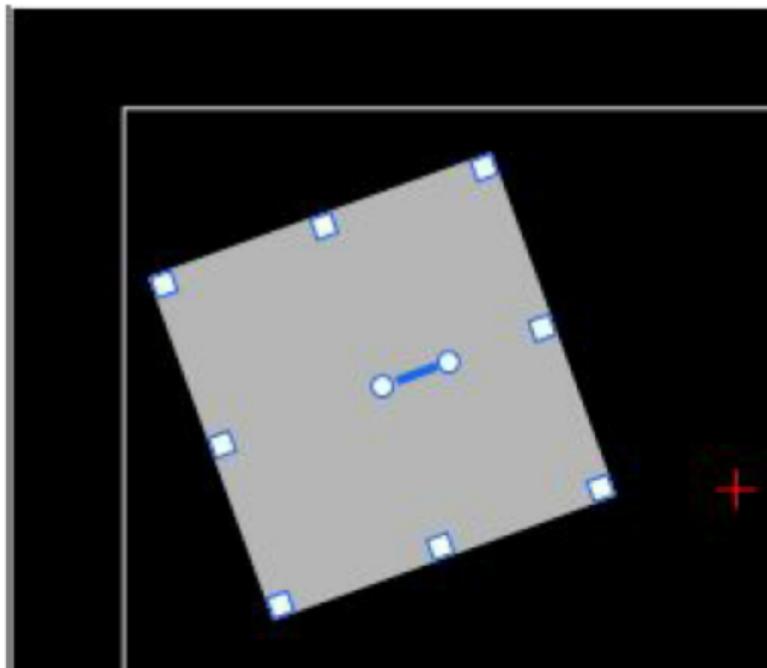
del actor, veremos que ya cuenta con valores y nos especifica su ubicación en X, Y. Su Anchura y Altura en W, H. y Su Rotación en R.

Actor 1 X: 124 Y: 194 W: 120 H: 120 R: 0

De igual forma nos muestra el nombre del Actor, en este caso no cuenta con Rotación, vamos a adherir un valor en el Campo R, le vamos a poner 20 como valor en el campo.

R: 20

Y vemos como el Actor con forma de Cuadro, Rota en la Escena.



esto finalizamos esta etapa para Conocer las opciones que nos ofrece Gamesalad, no sin antes mostrarte Finalmente una

opción muy importante que es la de  Game que se encuentra en Attributes.

Attributes		
 Game	 Scene	 Actor
Game Attributes		 
Name	default name	text
Time	0	real
▶ Display Size		Size
Actor Tags		text

Esta opción cuenta con los Atributos de juego por defecto de Name que tiene un valor de Text, Time que es un número Real, Display Size con un Valor de Size y Actor Tags que es un valor de Text.

En esta parte la prestación más importante es Display Size que despliega lo siguiente.

▼ Display Size	Size
Width	480
Height	320

El ancho que es Width y el Height que la altura, ambos son valores del Tipo Real, en esta opción podemos especificar el ancho y la altura que va a tener el juego que estemos desarrollando. Y es como se mostrara finalmente en el dispositivo en donde se ejecute el juego, ya sea en teléfonos iPhone o Android. Por eso resulta muy importante esta opción.

Conociendo ya todas las opciones con las que cuenta Gamesalad, vamos a dar paso a la Creación de Juegos con ejemplos de mover personajes, colisionar, destruir, rotar, lanzar, acelerar, seguimiento de personajes con la cámara y la creación de un juego de plataforma.

## Mover Actores

Ya hemos visto cómo funcionan las opciones de gamesalad, ahora vamos a ir aprendiendo las técnicas para crear un juego, vamos a mover los actores en Gamesalad. Lo primero que debemos hacer es adherir una imagen para nuestro actor, he utilizado el siguiente avión que he creado con photoshop.



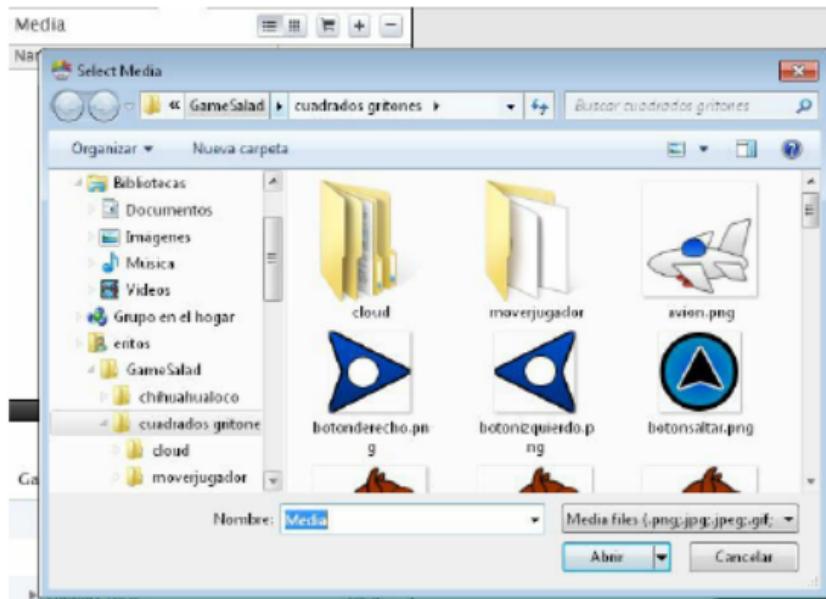
La imagen en PNG la adhiero presionando en la pestaña Media



presionamos en el botón



y nos aparece la siguiente ventana

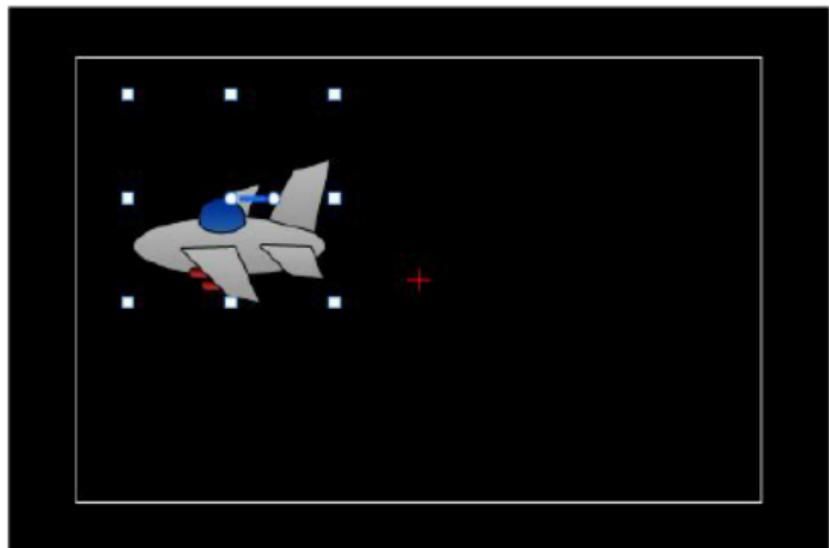


En esta ventana elijo el archivo PNG con nombre avión y este se coloca en la pestaña media.

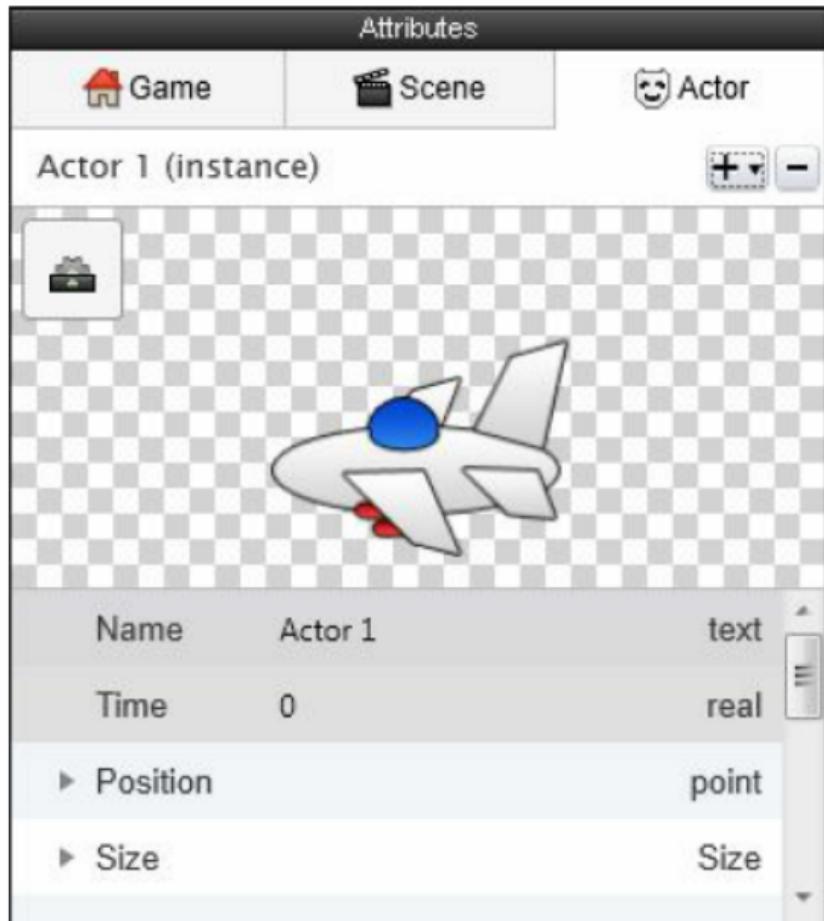
Name	Info
avion	128x128

Como puedes ver la imagen tiene un tamaño de 128x128 pixeles, recuerda que las imágenes siempre las vamos a crear al poder de 2, por ejemplo si tenemos una imagen de 8x8 está para hacerla más grande (escalarla) tenemos que multiplicarla por 2 y debe quedar de 16x16 y si el 16 lo multiplicamos por 2 nos queda la imagen de 32x32 y así sucesivamente.

Ok para colocar nuestro actor en la escena tan solo arrastramos la imagen hacia la escena y nos quedaría de esta forma.



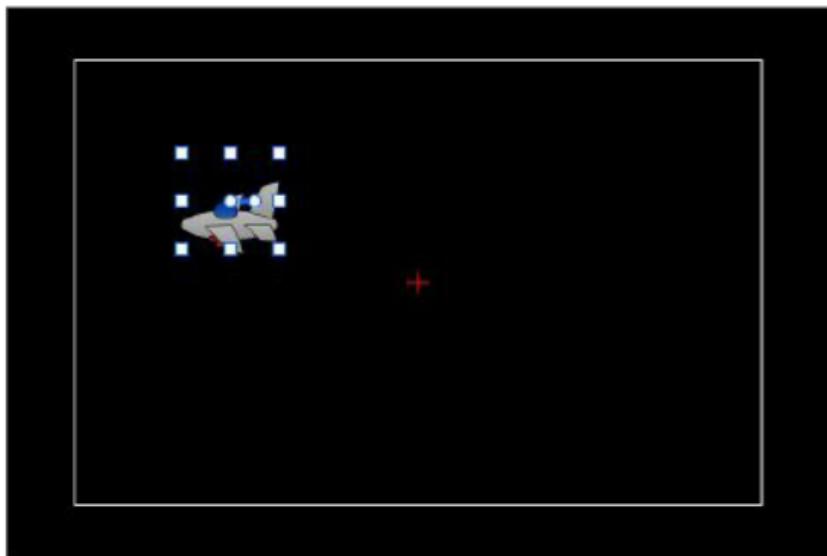
Vemos que nuestra imagen queda muy grande, la voy a hacer más pequeña, nos vamos a la opción de actor en Attributes



En donde tenemos la opción de Size, presionamos en la opción y se despliega la siguiente ventana.

▼ Size		Size
Width	128	real
Height	128	real
Rotation	0	angle

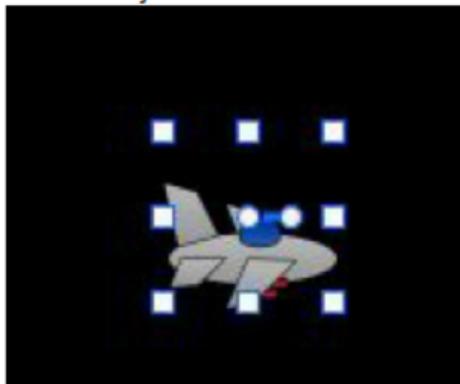
Y cambiamos el Width haciendo doble clic en la opción para que se habilite la edición del campo y ponemos 64, hacemos lo mismo con la Height. La imagen se escala para ser más pequeña.



Ahora el avión se ve muy bien, bueno pero como no quiero que el avión se mueva en forma horizontal de la derecha hacia la izquierda, sino de izquierda hacia derecha, entonces vamos la opción del Actor Flip Horizontally que significa Voltear Horizontalmente, el avión queda correctamente viendo hacia el lado derecho.

Horizontal Anchor	Left	enumeration
Vertical Anchor	Bottom	enumeration
Flip Horizontally	<input checked="" type="checkbox"/>	boolean
Flip Vertically	<input type="checkbox"/>	boolean

Así se vería el avión en la escena. Con la opción Flip Horizontally seleccionada.



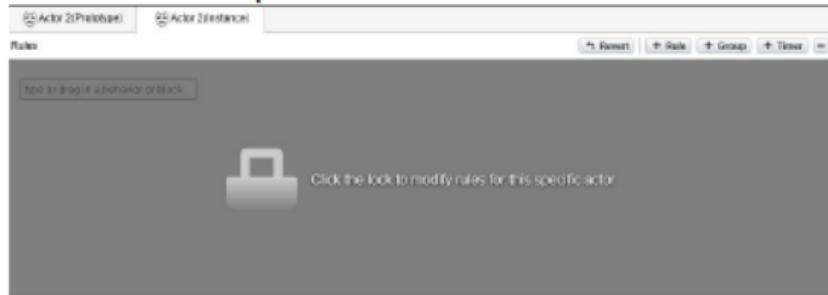
Ahora voy a adherir los Behaviors para mover el avión de arriba hacia abajo. Vamos a la pestaña Behaviors



con el Actor seleccionado en este caso el Avión, presionamos el botón



para que se despliegue la ventana Backstage, nos aparece esta ventana en la parte inferior.



Presionamos en el candado para habilitar y debe aparecer lo siguiente.



Vemos que hay un pequeño campo que dice "type or drag in a behavior or block" nos pide que tipemos o arrastremos dentro del campo un comportamiento o bloque. Y es lo que vamos a hacer a continuación.

Regresamos a la ventana behavior y arrastramos el Comportamiento Key, que luce así.



# Key

Al arrastrarlo hacia el backstage la ventana luce de la siguiente forma.

The screenshot shows the RoboRealm software interface with the 'Rules' tab selected. A rule is being configured with the following parameters:

- When:** **All** of the following are happening
- receives a key press:** **is: down**
- else:** (empty)

Tenemos una opción donde dice que es una Rule (Regla) y con una serie de opciones como:

When **All** of the following are happening

Esto quiere decir Cuando Todo lo siguiente esté ocurriendo, si presionamos en el botón de color Azul que dice

**All**

se despliegan una serie dos opciones All y Any, en donde All

significa Todo y Any significa Cualquiera.

Por ejemplo si seleccionáramos Any, la sentencia quedaría

When Any of the Following are happening (Cuando cualquiera de lo siguiente ocurra) por el momento vamos a seleccionar la opción ALL.

La siguiente sentencia es  
receives a key press:

**is: down**

y significa reciba

a presión de tecla, el campo se encuentra vacío, pero si colocamos el puntero dentro del campo y presionamos las teclas de direccionamiento o cualquier tecla en la computadora podemos ver que cambia, si presiono la tecla hacia arriba en mi computadora el campo

dice lo siguiente

up

que significa Arriba, si presión la tecla de la derecha dirá,

right

y así cualquier tecla que presiones en el campo se colocara la opción en este. Por el momento la voy a dejar en la tecla Up.

La siguiente opción nos muestra un campo que dice lo siguiente

type or drag in a condition

el cual nos sirve para colocar condiciones, en este momento no es necesaria alguna condición.

Las siguientes son las opciones Do y Else, como puedes ver se trata de un Control de programa para ser más exacto de un Si una condición es verdadera hacer lo siguiente, si no es verdadera hacer esto.

En un programa normal sería así.

*Si (esto es verdadero) entonces Ejecutar la sentencia*

*Else (por lo contrario)*

*Hacer esto otro*

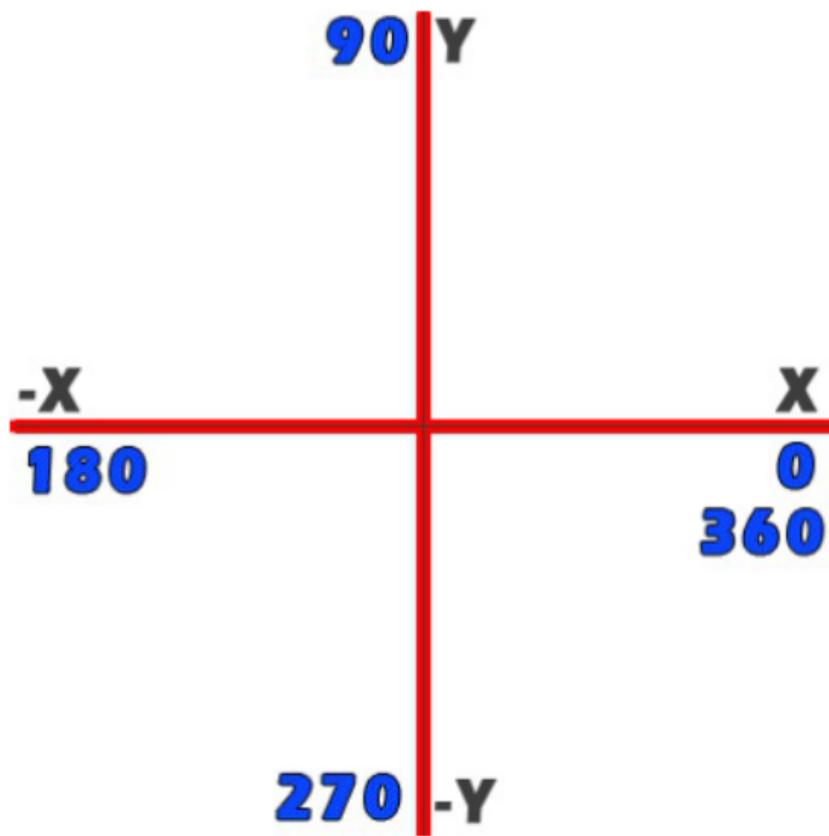
Como lo que quiero es que se mueva hacia arriba Coloco en do, otro comportamiento que le indique a mi actor que se mueva hacia arriba. Entonces arrastro el behavior move. El cual luce así.



En donde move in direction: colocamos 90 y at a Speed of: colocamos 200, las opciones relative to: actor, in a(n): additive way las dejamos como están.

Te explico brevemente, como estamos trabajando con 2d, pues estamos hablando de un plano cartesiano, el plano cartesiano se divide en Cuadrantes. Sobre los ejes X, Y. así bien que si es importante saber esto, para hacer un juego en dos dimensiones, X es moverse o colocarse horizontalmente, Y es moverse o colocarse Verticalmente.

Aquí te coloco un gráfico para que lo entiendas mejor.



Cuando partimos de X positivo el valor es Cero o sea que si queremos movernos hacia la derecha debemos poner 0 en el campo de Move in direction, si queremos ir hacia arriba ponemos 90 y este se encuentra en el eje Positivo Y, ahora si quisiéramos ir hacia la izquierda entonces el valor en el campo Move in direction debe ser 180, y si queremos ir hacia abajo usamos el valor de Y negativa que es 270, ¿pero porque

aparece ese 360? Bueno en un plano cartesiano como puedes ver vamos de 90 en 90 y como son cuatro cuadrantes entonces  $90 \times 4$  es igual a 360, pero si colocamos 360 en el campo de Move in direction 360 es como si colocáramos el Cero, nuestro objeto o Actor se moverá hacia la derecha, así que siempre usa Cero para moverte a la derecha.

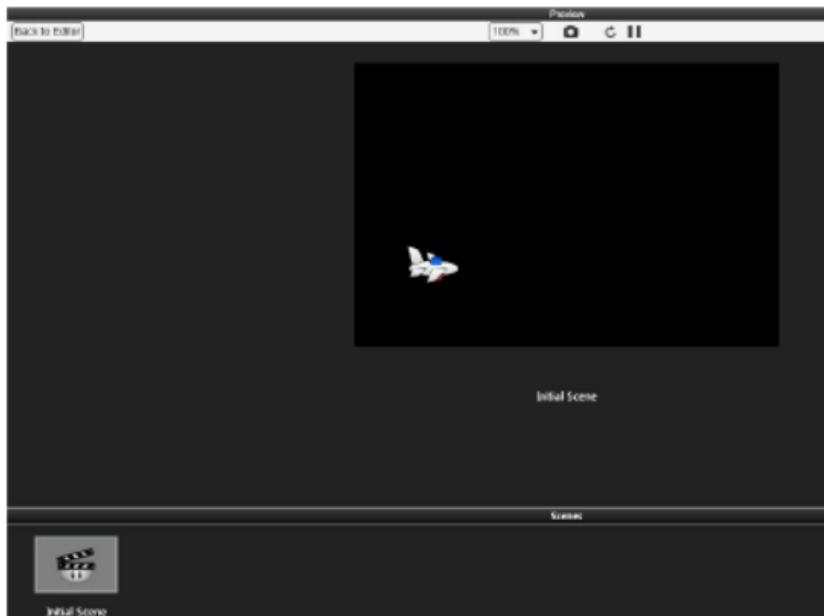
Bueno ya entendido esto, los valores de move quedan así.



En la opción Else no ponemos nada por el momento.

**else** Vamos a ejecutar el juego, en la opción

Stage, Nos aparece la siguiente ventana que ya es el juego en sí.



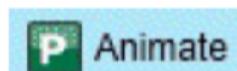
Si presionamos la Tecla Arriba de nuestro teclado en la computadora, veremos que el Actor Avión se mueve hacia arriba con una velocidad considerable, y si el botón es soltado el avión deja de moverse.

**Tarea:** Haz que tu actor se mueva hacia la derecha, izquierda, arriba y abajo, usando las teclas W para ir hacia Arriba, la tecla S para ir hacia Abajo, la Tecla A para ir hacia la Izquierda y finalmente la D para ir hacia la Derecha.

# Animar Actores

Ya tenemos el actor avión en el juego, pero ahora quiero que este avión tenga una animación, normalmente sería con un archivo .GIF pero si lo queremos hacer directamente incluyendo un GIF animado Gamesalad no lo reproduce, para hacer esta tarea Gamesalad incluye un Behavior llamado Animate, este behavior nos permite incluir fotograma por fotograma para controlar la animación en gamesalad. Dejamos el Behavior que incluimos en Mover Actores, en donde dice Do arrastramos el Behavior Animate que

lucen así.



Una vez que se ha arrastrado el Behavior anímate en el Do de la Regla para moverse debe lucir así.



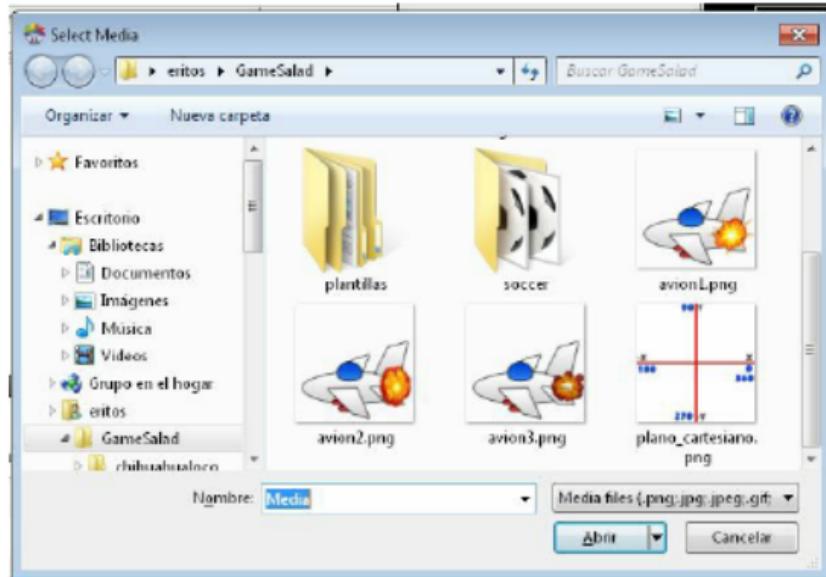
En el cuadro grande vamos a ir incluyendo fotograma por

fotograma, lo que debemos hacer es importar una serie de imágenes de una animación, he realizado las mías en archivo png, utilizando Photoshop.

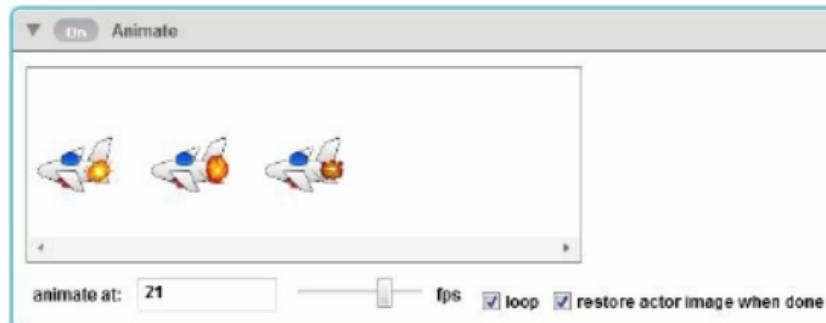
Vamos a la pestaña Media



importamos las secuencias de imágenes



Las imágenes son una representación de una explosión en la parte trasera del avión, arrastraremos imagen por imagen al cuadro del Behavior Animate, y debe quedar de la siguiente forma.



Ahora si ejecutamos el juego



podemos ver que cada vez que se presiona la tecla Arriba en la computadora, la animación de la explosión se reproduce por secuencias.



Te explico que puedes aumentar la velocidad de la animación en la opción anímate at: yo por mi parte he puesto una velocidad de 21, pero esto dependerá de tu gusto para ver cómo se reproduce la animación, tenemos dos casillas de verificación, loop que es un bucle y se repetirá hasta que el botón arriba se haya soltado, y restore actor image when done, en donde se reinicia la imagen del actor que tenemos por defecto una vez que se haya realizado la tarea.

# Hacer un Fondo para el juego

Una parte importante de un juego es como va a lucir ante la vista del jugador, y para esto tenemos que crear fondos, por defecto Gamesalad pone un fondo de color Negro, pero esto podemos configurarlo, vamos a Scene en attributes



Scene

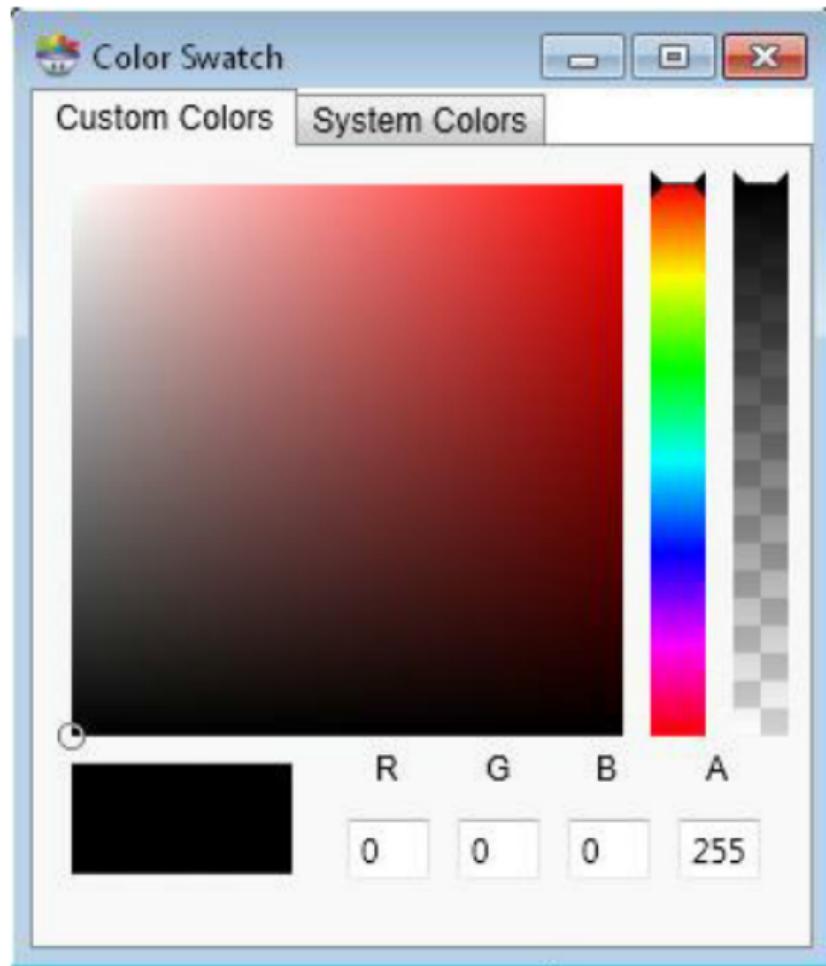
y nos aparecen una serie de opciones que ya hemos comentado anteriormente, nos vamos a la opción en dice color

► Color

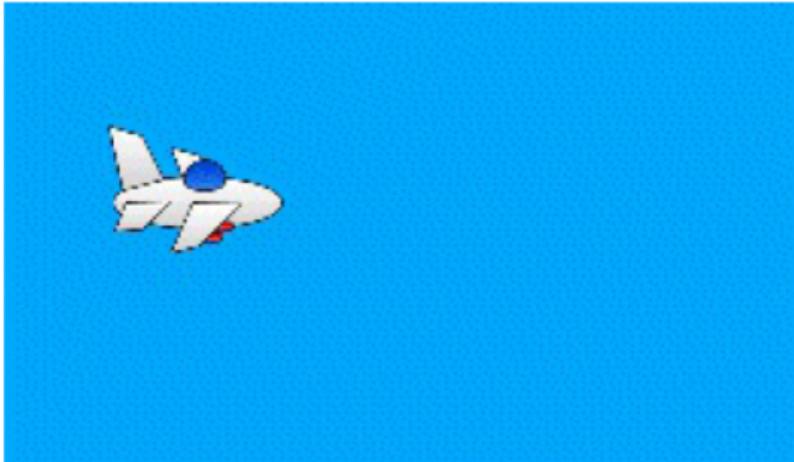


Color

y presionamos en el rectángulo de color negro. Nos deberá aparecer la siguiente pantalla.



Elegimos un color Azul para el fondo, para dar más realismo, pues se trata de un avión y este se mueve solamente el cielo. El juego debe lucir como se muestra en la imagen.



Hasta ese momento todo parece ir bien, mas sin embargo aún faltan elementos en el juego, como nubes. Vamos a adherir las nubes desde una imagen. Vamos nuevamente a Media



importamos las nubes como un fondo en una imagen PNG, la imagen tiene un

tamaño de 1024x512 y al incluirla en la escena la escalo para que ajuste correctamente, la escena luce de la siguiente forma.



El avión extrañamente luce opaco, esto se debe a que la capa de la imagen Nubes ha quedado por encima del avión, y al ser una transparencia este luce de esta forma, para corregirlo en el editor de escena presiono clic secundario del mouse para que despliegue lo siguiente.

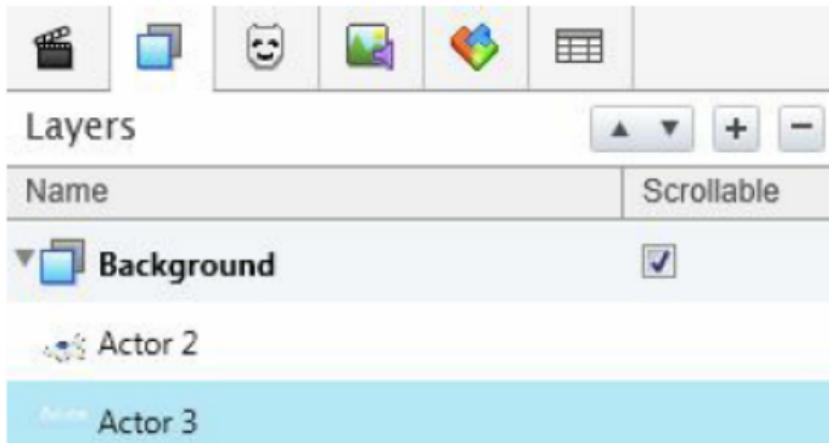


Elijo la opción Send to Back y la imagen se enviará hasta el fondo. Y ahora el avión luce correctamente.



## Vamos a Layers

 podemos ver que existe una capa que se llama Background y ella tiene a los dos Actores del juego el avión y las nubes.



The screenshot shows the 'Layers' panel in Construct 2. At the top, there is a toolbar with icons for file operations (New, Open, Save, etc.) and a scrollable list of layers. Below the toolbar, the word 'Layers' is displayed. The main area is a table with two columns: 'Name' and 'Scrollable'. The 'Background' layer is listed, with its name highlighted in blue and a checked checkbox in the 'Scrollable' column. Below this, 'Actor 2' and 'Actor 3' are listed, with 'Actor 3' currently selected, indicated by a blue background.

Name	Scrollable
Background	<input checked="" type="checkbox"/>
Actor 2	
Actor 3	

Ojo y mucho ojo no es bueno tener a los mismos actores en una sola capa, lo ideal es crear una por diferentes actores.  
Presionemos en el botón con el símbolo



y se creara una nueva capa, con el nombre Layer1 y con la opción scrollable activada la dejamos tal cual.

Name	Scrollable
Background	<input checked="" type="checkbox"/>
Layers	
Name	Scrollable
Background	<input checked="" type="checkbox"/>
Actor 2	
Actor 3	
Layer 1	<input checked="" type="checkbox"/>
Nubes Fondo	<input checked="" type="checkbox"/>

Arrastramos el Actor 3 que en realidad son las nubes hacia la Layer con el Nombre **Nubes Fondo**. Y ahora si los actores nubes los voy a colocar siempre en esta capa. Y como puedes ver la Layer ya tiene una pestañita para hacer retráctil o expandible la opción.

## Layers



Name	Scrollable
Background	<input checked="" type="checkbox"/>
Actor 2	
Nubes Fondo	<input checked="" type="checkbox"/>
Actor 3	

Si observamos detalladamente en Layers existen dos botones



que sirven para ir

hacia arriba o hacia abajo, esto es para colocar las layers, encima o por debajo de otras, experimenta con ellas para que veas su funcionamiento. Efectuan el mismo procedimiento de colocar la imagen hasta el fondo, atrás o adelante que previamente vimos cuando pusimos el fondo en el juego.

# Colocar una Cámara

Seguramente ya te has dado cuenta que el actor de un juego va caminando y el fondo se mueve y van apareciendo nuevos enemigos según el movimiento de pantalla, bueno en Gamesalad esto es fácil de realizar con el Behavior Control Camera.



Retraemos las reglas presionando en la el botón de la flechita

que se encuentra en la siguiente forma.

Ok ahora vamos a elegir el Behavior Control Camera

**P Control Camera** y lo arrastramos sin tocar ninguna de las reglas que ya has creado.



Y Eso sería todo, ahora si reprodujimos el juego el Behavior Control camera ya debe estar funcionando y seguirá al actor hacia donde este se mueva.

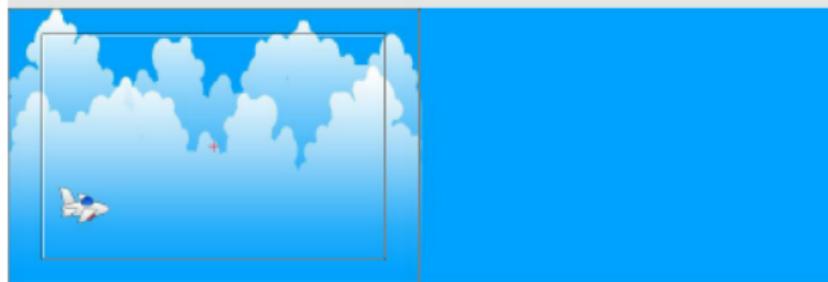
Seguramente no estás viendo que la cámara siga al jugador ¿verdad? Mas sin embargo efectivamente lo está haciendo, lo que sucede es lo siguiente como tu Escena tiene el ancho y el alto predeterminado, entonces ese es el tamaño del lente de la cámara, lo que debemos hacer es lo siguiente.

Vamos Scene

**Scene** y desplegamos la opción Size.

▼ Size	Size	
Width	480	real
Height	320	real

En donde dice Width ponemos el doble de 480 que seria 960, ahora podemos ver en la ventana del Stage que la escena se ha ampliado.

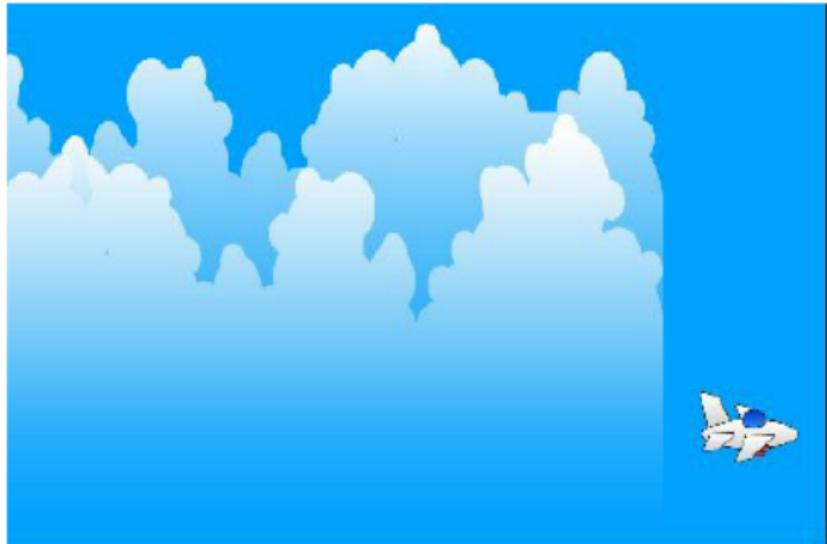


Ok ahora vamos a reproducir nuevamente el juego



y ahora si el actor se mueve hacia la derecha y la cámara sigue al actor.

Y encontramos otro pequeño problemita, si el actor llega hasta el borde de la pantalla en el lado derecho entonces la cámara se mueve, y esto realmente luce mal.

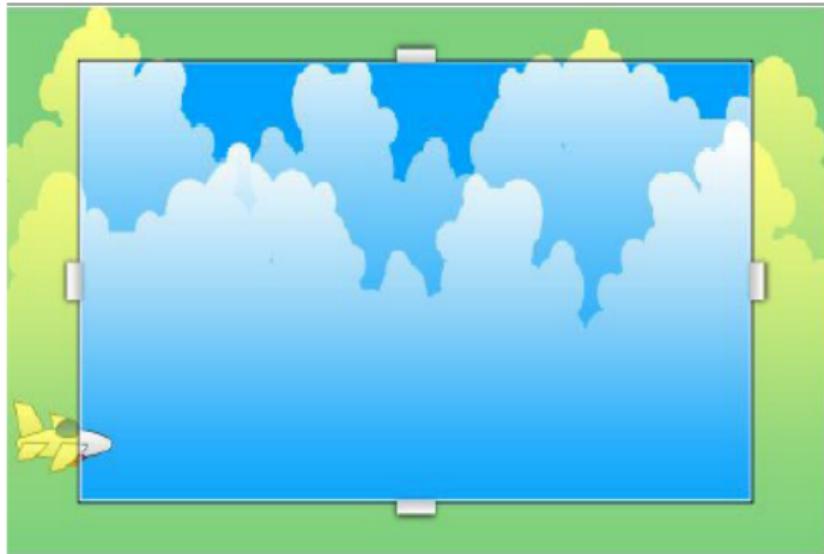


Para Arreglar este pequeño inconveniente vamos a presionar en

el icono de cámara  que se encuentra en la barra del tope.



Al presionar nos aparece un pequeño rectángulo en la escena de color amarillo en sus bordes externos.



Para que la cámara empiece a seguir al avión al llegar al medio de la escena, entonces debemos ajustar este rectángulo para hacerlo más pequeño, imagina que es una mira telescópica, entre más pequeña se supone que tiene más probabilidad de ser más exacto o exacta, lo mismo ocurre con Gamesalad. La voy a dejar así.



Ahora si reproduzco el juego, el avión se va a mover acorde al medio y se verá muy bien el efecto de seguimiento de cámara.



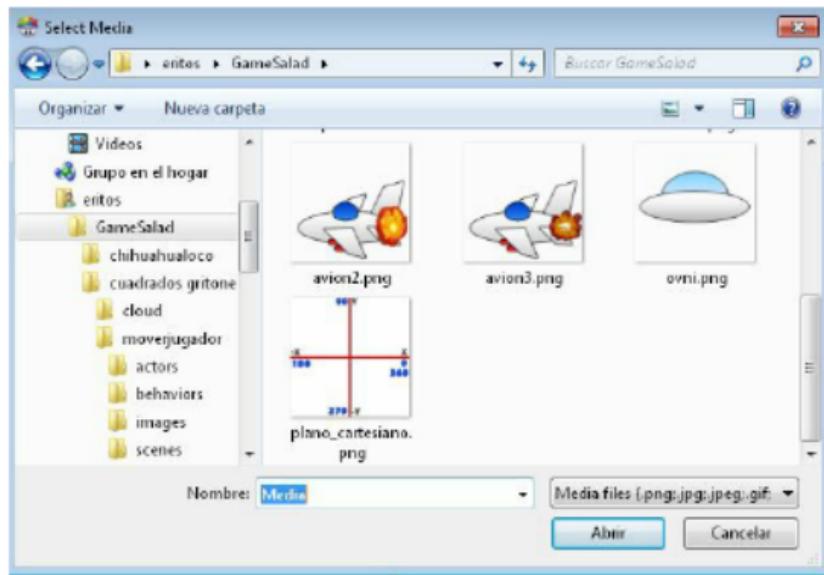
Ya luce correctamente, aunque en este momento no interesa mucho que el fondo de las nubes se mueva acorde a la cámara, para hacer un bucle del efecto de las nubes lo veremos más adelante.

# Colisionar y Rebotar Actores

Una parte importante de un juego es cuando los actores colisionan, porque permite definir qué hacer con la colisión, si se trata de un suelo, el actor caminara o rebotara sobre él, si fuera un enemigo los dos se destruirán rebotando y darán paso a otro evento, es por eso que resulta muy importante aprender a crear las colisiones de rebote, y en este momento nos vamos a centrar en este aspecto.

Voy a importar una imagen de un ovni a media





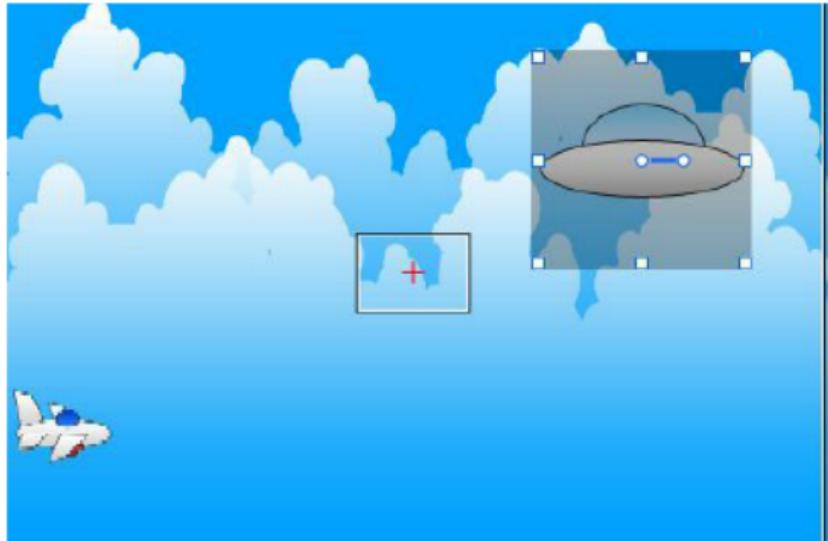
Y debe de quedar así, en la lista de imágenes en Media.

## Media



Name	Info
avion	128x128
avion1	128x128
avion2	128x128
avion3	128x128
blue-clouds	1024x512
ovni	128x128

Arrastro la imagen de Ovni Al Stage o Escena para crear el actor.



Como el actor Ovni ha quedado muy grande, le voy a modificar su Anchura y Altura a 64 la pestaña Actor

### Actor

▼ Size	Size
Width	64
Height	real

Ok ahora vamos a lo bueno, en el Backstage del Actor Avion vamos a adherir un Behavior que se llama Collide

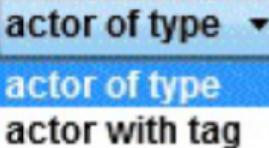
### Collide

Lo arrastramos hacia Rules y nos aparece lo

siguiente.

The screenshot shows the Construct 2 rule editor interface. At the top, there are two tabs: 'Actor 4(Prototype)' and 'Actor 4(instance)'. Below the tabs, there's a 'Rules' section with a 'Revert' button and buttons for '+ Rule', '+ Group', '+ Timer', and '-'. A large central window is titled 'Collide'. It contains a dropdown menu labeled 'bounce when colliding with:' with three options: 'actor of type', 'actor of type', and 'actor with tag'. The first two options are highlighted with a blue selection bar.

En donde dice "bounce when colliding with: Actor of type" desplegamos la opción Actor of Type y nos aparece lo siguiente.



Que quiere decir que Colisione y rebote con un tipo de actor o bien con un actor con etiquetas, aquí es donde la cosa se pone buena, porque con el uso de etiquetas solamente tenemos que hacer las cosas una sola vez en lugar de hacerlo cada vez que se cree el mismo objeto en este caso enemigos.

Vamos a la pestaña Actors



y presionamos en el siguiente botón



este botón es el de etiquetas, nos aparece lo siguiente.



Vamos a cambiarle el nombre por el de enemigos haciendo doble clic sobre la nueva etiqueta.



## Enemigos

Ahora vamos a colocar al objeto ovni dentro de esta etiqueta, Desplegamos la pestaña All



arrastramos el actor 4 que es el ovni hacia la etiqueta enemigos.

Para verificar podemos ver que en la pestaña Actor en Tags el actor se adhirió a la etiqueta Enemigos.

Ahora regresamos al Behavior Collide en el Actor Avión y podemos elegir dos opciones, Que el actor colisione y rebote

con un actor en específico en este caso el Actor 4 que es el Ovni.

The screenshot shows a game editor interface with a toolbar at the top. A dropdown menu is open, showing 'On' and 'Collide'. Below the toolbar, there is a configuration section for collision. It says 'bounce when colliding with:' followed by a dropdown menu set to 'actor of type' and another dropdown menu set to 'Actor 4'. There is also a small button with a downward arrow.

O bien con un actor que se encuentra dentro de una etiqueta y esto sería lo ideal, te voy a decir porque, supongamos que aparte de ovnis tengamos otro tipo de enemigos, como sean tanques de guerra, entonces deseamos que de igual forma el avión al colisionar y rebote con el tanque de igual forma rebote o explote, entonces para ahorrarnos tiempo, tan solo tendríamos que arrastrar el actor tanque de guerra hacia la etiqueta Enemigos y de esta forma se ejecuta la misma acción, es como decir, todos los actores que se encuentren dentro de la etiqueta Enemigos son los actores que no se llevan con el Actor 1 que es el Avión.

Entonces dejamos la opción de Collide así.

The screenshot shows a game editor interface with a toolbar at the top. A dropdown menu is open, showing 'On' and 'Collide'. Below the toolbar, there is a configuration section for collision. It says 'bounce when colliding with:' followed by a dropdown menu set to 'actor with tag' and another dropdown menu set to 'Enemigos'. There is also a small button with a downward arrow.

Ejecutamos el juego



y podemos ver que el avión al chocar con el actor ovni lo avienta hacia el infinito y el choque hace que se cambie la

trayectoria del actor avión. Porque ha ocurrido un rebote.

# Colisionar y Desaparecer Actores

Pues bien ya vimos cómo hacer que un actor colisione y Rebote, ahora veremos cómo podemos hacer que un actor Colisione y desaparezca ya que la mayoría de juegos sino es que todos basan en esto su tipo de juego, en hacer que el enemigo no acabe contigo y si lo hace desaparecer al actor jugador.

Este Behavior es muy parecido a Collide, pero para colisionar y desaparecer actores tenemos que usar una regla



Rule

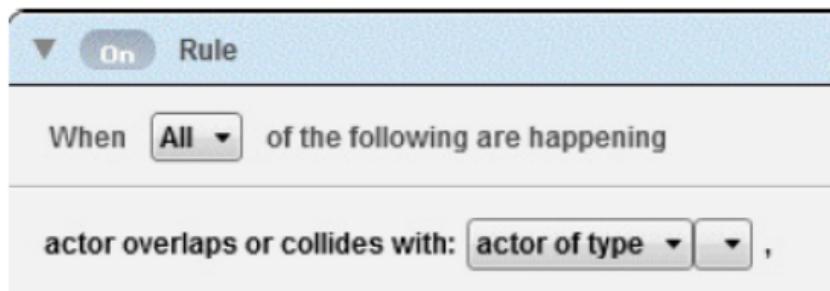
tan solo la arrastramos nuevamente hacia el

Backstage del Actor1 que es el Avión.

The screenshot shows the RoboRealm software's "Rule" configuration window. At the top, there is a dropdown menu with a downward arrow icon, followed by the word "On" and the word "Rule". Below this, a condition is defined: "When All of the following are happening". A large text input field below this contains the placeholder text "type or drag in a condition". Two options are listed under this field: "do" and "else".

Vamos ahora a arrastrar la behavior Collision que es una

Condición, hacia la parte en donde dice "type or drag in a condition".



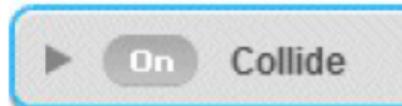
▶ do

▶ else

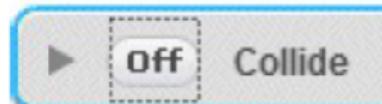
Vamos a cambiar las opciones en actor overlaps or collides with por actor with tag y elegimos la etiqueta Enemigos.

actor overlaps or collides with: actor with tag ▾ Enemigos ▾ ,

Recuerda desactivar poner en Off la regla Collide que usamos previamente en la sección Colisionar y Rebotar Actores. Como se muestra en la imagen, presiona en el botón On.



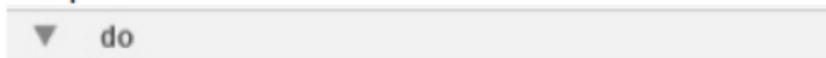
Para que la dejes en Off o fuera de servicio.



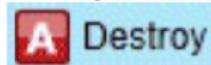
Hecho esto regresamos a la regla en donde colocamos el behavior collision, en donde dice Do vamos a colocar un behavior



Así debe lucir la regla de colisión y desaparición del actor.



Como puedes ver el behavior



dice lo siguiente que al activarse el behavior Destroy, destruir a este actor. Ya hecho esto vamos a probar si realmente al colisionar el avión en contra de los enemigos el actor avión va a desaparecer.



El avión impacta en contra del ovni y el actor1 o sea el jugador desaparece.



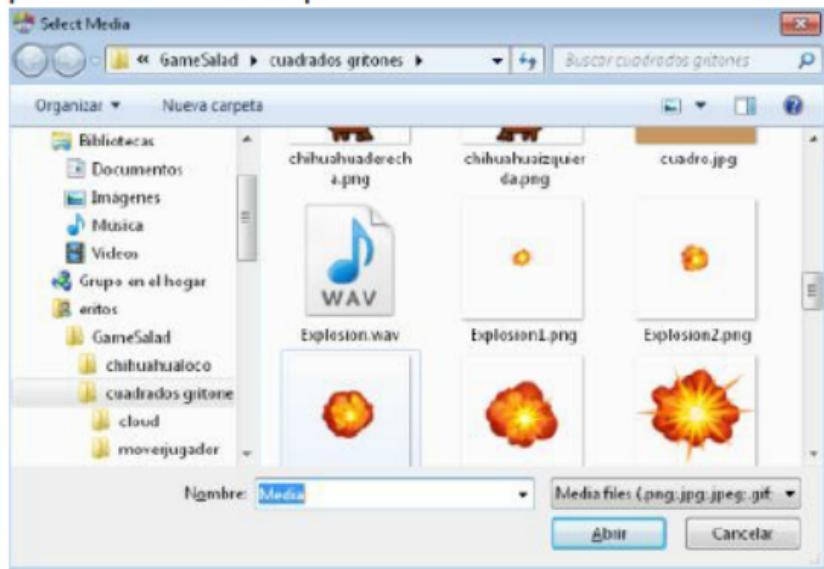
Aun se nota muy burdo, pero podemos adherir un efecto de fotogramas para hacer una explosión cuando el actor impacte al ovni, de igual forma podemos adherir un sonido, recuerda que la adición de un fotograma ya lo vimos con anterioridad en la sección **Animar Actores**, y puedes colocar un behavior Animate arriba del Behavior Destroy.

# Adherir Sonidos Al Juego

El sonido también es importante ya que hace que el jugador se interiore con el juego haciéndolo más atractivo, imagínate un juego en donde el jugador impacte en contra de otro enemigo y no escuches una explosión, o simplemente veas nada más la animación de la explosión, esto rápidamente lo convertiría en un juego muy aburrido, ya que el sonido también es importante, o imagínate Angry Birds sin el sonido gracioso que hacen las aves, ¿hubiese tenido el éxito que ahora tiene? Pienso que no.

A continuación voy a agregar un sonido de explosión desde la

pestaña media  aparece la ventana.



Elijo el archivo explosión.wav y este se coloca en lista de archivos en Media. Y es convertido en un archivo con extensión .ogg de igual forma cuenta con icono de reproducción.



Explosion

.ogg

Presiona en el botón de reproducción y escucha el sonido. Nos vamos a la pestaña Behaviors



vamos a elegir el behavior



Play Sound

el cual sirve para reproducir sonidos, como puedes ver también existe otro behavior que se llama



Play Music

el cual

sirve para adherir música, la diferencia entre un sonido es que cuenta con una reproducción muy corta de audio, mientras que la música tiene una extensión un poco más larga de audio, normalmente pasando del minuto.

Arrastramos el behavior



Play Sound

hacia la regla que creamos anteriormente para colisionar y desaparecer el actor, y la coloco en Do después de la animación de explosión que he creado y te sugiero que tu realizaras.

▼ On Play Sound

play sound:   loop  run to completion

at volume:     position sound

at pitch:     velocity Shift

En donde dice play sound: desplegamos y elegimos el sonido explosión.

▼ On Play Sound

play sound:  **Explosion**  loop  run to completion

at volume:     position sound

at pitch:     velocity Shift

Reproducimos el juego



y el sonido se reproduce al impactar el jugador avión en contra

del enemigo ovni, sin ningún problema, ahora bien puedes usar sonidos para lo que tú quieras, tan solo asegúrate de colocarlo cuando se ejecute una regla o un comportamiento para que se reproduzca correctamente.

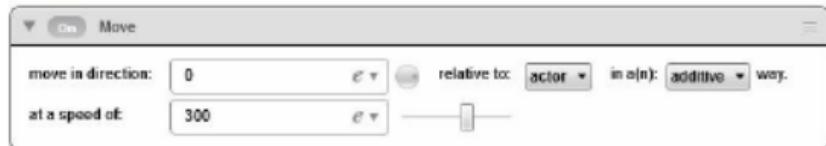
# Adherir Gravedad a un Actor

Ok, la mayoría de los juegos trabajan usando un tipo de gravedad pero eso sí, depende del tipo de juegos que estemos creando, como estamos haciendo un juego de disparos, le voy a adherir gravedad al avión, pero ojo, podemos adherir gravedad al nivel, aunque para ser honesto, prefiero adherir gravedad directamente al actor. Vamos a hacer que la gravedad empuje el avión hacia abajo como ocurre en la vida real.

Vamos al Behavior



y la colocamos en el backstage del actor1 que es el avión. Luce de la forma siguiente.



Recuerdas las coordenadas cartesianas, entonces aquí las vamos a usar nuevamente, como quiero que la gravedad se aplique como ocurre en la vida real, voy a 270 en move in direction y en at a speed of: 120

Reproducimos el juego

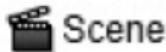


y vemos que la fuerza de gravedad es aplicada

individualmente al avión, si presionamos el botón de ir hacia arriba el avión levanta el vuelo, pero si lo soltamos la fuerza de

gravedad aplica su fuerza para empujar el objeto hacia el suelo. Podemos agregar la fuerza de gravedad para todos los actores en Gamesalad, debemos ir

a la pestaña de escena



desplegamos en donde dice Gravity y cambiamos el valor de Y a 120.

Gravity	point
X 0	real
Y 120	real

Ahora ejecutamos el juego



y vemos que todos los actores en el juego reciben la fuerza de gravedad, el ovni, el fondo de nubes y el propio avión, vamos a desactivar el behavior Move en el actor avión.

The screenshot shows the 'Move' behavior settings for an actor. It includes fields for 'move in direction' (set to 270), 'at a speed of' (set to 120), and various options like 'relative to:' (set to 'actor'), 'in a(n):' (set to 'additive'), and 'way'.

Ejecutamos nuevamente el juego



y la fuerza de gravedad se sigue aplicando a todos los actores en la escena, para que no se aplique la fuerza de gravedad en todos los actores vamos a la pestaña actor



de las nubes y deseleccionamos la pestaña Movable.

Attributes

Game	Scene	Actor
Actor 3 (instance)		
Friction	3	real
Bounciness	1	real
Fixed Rotation	<input type="checkbox"/>	boolean
Movable	<input checked="" type="checkbox"/>	boolean

Con esto el fondo de las nubes ya no se moverá y la fuerza de gravedad no le aplicara. ¿Pero y con el actor ovni? Bueno aquí es donde viene la complicación, el ovni pues si queremos que se mueva en algún momento así que no podemos deseleccionar

la casilla movable. Entonces lo ideal es crear una gravedad individual para cada actor, como lo hicimos en el primer procedimiento, agregando el Behavior Move



Move

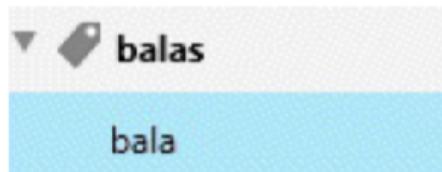
Ahora si ejecutamos el juego, tenemos lo siguiente, un avión al que la fuerza de gravedad lo empuja hacia el suelo, y que si presionas la tecla hacia arriba toma vuelo, y hacia la derecha hace que se mueva la cámara para que le siga, de igual forma si colisiona con un Ovni (enemigo) el avión explotara y se reproduce una animación de explosión así como un sonido del mismo tipo, y esto tan solo lo hemos logrado en unos cuantos minutos. ¿Qué te ha parecido? Fácil ¿verdad?

# Crear y Realizar Disparos

En los juegos siempre es importante que el jugador o los enemigos realicen disparos, pero eso si no en todos los juegos se tiene esta prestación, pero como el objetivo de este libro es enseñarte todos los trucos para hacer juegos, entonces vamos a ponerle al avión la opción de realizar disparos y cuando este disparo impacte sobre el enemigo ovni, este desparezca.

Lo primero que vamos a hacer es lo siguiente, nos

positionamos en la pestaña Actors  creamos un actor llamado bala y lo colocamos dentro de una etiqueta llamadas balas.



Ahora a la pestaña Actor en attributes

 Actor y configuraremos las opciones del actor bala, primero cambiamos el color del actor por un Azul Rey.

► Color		Color
Image	<input type="button" value="No Image ▾"/>	Image
Tags	<input type="button" value="Select Tags ▾"/>	text

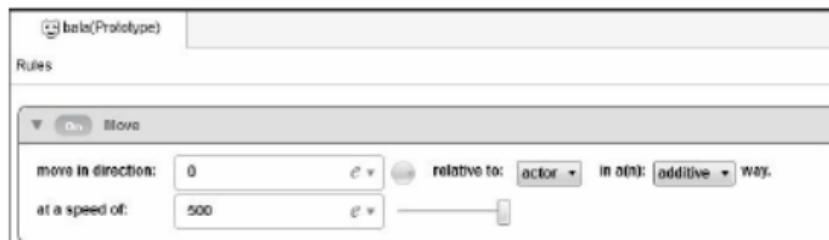
Modificamos el tamaño en Width ponemos 10 y en Height de igual forma 10, ahora ya está pequeño.

▼ Size	Size
Width	10
Height	10

Ahora debemos adherir un behavior a este mismo actor, utilicemos el behavior Move



y ponemos los siguientes valores como se muestra en la imagen.



Ok, dejamos al actor bala, para centrarnos ahora en el actor avión, adherimos una nueva

regla, adhiriendo un behavior key



y ponemos como la tecla a accionar Space o

## Espacio.

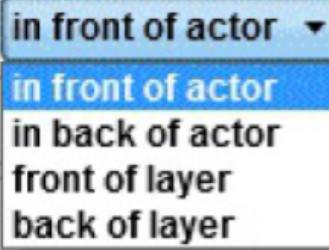
A Scratch script titled "Rule". The condition is "When All of the following are happening". The first event is "receives a key press: space is: down".

En la acción do de la misma Regla ponemos un behavior

 **Spawn Actor** como se muestra en la imagen a continuación.

A Scratch script titled "Balas". The behavior is "do". It contains a "spawn actor" block set to "bala" at "in front of actor" with "relative to: actor". It also includes "in the direction: 0" and "from position: 0" blocks.

Como puedes en la acción spawn actor: seleccionamos el actor bala y en place (que significa lugar) decimos que el spawn (espectro) debe salir del frente del actor, si despliegas place: cuenta con las siguientes opciones.



Como ves la opción nos pone en frente del actor (es la que estamos seleccionando) detrás del actor, frente a la capa y detrás de la capa.

Cuando activemos la tecla espacio, el actor creado previamente (bala) será lanzado desde el actor avión, en su parte frontal, en la dirección cero o sea en X positiva, relativamente desde el actor.

Si reproduzcamos el juego



podemos ver como el avión ahora ya realiza disparos desde el frente del avión.



Pero el disparo no surte efecto alguno en el actor ovni, porque necesitamos incluir otra regla, vamos a seleccionar el actor

ov ni.

Colocamos una Regla e incluimos una behavior collision



Collision

como puedes ver en la imagen.

The screenshot shows the GameSalad rule editor interface. At the top, there is a header bar with a 'Rule' tab. Below it, the 'When' section is set to 'All' of the following are happening. The condition listed is 'actor overlaps or collides with: actor with tag = balas'. The 'do' section contains a single action: 'destroy this actor'. This action is triggered by the 'On Destroy' event.

When All of the following are happening  
actor overlaps or collides with: actor with tag = balas

do  
destroy this actor

En actor overlaps or collides with: seleccionamos actor with tag y la etiqueta balas. En Do ponemos un behavior



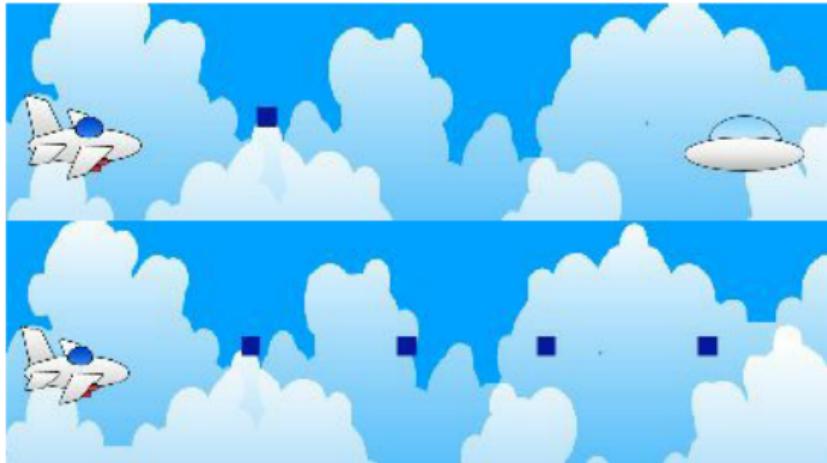
Destroy

con esto le decimos al actor ov ni que cada vez que sea golpeado por una bala entonces debe ser destruido.

Hacemos la prueba reproduciendo el juego



y podemos ver que cada vez que la bala del avión impacte al ov ni, este desaparecerá, con esto hemos creado disparos en GameSalad.



Para que el disparo no se pase cuando impacta al Ovni, debemos adherir una regla al disparo y cuando este choque con Enemigos entonces desaparezca.

Adherimos la Regla con el behavio Collision



Collision

como se muestra en la imagen.

The screenshot shows the Construct 2 rule editor interface for a Collision behavior. The title bar says "Collision". The main area has a "Rule" tab selected. The condition section is set to "When All of the following are happening". The first condition is "actor overlaps or collides with: actor with tag Enemigos". Below this is a placeholder box for additional conditions. The action section contains two lines: "do" and "else".

En do adherimos un behav ior



Destroy

para que cuando el disparo impacte al ovni que

previamente se encuentra en la etiqueta Enemigos sea destruido.



De esta forma ya no veremos ese horrible efecto que ocurre cuando el disparo impacta y sigue su trayectoria, lo más común es que un disparo desaparezca cuando impacta su objetivo.

A continuación vamos a realizar un Score para ir contabilizando los enemigos que vamos destruyendo en el juego.

# Mostrando y creando un Score

Ok, ya tenemos un avión que dispara y destruye a su enemigo pero necesitamos que se vaya contabilizando cuando estos sean destruidos, para eso necesitamos crear un Score (Puntuación).

Primero debemos crear un atributo en la pestaña



Game

un atributo es crear una

variable para ser utilizada y manipulada, recuerda que variable significa algo que cambia o tiene la propiedad de cambiar, imagina una caja de cartón, en donde un día podemos poner unos zapatos, otro día juguetes, etc.

Creamos el atributo en la pestaña antes mencionada con el nombre **puntuación** del tipo **integer**.



Boolean



Text



Integer



Real



Angle



Al elegir Integer nos debe aparecer la variable de esta forma.

New Attribute1 0

integer

Cambiamos el nombre de New Attribute a puntuación haciendo doble clic sobre la variable.

puntuacion 0

integer

Ahora vamos a crear un nuevo actor en la pestaña



presionamos sobre el botón una vez que este creado le cambiamos el nombre por el de Score.

Score

Seleccionamos al actor sin adherirlo a la escena nos movemos a la pestaña de opciones de Actor



Actor

y adherimos un behavior



## Display Text

Vemos las siguientes opciones que se muestran en la imagen.

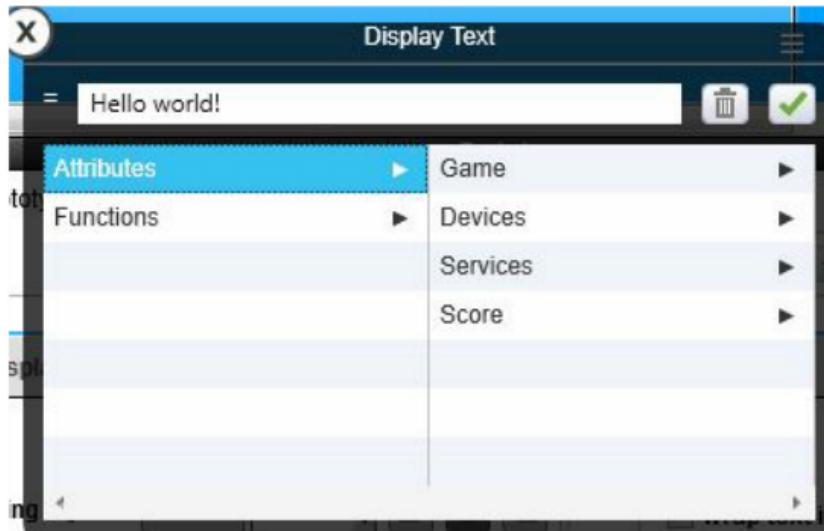
The screenshot shows the 'Display Text' configuration window. It has a title bar with 'On' and 'Display Text'. Below it, there's a 'display:' field containing 'Hello world!', a font style dropdown set to 'Arial' with a size of '30', and a color palette. To the right of the font settings are several icons: a bold button, a italic button, a underline button, a font size up/down button, and a 'wrap text inside actor' checkbox.

Como deseamos que se muestre la variable puntuación que previamente hemos creado, lo que tenemos que hacer es lo

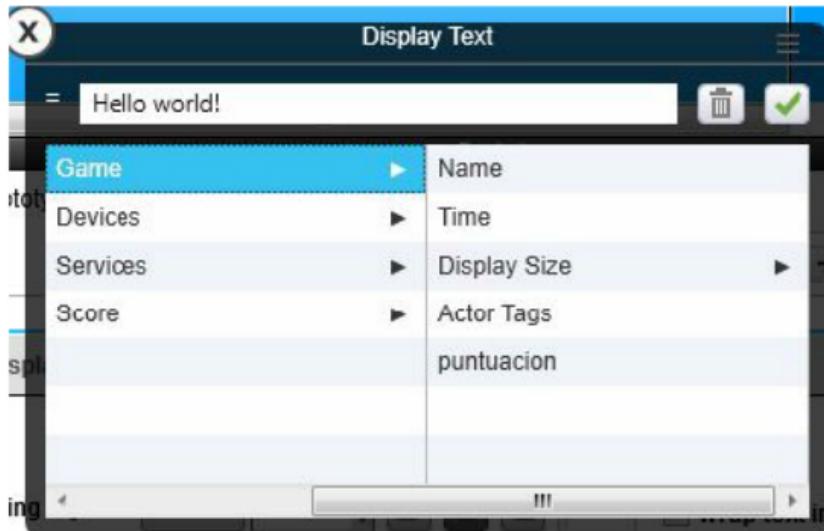
siguiente. Presionamos sobre este ícono y se nos mostrara a continuación la siguiente ventana.

The screenshot shows the 'Display Text' configuration window again. This time, the 'Attributes' option is selected in the left sidebar, which causes the main panel to change. The main panel now displays a list of attributes: 'Hello world!', 'Attributes', and 'Functions'. There are also some other partially visible items like 'ispol' and 'ring'. At the bottom of the main panel, there are three buttons: a trash can, a red X, and a green checkmark.

Presionamos en Attributes y se despliegan otras opciones.



Seleccionamos Game a continuación y se muestran más opciones.



Si eres observador podrás ver que del lado derecho al final se muestra la variable

puntuación que hemos creado en



lo que debemos hacer es seleccionarla haciendo doble clic y nos quedaría de la siguiente forma.

Display Text

game.puntuacion

Game	Name
Devices	Time
Services	Display Size
Score	Actor Tags
	puntuacion

vino

Al seleccionarla nos muestra lo siguiente

game.puntuacion

Presionamos en el botón



para aceptar el parámetro, si deseas eliminarla tan solo presiona el botón



y el parámetro se borra.

Aceptamos el parámetro y ahora el behavior display text se muestra así.

▼ On Display Text

display: game.puntuacion e ▾

in the following style: Arial 30 ▾

wrap text inside actor

He puesto los parámetros de in the following style así.

in the following style: Times New Roman 40  wrap text inside actor

Ahora que ya está listo el behavior Display Text, regresamos a



la pestaña debemos arrastrarlo a la escena, lo voy a colocar en la parte derecha superior.

Si reprodujimos el juego



podemos ver que el Score ya se encuentra en la pantalla del juego, pero el problema ahora es que tiene un horrible fondo blanco y eso en realidad no luce nada bien.



Lo que tenemos que

hacer es irnos a las propiedades del actor



Actor

y presionar en Color.



Game



Scene



Actor

Score (instance)

Revert To Prototype

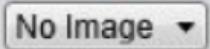


▶ Color



Color

Image



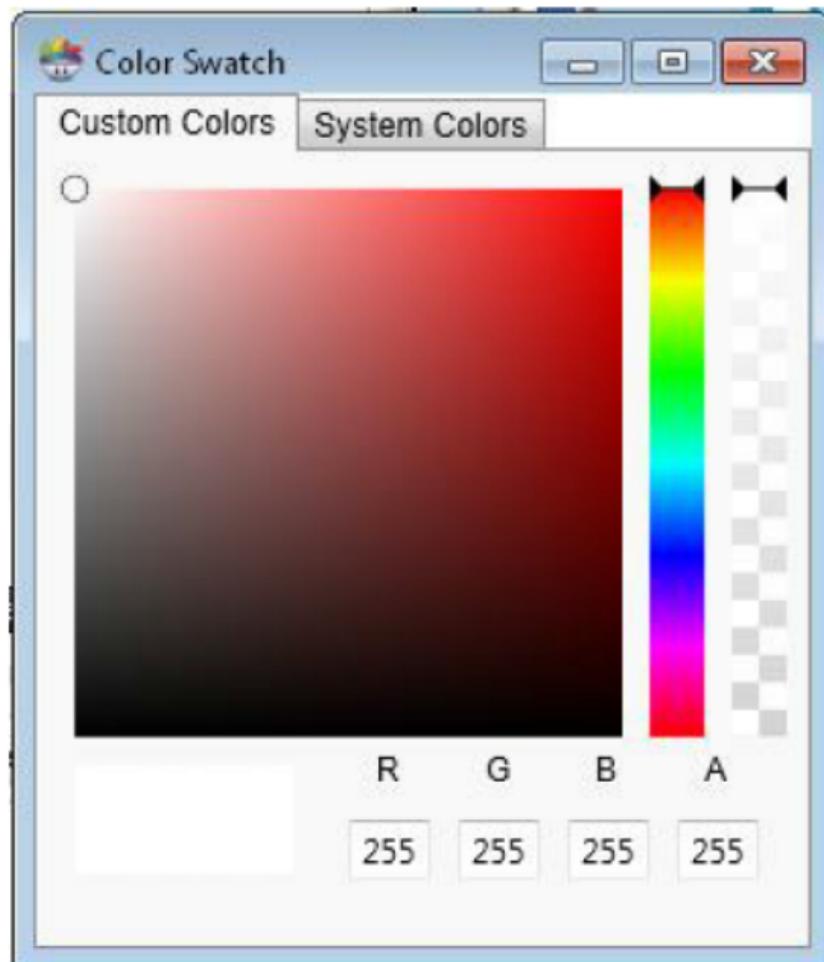
Image

Tags

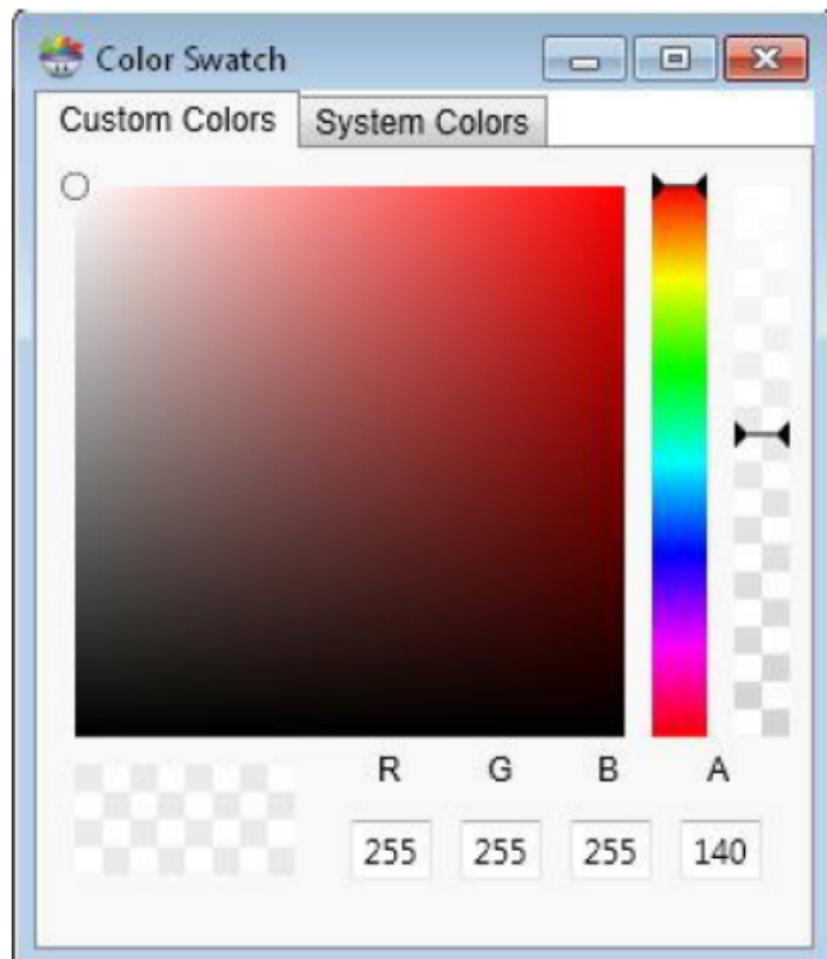


text

Nos aparece la siguiente ventana.



Y en el selector que tenemos en la parte derecha en donde se ve cierta transparencia la seleccionamos y la bajamos hasta cero.



Ahora si vemos el actor en la escena se ha tornado transparente.



Y si reproduzcamos el juego, podemos ver que ahora se ve el Score como luciría en cualquier juego.



Existe otro pequeño inconveniente, si eres detallista, al mover el avión hacia el lado derecho pues la cámara sigue al actor pero el Score se va quedando atrás. Para arreglar este problema vayamos a Layers



y Creamos una nueva layer llamada Graficos Fijos.



Name	Scrollable
►  <b>Background</b>	<input checked="" type="checkbox"/>
►  <b>Nubes Fondo</b>	<input checked="" type="checkbox"/>
►  <b>Graficos Fijos</b>	<input checked="" type="checkbox"/>

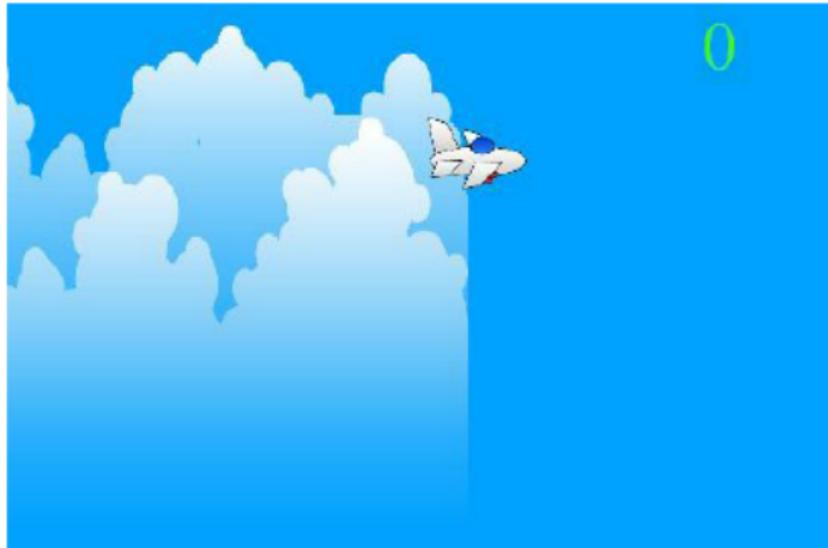
Pasamos el actor Score que se encuentra en Background a Graficos fijos arrastrándolo.

►  <b>Background</b>	<input checked="" type="checkbox"/>
Score	
Actor 4	
Actor 2	

Debe quedar de la siguiente forma.

Layers					
Name		Scorable			
<b>Background</b>		<input checked="" type="checkbox"/>			
Actor 4					
Actor 2					
<b>Nubes Fondo</b>		<input checked="" type="checkbox"/>			
<b>Graficos Fijos</b>		<input checked="" type="checkbox"/>			
Score					

En Graficos Fijos tenemos una casilla de verificación que se llama Scorable, la deseleccionamos y reproduccmos nuevamente el juego, ahora podemos ver que el grafico ya no se queda fijo al moverse el avión por la pantalla.



Ahora lo último que debe suceder es que cada vez que el disparo del avión impacte al ovni debemos ir adquiriendo puntos.

Vamos al actor Bala en la pestaña Actors



y vamos hasta la regla que creamos para que el actor balas colisionara con la nave y la destruyera.

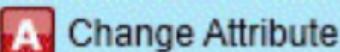
When **All** of the following are happening

actor overlaps or collides with: **actor with tag** **Enemigos**,

Type or drag in a condition

**do**

En do metemos un behavior



**Change Attribute**

y ponemos lo siguiente, que ya habíamos

hecho con anterioridad.

On **Change Attribute**

**set:** **game.puntuacion** **to:** **game.puntuacion+10**

En set ponemos **game.puntuacion** y **to:** **game.puntuacion+10** esto quiere decir que cada vez que una bala impacte al actor ovni debe adherir 10 puntos al atributo puntuación el atributo **puntuación** es mostrado en el actor Score.

Reproducimos el juego y si la bala del avión pega en el ovni entonces obtengo diez maravillosos puntos.

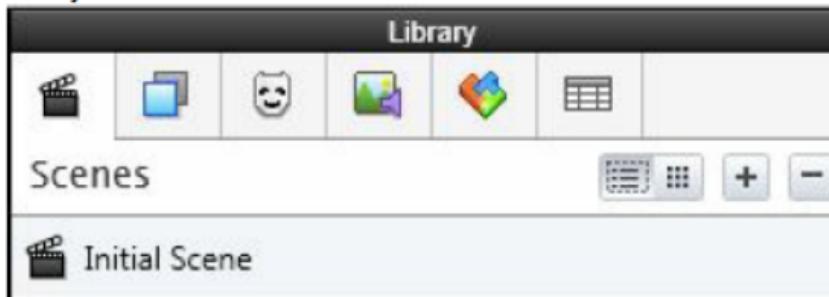


Ahora ya puedes crear tu propio sistema de puntuación para tu juego, ya ves lo poderosos que son los atributos en Gamesalad, con ellos podemos realizar funciones inimaginables, pero eso sí, necesitas pensar bien la idea para que los atributos realicen su tarea.

# Crear Escenas

Hasta este momento hemos estado trabajando en una escena y es la que trae por defecto Gamesalad, para hacer más escenas nos dirigimos a la pestaña Scenes

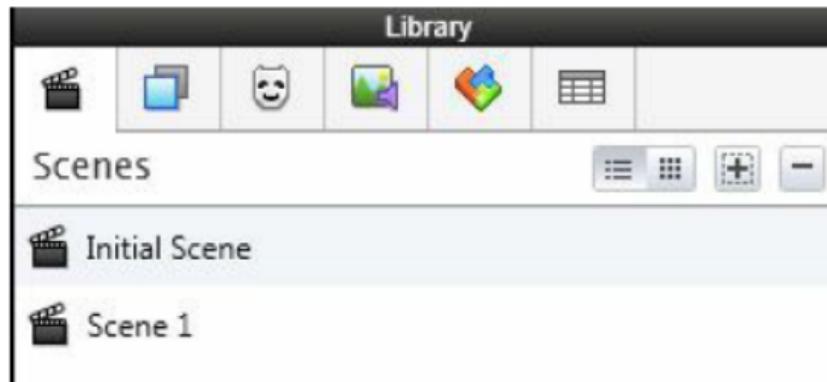
 y tenemos una escena que es con la que hemos estado trabajando.



Ahora presionamos en el botón

 y la escena se adhiere a la lista con el nombre Scene1

si existiera ya Scene1 se crearia Scene2 y así sucesivamente.



Modificamos el nombre por el de Game Over. Y dejamos en opciones el color que trae por defecto que es el negro.

A esta escena voy a adherir dos actores, unas letras de Game Over y un botón de Reiniciar el Juego.



Esta escena nos va a servir para mostrar el final del juego, pero primero el avión tiene que desaparecer para que se muestre la escena, entonces si el avión impacta al ovni se muestra la pantalla de game over, y si presionamos en el botón Reiniciar, el juego comienza nuevamente.

Vamos a la pestaña



Game y Creamos un nuevo atributo del tipo booleano, que se llame

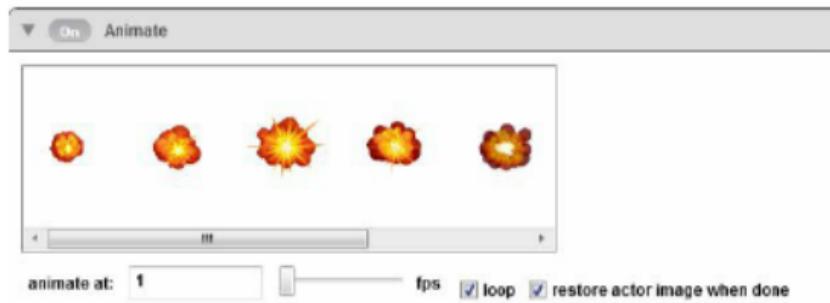
juego\_terminado. No selecciones la casilla de verificación déjala tal cual esta.

Attributes			
Game	Scene	Actor	
Game Attributes			+ -
Name	default name	text	
Time	0	real	
▶ Display Size		Size	
Actor Tags	All	Enemigos	text
puntuacion	0		integer
juego_terminado			boolean

Ahora vamos al actor avión y vamos a la regla en donde creamos una colisión con la etiqueta enemigos.

The screenshot shows the RoboRealm rule editor interface. At the top, there's a toolbar with icons for Rule, Add, and Remove. Below it, a dropdown menu says "When All of the following are happening". Underneath, a condition is listed: "actor overlaps or collides with actor with tag Enemigos". A note below says "type or drag in a condition".

En dó ya teníamos lo siguiente.



## La animación.

play sound: **Explosion**  loop  run to completion  
at volume: **1**  position sound  
at pitch: **1**  velocity Shift

El sonido de la explosión al chocar contra el ovni

On **Destroy**  
destroy this actor

Y finalmente el behavior destroy, lo que vamos a hacer ahora es agregar debajo del behavior destroy un behavior

**A Change Attribute** para cambiar el atributo de juego\_terminado a verdadero.  
Quedaría de la siguiente forma.

▼ On Change Attribute

set:  a ▾ to:  e ▾

En to: debemos poner el valor siguiente

to:  e ▾

Creamos una nueva Regla en el mismo actor avión

▼ On Rule

When  of the following are happening

type or drag in a condition

▶ do  
▶ else

Ponemos un behavior

**C Attribute** para consultar si el atributo juego\_terminado es verdadero realizar una acción en do.

▼ On Rule

When  of the following are happening

if  a ▾ is  e ▾

En donde ponemos un behavior



Change Scene

para ir a la escena Game Over cuando el avión sea destruido.

Probamos el juego



y podemos verificar que si el avión choca en contra del ovni, entonces se nos muestra la escena de Game Over.



Ahora vemos que se nos muestra la opción de Reiniciar el juego, entonces debemos hacer lo siguiente.

Seleccionamos el actor Boton\_reiniciar en la pestaña Actors

## Score

--- Actor 5



Boton\_reiniciar

Nos vamos al Backstage del botón para poner behaviors. Agregamos una nueva regla que indique lo siguiente.

▼ On Rule

When All of the following are happening

type or drag in a condition

► do

► else

Vamos a poner un behavior Touch



Touch

que le indique cuando el toque sea presionado entonces ir a la escena del juego.

▼ On Rule

When All of the following are happening

touch is: pressed ▾

type or drag in a condition

En dónde ponemos el siguiente behavior



Reset Game

para reiniciar el juego desde su principio.



do



On

Reset Game

reset the game

type or drag in a behavior or block



else

Y ahora el botón de Reiniciar queda totalmente habilitado para poder iniciar el juego nuevamente, ahora ya contamos con un juego y en tan solo unas cuantas horas. ¡Increíble! ¿Verdad?

# Hacer que Enemigo Ataque el Jugador

Para agregarle más dificultad al juego, vamos a hacer que el enemigo ataque al jugador, para que el juego no resulte tan aburrido.



Accelerate Toward

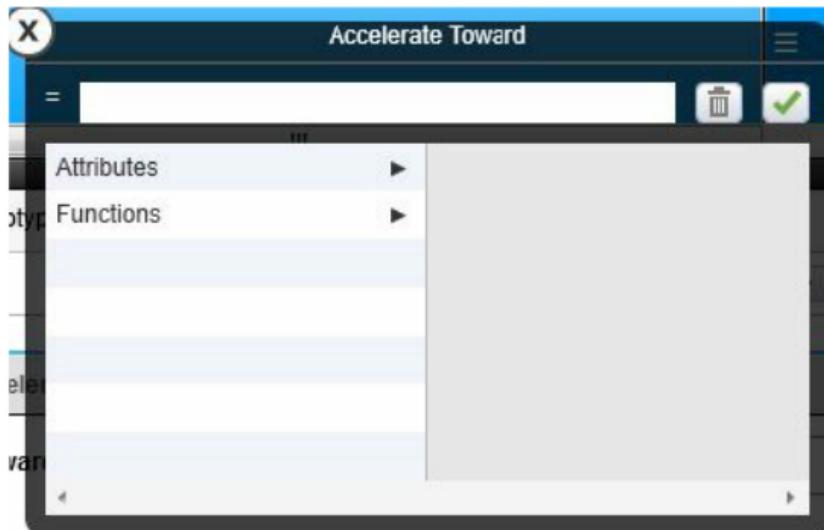
Lo único que tenemos que hacer al ovari, el behavior luce de la siguiente forma. es lo siguiente, adherimos un behavior

The screenshot shows the RoboRealm configuration interface with the 'Accelerate Toward' behavior selected. The settings are as follows:

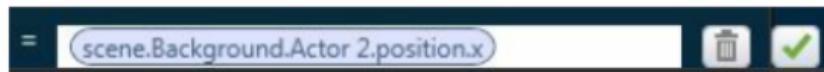
- On: On
- accelerate towards the point: 0, 0 (relative to: 0, 0)
- scene: (dropdown menu)
- with acceleration: 100

En donde dice accelerate towards presionamos en el ícono

y nos aparecerá la siguiente ventana.



Y ponemos el siguiente atributo. En el campo.



Seleccionamos Scene, Background, Actor 2 (Actor Avión) en la posición X del actor avión, lo que hicimos fue decirle al actor ovni que se mueva en la dirección X del actor avión.

Hacemos lo mismo con el otro parámetro después de la coma, pero en la position Y.



Ahora si reproduccimos el juego



el actor ovni, seguirá al actor avión y tratará de colisionar

con él, de esta forma hemos hecho esto con un solo behavior.



# Juego para Android

Como este libro es para crear juegos para dispositivos móviles como Android ahora viene realmente lo bueno, a partir de aquí si deseas publicar tu juego para dispositivos móviles deberás contar con los siguientes requisitos.

## **Descargar el SDK para Android**

Lo primero a realizar es descargar el SDK (Software Development Kit) para Android, el cual podemos encontrar en la siguiente dirección web.

<http://developer.android.com/intl/es/sdk/index.html>

Presionamos en el botón azul para descargar ADT bundle para Windows.



**Download the SDK  
ADT Bundle for Windows**

Nos aparece la siguiente pantalla. Que nos verificar la casilla de en donde hemos entendido los términos y condiciones, verificamos la casilla.

I have read and agree with the above terms and conditions

32-bit     64-bit

En donde dice 32-bit y 64-bit dependerá del sistema operativo que estemos utilizando, el mio es Windows 7 profesional de 32-bit, normalmente todas las computadoras tienen instaladas el sistema operativo de 32-bit, no es común encontrar el sistema operativo de 64-bit. Así que elegimos el de 32-bit.

Entonces el botón para descargar el SDK se habilita.

### Download the SDK ADT Bundle for Windows

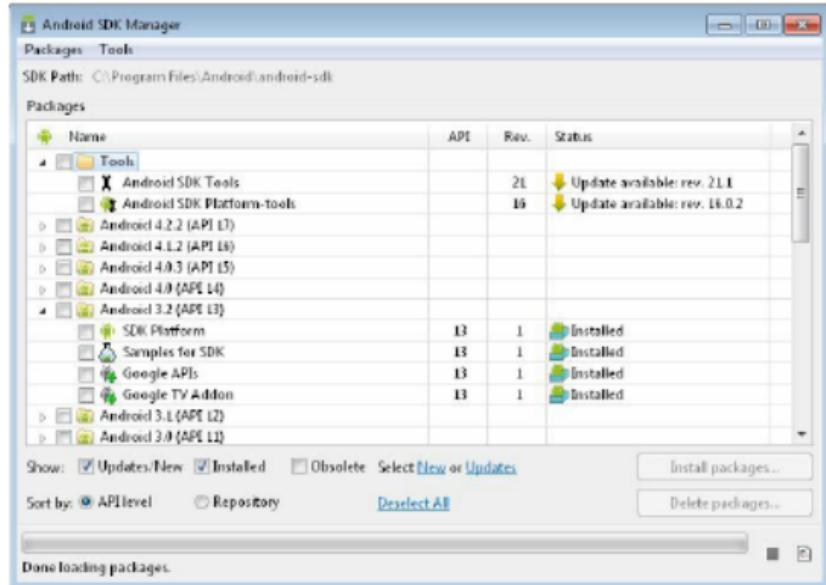
La descarga comienza y el SDK es descargado a la computadora. Cuando lo instalamos nos muestra dos opciones.

 AVD Manager.exe

 SDK Manager.exe

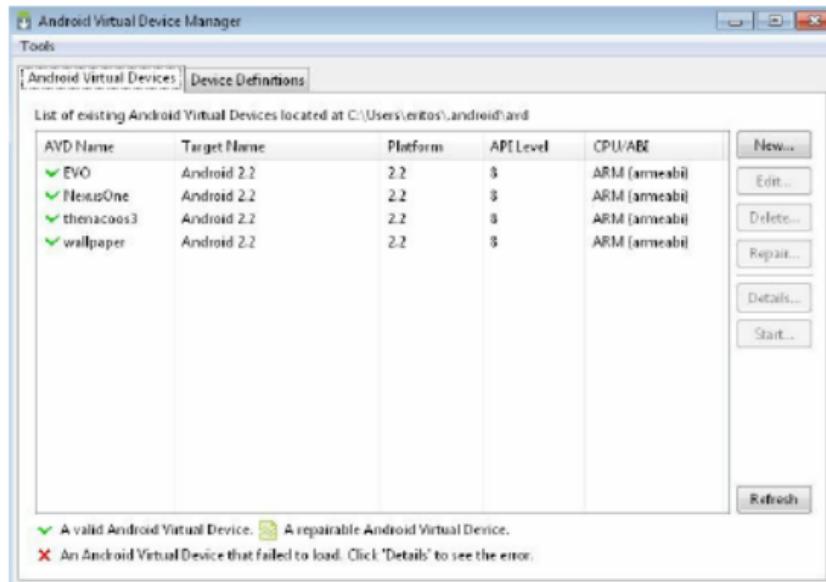
El SDK manager es en donde administraremos

todas las actualizaciones y versiones anteriores del sistema Operativo Android. Al hacer doble Clic, nos aparece la ventana siguiente.



En donde se muestran versiones 4.2.2 hasta la más antigua que es la Android 1.5. de esta forma se puede actualizar el SDK de Android.

El ícono AVD Manager no lo utilizaremos en este libro ya que sirve para crear emuladores de teléfonos Celulares para android. La ventana luce así.



**Ahora que ya descargamos el SDK de Android debemos dejarlo tal cual.**

## Crear una Keystore

Para publicar una app en la tienda google play debemos contar con una key store, esta key store es un identificador que solamente debe tener el desarrollador del software, con esto se evita que alguien descargue el archivo APK y se apodere de la aplicación, por eso es muy importante que solamente tu tengas acceso a esta y nadie más. Todas las aplicaciones para Android tienen una keystore. Vamos a aprender a crear una para el juego.

Abrimos la línea de Comandos, presionando en el icono de Windows.



Tipeamos las siguientes letras CMD.

Y se nos muestra la línea de comandos de Windows.

C:\Users\eritox>

Debemos poner lo siguiente:

```
"keytool -genkey -v -keystore  
nombre_compania .keystore -alias avionjuego-  
keyalg RSA  
-keysize 2048 -validity 10000"
```

En donde dice nombre\_compania debes poner un nombre que elijas para tu empresa desarrolladora de juegos, o bien tu nombre, en donde dice avionjuego, puedes poner el nombre del juego o como le quieras nombrar.

Si has realizado los pasos correctamente se te mostrara una opción para que pongas una contraseña para la keystore, ojo nunca olvides esta contraseña te lo digo por experiencia, si la pierdes, a veces crear una nueva se convierte en una pesadilla, asegúrate de guardar bien tu contraseña, después te muestra una serie de preguntas que tu solamente sabes, y finalmente elije otra contraseña para tu alias.

Tu key store es guardada en tu computadora, comúnmente en la carpeta del usuario, esta key store la necesitaras para publicar

tu app en Google play.

## **Instalando tu Telefono**

1. En tu teléfono android o tableta android, ir a configuración y luego a la opción Aplicaciones.
  2. Verifica la casilla Fuentes desconocidas para permitir la instalación del Visor de Gamesalad.
  3. Navega a la opción almacenamiento USB.
  4. Activa Depuración USB Desarrollo.
- Almacenamiento en Configuraciones, des verifica que se encuentra en Configuración, Aplicaciones,

## Instalando el Visor de Gamesalad

1. Descargar el ultimo Visualizador de Gamesalad para Android desde la página,

<http://arcade.gamesalad.com/download/getApk>

2. Ahora Conecta tu Teléfono o Tableta con Android a tu computadora vía USB.

3. Para verificar la conexión, corre la siguiente línea de comandos en un terminal CMD "~/sdk/platform-tools/adb devices"

4. En la ventana de Comando, en donde haz descargado el visor de Gamesalad que debe estar en un archivo APK por ejemplo, "cd ~/downloads".

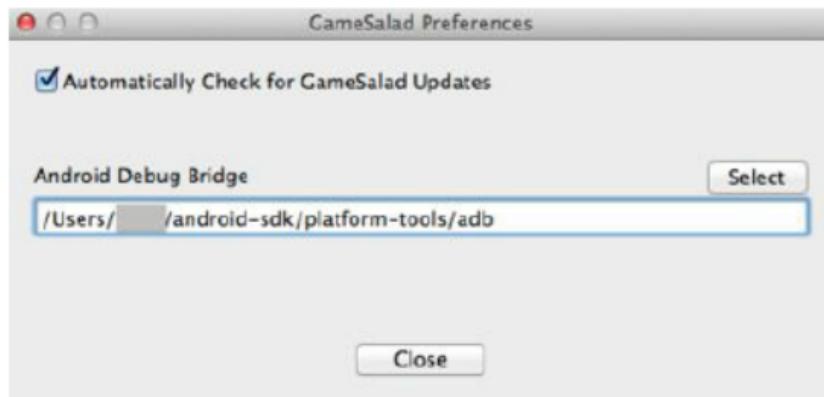
5. Instala el visor de Gamesalad usando adb haciendo correr la siguiente línea de comandos desde CMD:

"~/sdk/platform\_tools/adb install GameSaladAndroid\_Viewer-0.10.01-beta.apk"

# Probando un dispositivo Android usando el visor de Gamesalad

Para usar la opción de crear juegos para android, deberás contar con la versión pro de Gamesalad. Deberás asegurarte que el creador de gamesalad tiene la ruta Correcta en donde se encuentra el SDK de Android.

1. Primero Lanza el Creador.
2. Ir a las preferencias en Gamesalad, y cargar el folder del SDK como se muestra en la imagen, presiona en el binario de adb, ahora ya podrás ver el proyecto en tu dispositivo con Android.



## **Usar el Teléfono o Tableta vía USB**

Por recomendación siempre trata de utilizar la opción de probar tus juegos vía usb.

1. Conecta tu dispositivo Android con el cable usb a tu computadora.
2. Asegurate que el dispositivo esta puesto para carga solamente, y que no se inicie la SD Card.
3. Inicia el visor de Gamesalad en tu dispositivo Android
4. Espera hasta que el visor de Gamesalad te diga lo siguiente “Listening on USB” en la parte izquierda del fondo de tu pantalla con un icono verde de USB.
5. En el Creador de Gamesalad, presiona en “Preview on Android USB”.
6. Selecciona tu dispositivo desde la ventana desplegable.

## Por medio de Wi-Fi

1. Asegúrate que tu dispositivo Android y sistema están en la misma red inalámbrica (Conectados al mismo Modem).
2. Inicia el visor de Gamesalad en tu dispositivo.
3. Espera hasta que el visor de Gamesalad te diga lo siguiente "Connected to Wi-Fi" en la parte izquierda del fondo de tu pantalla con un icono verde circular como un punto.
4. Un icono "Preview on Android" aparecerá en Gamesalad.
5. Seleccionalo y despliega para ver tu dispositivo.

De esta forma ya podríamos tener un juego corriendo en nuestro dispositivo en Android, ya puedan ser teléfonos celulares, tabletas u otros electrónicos, siempre y cuando cuenten con sistema Operativo Android, podemos crear un juego de Gamesalad para estos.

A en el siguiente capítulo veremos algo de lo que posiblemente hasta ahora ya te has dado cuenta, si es que probaste tu juego en tu dispositivo, ¿Cómo se puede manejar el avión, si el teléfono y la tableta no cuentan con botones? Y a esto te digo, como ves desarrollar un juego necesita de mucha concentración, al principio del libro te dije que era cosa seria, mas no imposible, y lo mejor de todo fácil de hacer, tan solo hay que prestar un poquito de atención y usar mucho la lógica. Por ejemplo en este momento digo, si el dispositivo no cuenta con botones, entonces lo que debo hacer es crearlos. Y es lo que a continuación te voy a enseñar en este grandioso libro.

# Interactuar con el Acelerómetro

Estoy seguro de que ya has visto esos juegos en donde si mueves o inclinas el teléfono hacia la derecha el jugador se mueve hacia esa posición y si lo mueves hacia la izquierda realiza lo propio.

La mayoría de los teléfonos celulares hoy en día ya cuentan con esta función, se llama Acelerómetro y es un sensor que detecta movimientos sobre un espacio tridimensional, utilizando los ejes X, Y, Z. en donde X es el eje horizontal, Y el eje vertical y Z el eje de profundidad.

Hacer juegos que interactúan con el Acelerómetro resulta una tarea muy ardua, pero con Gamesalad esta tarea no tiene complejidad alguna. A continuación vamos a aprender a usar el acelerómetro.

Seleccionamos el Actor avión y en el Backstage vamos a crear una nueva Regla que llamaremos *Moviendo Jugador con Phone*.

The screenshot shows the Gamesalad interface with a new rule named "Moviendo Jugador con Phone". The rule is set to trigger "On" the "Phone" actor. The condition dropdown is set to "All", indicating that all specified conditions must be met. Below the condition dropdown, there is a text input field labeled "Type or drag in a condition" and a list of available conditions: "do" and "else".

Colocamos un Behavior



# Attribute

para consultar los atributos.

On Moviendo Jugador con Phone

When All of the following are happening

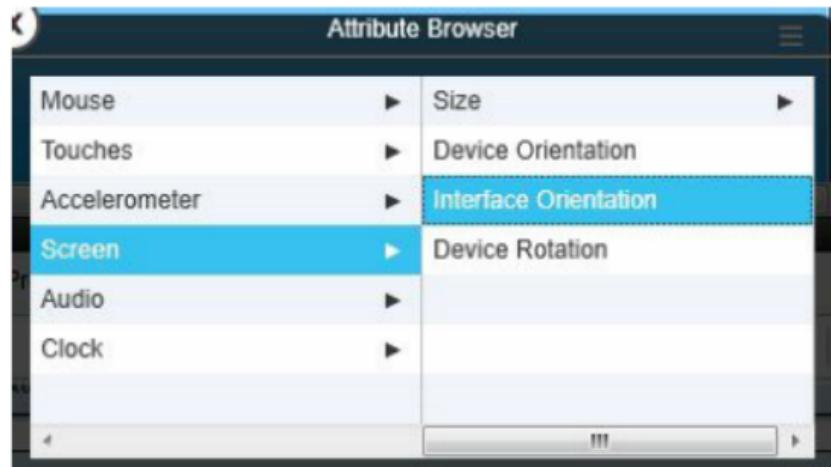
If  a ▾

En donde tenemos el campo con la palabra al inicio **if**,

presionamos el ícono para desplegar la siguiente ventana.



Seleccionamos Attributes, Devices, Screen, Interface Orientation.



El campo if debe quedar de la siguiente forma, lo que vamos a hacer, es mover el avión hacia el lado izquierdo de la pantalla cuando el teléfono se incline hacia la izquierda.

A screenshot of a Scratch script titled "Moviendo Jugador con Phone". The script starts with a "When [All] of the following are happening" hat. It contains an "if" condition with the following logic:  
if `(game.DeviceScreen.is [Landscape Left v])` is `[is]` `[Landscape Left v]`

Si desplegamos el cuadro que dice Landscape Left, nos mostrara las siguientes opciones.



A continuación te explico cada una de las opciones.

**Landscape Left**, es para cuando el teléfono se encuentra posicionado en forma horizontal e inclinado hacia la izquierda.

**Landscape Right**, El teléfono se encuentra posicionado en forma horizontal e inclinado hacia la derecha.

**Portrait**, El teléfono o tableta se encuentra en forma vertical en forma de portarretrato.

**Portrait Upside Down**, el teléfono o Tableta se encuentra en forma de portarretrato pero de cabeza.

**Face Up**, el teléfono o tableta está viendo de frente al usuario.

**Face Down**, el teléfono o tableta está viendo de frente al usuario pero el dispositivo se encuentra de cabeza.

**Unknown**, el teléfono o tableta se encuentra en una posición desconocida. Vamos a continuar, si vamos a los atributos de la escena, desplegamos la opción que dice Autorotate, y **Landscape Left** no está seleccionado, Seleccionalo.

▼ Autorotate

attributes

Landscape Left



boolean

Portrait



boolean

Landscape Right



boolean

Portrait Upside Down



boolean

Regresemos al backstage y a la regla *Moviendo Jugador con Phone* en do vamos a hacer lo siguiente.

Agregamos una regla dentro de do y ponemos un behavior



Attribute

y en if ponemos lo siguiente.

= game.Accelerometer.Y



Y en la pestaña desplegable el signo mayor que.



Y en el siguiente campo ponemos el valor de 0.01

0.01

e ▾

Lo que estamos diciéndole a la regla es que cuando estemos usando el acelerómetro, si el eje y es mayor que 0.01, ejecutemos lo que este en do.

En do agregamos el siguiente behavior



Accelerate

para que el avión se desplace hacia la izquierda a una velocidad de 200. Relativo a la escena, como estamos trabajando con la escena, entonces el movimiento debe ser relativo a esta.

▼ do



On Accelerate

accelerate in direction:

180

e ▾



relative to:

scene ▾

with acceleration:

200

e ▾

---

Vamos a poner una velocidad máxima al actor avión para que no acelere infinitamente, vamos a la pestaña



Actor y desplegamos en donde dice Motion

► Motion

attributes

Vamos hasta Max Speed y Apply Max Speed, en Max Speed ponemos el valor 200 y Apply Max Speed seleccionamos la casilla.

Max Speed 200 real

Apply Max Speed  boolean A

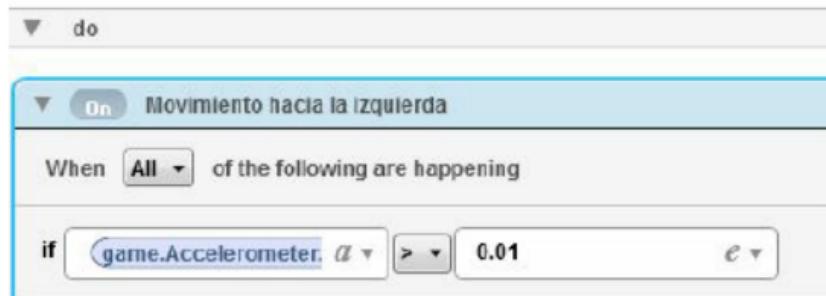
esta Regla que pusimos en do le vamos a poner el nombre de movimiento hacia la izquierda.

▼ do

▼ On Movimiento hacia la izquierda

When All of the following are happening

if game.Accelerometer.  $a <$  0.01  $> <$  0.01  $e <$



Ahora si quisiéramos movernos hacia la derecha tan solo tenemos que Copiar la regla *Movimiento hacia la izquierda*, presionando con Alt en el teclado de tu computadora y arrastrándola hacia abajo se realiza la copia.

► On Movimiento hacia la izquierda

▼ On Rule

When All of the following are happening

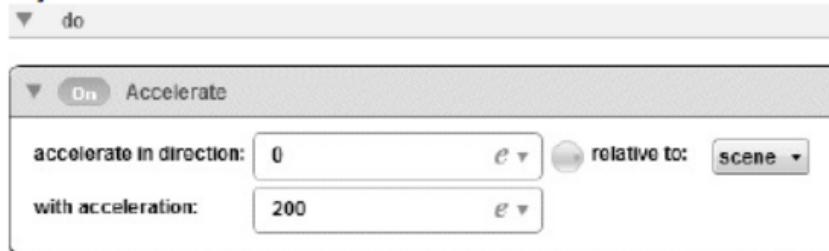
if game.Accelerometer.  $a <$  0.01  $> <$  0.01  $e <$



A esta nueva regla la vamos a llamar *Movimiento hacia la derecha* y vamos a cambiar los valores, el signo *Mayor que* lo cambiamos por *Menor que*, y en 0.01 le ponemos el valor pero negativo, -0.01



En la regla *Movimiento hacia la derecha* ponemos 0 en *direction* y en *with acceleration* valor de 200, en *relative to*: lo dejamos con Scene.



Si reprodujimos el juego en el dispositivo Celular o Tableta, entonces si movemos el eje Y del dispositivo el actor se moverá hacia atrás o hacia delante, según la posición del Teléfono o Tableta. De esta forma hemos aprendido a crear movimientos de actores con el acelerómetro.

Como un simple comentario y como algo lógico estos comandos no funcionan si testeas el juego en la ventana que aparece en la previa del juego. Debe ser obligatoriamente en un

dispositivo que cuente con acelerómetro.

# Incluir Botones en pantalla para Juegos Android

Bueno ya se usar el acelerómetro, y eso resulta excelente para un juego, pero ¿Qué pasa si quiero usar un control y botones para mi juego? Como la mayoría de los juegos para Android, así lo tienen.

Tal vez pienses que se trate de una tarea muy laboriosa y desgastante, pero no, resulta una tarea muy sencilla que con un poco de práctica te convertirás en un experto o experta creando controles para juegos android.

La problemática surge porque los teléfonos de ahora no cuentan con teclados físicos, sino que son virtuales, de igual forma cuentan con una característica que se llama multitouch, esto quiere decir que puedes elegir presionar varias partes de la pantalla al mismo tiempo o secuencias y la tarea programada se realiza.

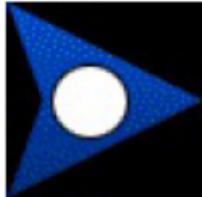
Y la solución a esta problemática es la siguiente.

Primero tengo dos botones, uno que tiene la flecha hacia arriba y que indica que es para dar fuerza al motor del avión y para que no pierda altura.



Y Otro para hacer que el avión se mueva hacia

derecha.



Los voy a adherir a la pestaña Media



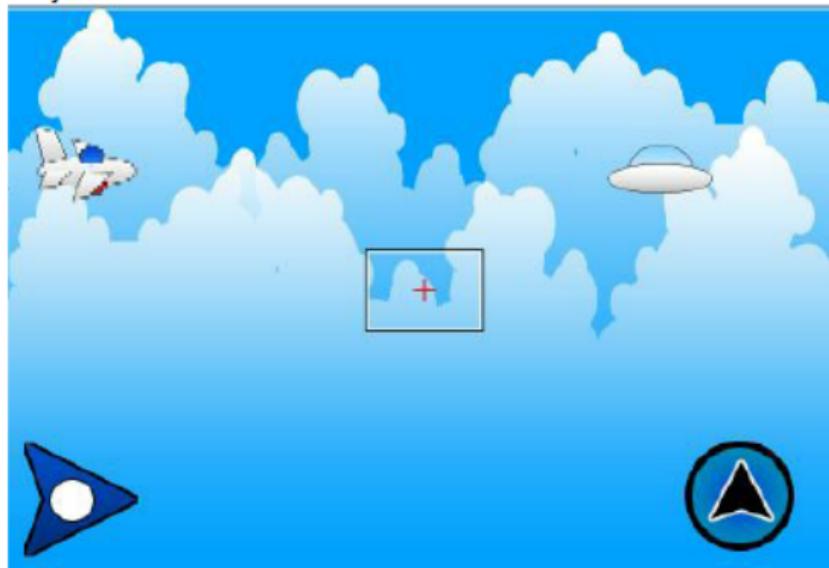
botonderecho

64x64

botonsaltar

64x64

Estos dos botones los adhiero a la escena del juego, porque al fin y al cabo es ahí en donde vamos a interactuar con ellos.



Estos dos botones los voy a adherir a una layer que se llame interface, nos movemos a la pestaña Layers



y creamos una nueva layer con el nombre Interface.



**Interface**



Arrastro los actores botón\_derecha y botón\_impulso hacia la layer recién creada (Interface).



**boton\_derecha**



**boton\_impulso**

Así debe quedar, y les cambie el nombre a los actores, pero es posible que si no le cambias el nombre te puedan aparecer como Actor6 y Actor7. Deseleccionamos Scrollable en la layer Interface, para que los botones se mantengan en la pantalla.



**Interface**



**boton\_derecha**



**boton\_impulso**

Vamos a la pestaña Game Attributes



**Game**

y agregamos dos atributos del tipo booleano, uno se llama derecha y el otro impulso.

derecha



boolean

impulso



boolean

Los dejamos tal cual.

Seleccionamos el actor botón\_derecha y vamos al backstage, y agregamos una regla, ponemos un behavior



Touch

para que podamos interactuar con la pantalla del dispositivo con Android.

The image shows the backstage rule editor in Scratch. At the top, there's a dropdown menu with 'On' selected. Below it, the word 'Rule' is displayed. The main area starts with the condition 'When All of the following are happening'. Underneath this, a specific condition 'touch is: pressed' is selected. A large, rounded rectangular button below these conditions contains the placeholder text 'type or drag in a condition'. Below this button, there are two options: 'do' and 'else', each preceded by a right-pointing arrowhead.

En do vamos a poner un behavior



Change Attribute

para cambiar el atributo *derecha* que se encuentra en la pestaña



# Game

▼ do

▼ On Change Attribute

set: game.derecha a to: true e

Cada vez que presionemos la imagen del botón derecha el atributo *derecha* se volverá verdadero.

A Continuación vamos a seleccionar el actor avión y vamos a adherir una nueva regla en la cual vamos a adherir un behavior



Attribute para consultar el atributo *derecha*.

▼ On Rule

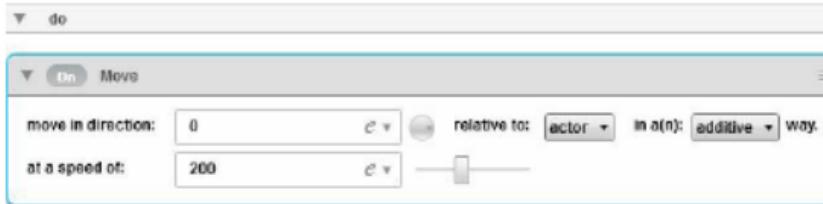
When All of the following are happening

if game.derecha a is true

En do colocamos un behavior



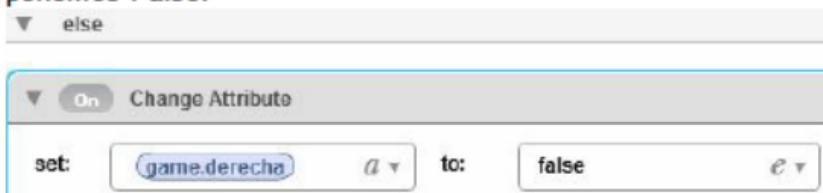
Move para que el avion se mueva hacia la derecha al presionar la tecla.



Si reprodujimos el juego

 podemos ver que el avión se mueve hacia la derecha, pero no se detiene cuando soltamos el botón, esto se debe a que el parámetro *derecha* no regresa a su estado inicial. Para corregir este problema vamos nuevamente a seleccionar el actor *botón\_derecha* y agregamos un behavior

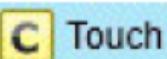
 **Change Attribute** pero esta vez en Else, y ponemos los mismos valores, nada más que en to: en lugar de true ponemos False.



Reproducimos nuevamente el juego

 y ahora el avión se mueve solamente cuando el botón en pantalla es presionado, y si no el avión se queda fijo sin moverse. Ya hemos colocado el botón y si lo verificas está trabajando correctamente.

Ahora vamos con el botón de impulso, hacemos lo mismo, seleccionamos el actor *botón\_impulso* y adherimos una nueva



regla. En donde colocamos un behavior **C Touch** para el toque de pantalla sobre el botón.

On Rule  
When All of the following are happening  
touch is: pressed

Ahora ponemos un behavior

**A Change Attribute** en do que cambiar el valor de la variable o atributo *impulso*.

On Do Change Attribute  
set game.impulso to: true

En Else ponemos false en true.

On Else Change Attribute  
set game.impulso to: false

Seleccionamos el actor avión nuevamente. Y adherimos una nueva regla o bien copiamos la que hicimos anteriormente para

el botón de moverse a la derecha.

The screenshot shows a Scratch script. At the top, there's a button labeled "On" with a dropdown menu set to "impulso con boton en pantalla". Below it, a "When" hat block has a dropdown menu set to "All" with another dropdown menu below it. The main script area contains an "if" control block with a condition "game.impulso" followed by a dropdown menu set to "a", and an "is true" dropdown menu.

En dónde debemos cambiar el valor move in dirección a 90. Como se muestra en la imagen.

The screenshot shows the "Move" block settings. It includes fields for "move in direction" (set to 90), "relative to" (set to "actor"), "in a(n)" (set to "additive"), and "way". There are also fields for "at a speed of" (set to 200) and a slider for speed.

De esta forma si presionamos en el botón impulso el avión ahora debe ir para arriba. Y con esto puedes ver que fácil resultado crear un videojuego con Gamesalad para Sistemas Operativos Android.

Hicimos primero que el avión se moviera de izquierda a derecha y viceversa, de igual forma adherimos gravedad al juego, aprendimos a usar las teclas de la computadora, vimos que importante es conocer las variables en gamesalad, creamos una jugador que realiza disparos y si impacta a un enemigo ovni, este será destruido, aprendimos a animar al actor jugador (avión) de forma fácil y sencilla. y muchos trucos que sin duda te ayudaran a crear un excelente juego, ya sea de plataformas, carreras, puzzles, aventuras, acción, etc.

Un juego básicamente se basa en este tipo de interacciones, por ejemplo si deseas crear un juego de plataformas, simplemente sigue los pasos de colisión, animar, moverse, saltar.

Un salto se realiza sobre un eje Y y con cierta velocidad en X todo es cuestión de visualizar que deseas realizar y experimentalo con Gamesalad, lo más importante para hacer juegos con gamesalad, es aprender las coordenadas X, Y si eres un excelente observador y observadora, te podrás dar cuenta que casi todo se realiza sobre estos dos ejes.

Un hecho es que ningún libro te puede enseñar a tener imaginación, eso ya depende de ti que es lo que deseas hacer, pero siempre y siempre se original al crear un juego, y el éxito lo tendrás garantizado.

Como ves hemos cubierto 18 horas sino es que menos para te vuelvas un experto usando este gran Software que se llama Gamesalad, pero lo mejor de todo es que ahora puedes sorprender a tus amigos, familiares y hasta gente extraña con los juegos que realices.

Gracias por leer este libro y prestarle atención, te deseo mucho éxito en el desarrollo de tus juegos, tal vez en unos días hasta yo mismo me encuentre descargando algún juego que tu hayas desarrollado, apoyándote en este libro.