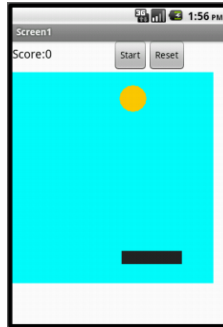




App Inventor

Pong es un juego sencillo, que consiste en una paleta (que es controlada por el usuario) y una pelota. La pelota rebota en la paleta y tres paredes. Si la bola toca la pala, suben la puntuación del usuario, y si la bola pasa por el lateral de la paleta, el juego ha terminado.



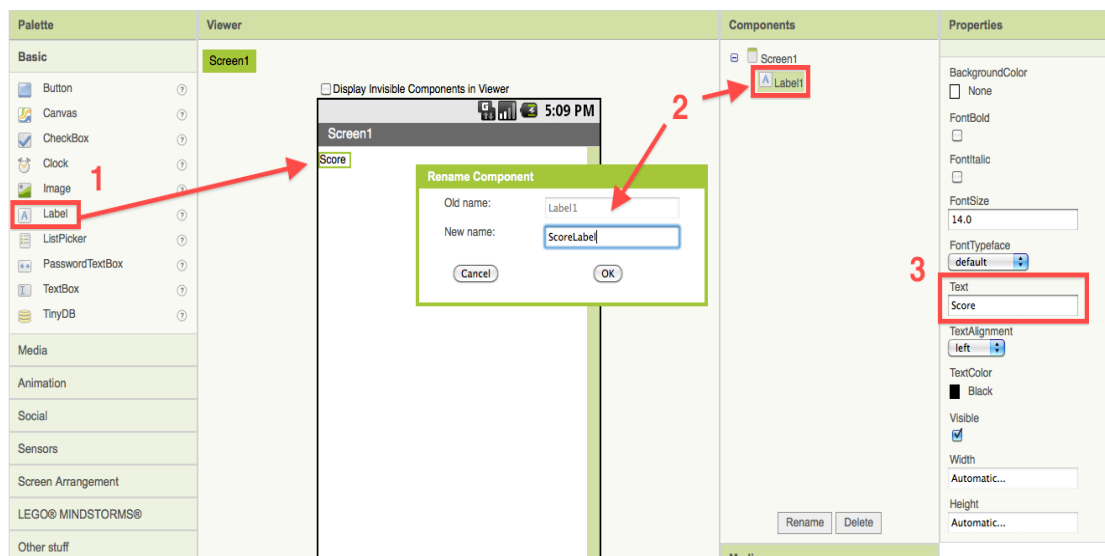
Primera parte. DISEÑO: App Inventor Designer

1.- Abre el AppInventor 2.

2.- Descargar y guardar una imagen de una paleta (paddle.gif ó .png):



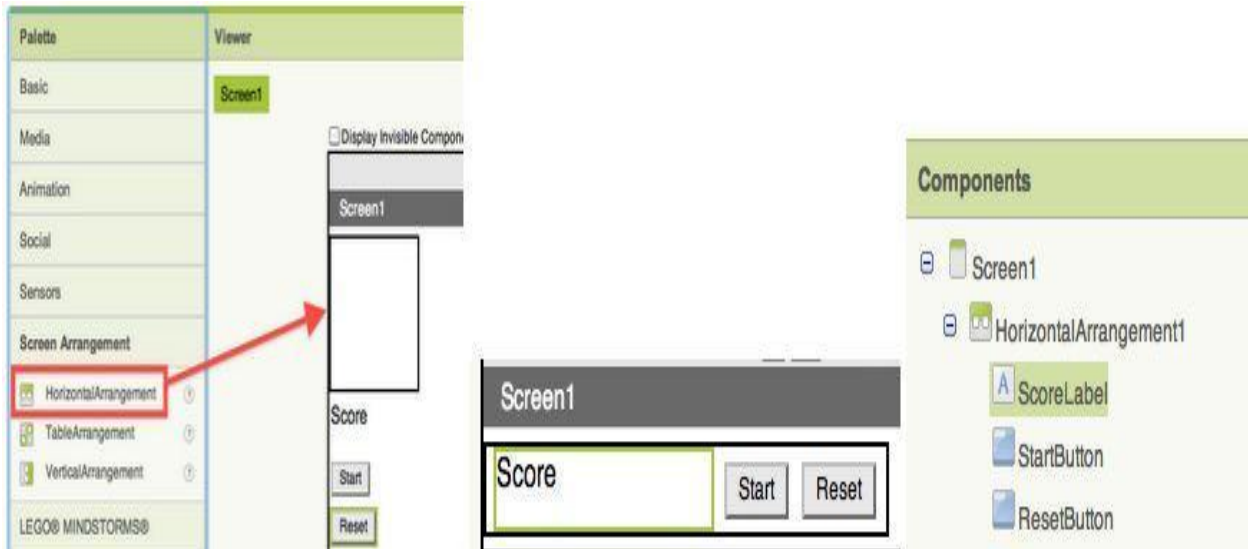
3.- En la columna izquierda del **Designer**, abre la paleta **Basic** y arrastra primero un componente **Label** a la pantalla (1). En el panel **Components**, selecciona el componente **Label1** y cambia su nombre a "**ScoreLabel**" (2). En el panel **Propiedades** (lado derecho), elimina el texto de la pantalla del componente **Label1** para reemplazar por "**Score**" (3) y cambia su tamaño a **18**, su anchura a **150 píxeles** y la altura a **30 píxeles**.





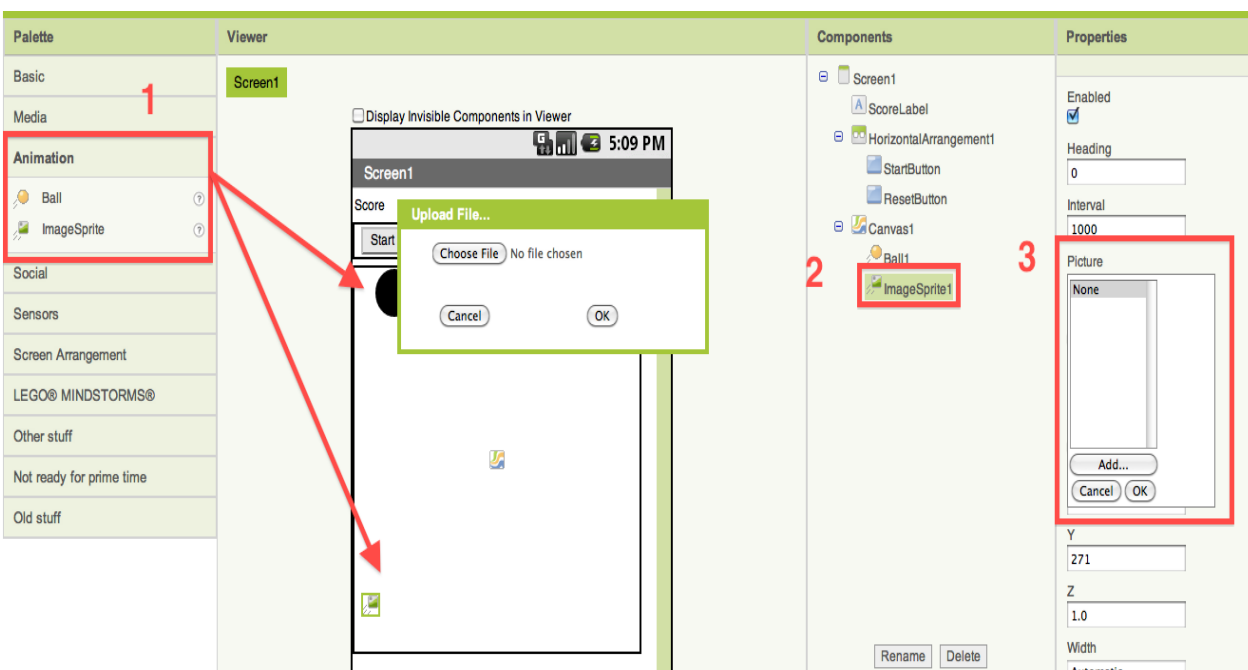
4.- Desde la paleta **Basic**, arrastra **dos botones** a la pantalla debajo de la etiqueta. Cambia el nombre del primer botón a "**StartButton**" y cambia su campo de texto a "**Start**". Cambia el nombre del segundo botón a "**ResetButton**" y cambia su campo de texto a "**Reset**".

5.- De la paleta **Screen Arrangement**, arrastra un componente **HorizontalArrangement** a la pantalla. Arrastra el **ScoreLabel** al **HorizontalArrangement** primero, y a continuación, arrastra los **dos botones** junto al componente **ScoreLabel**.



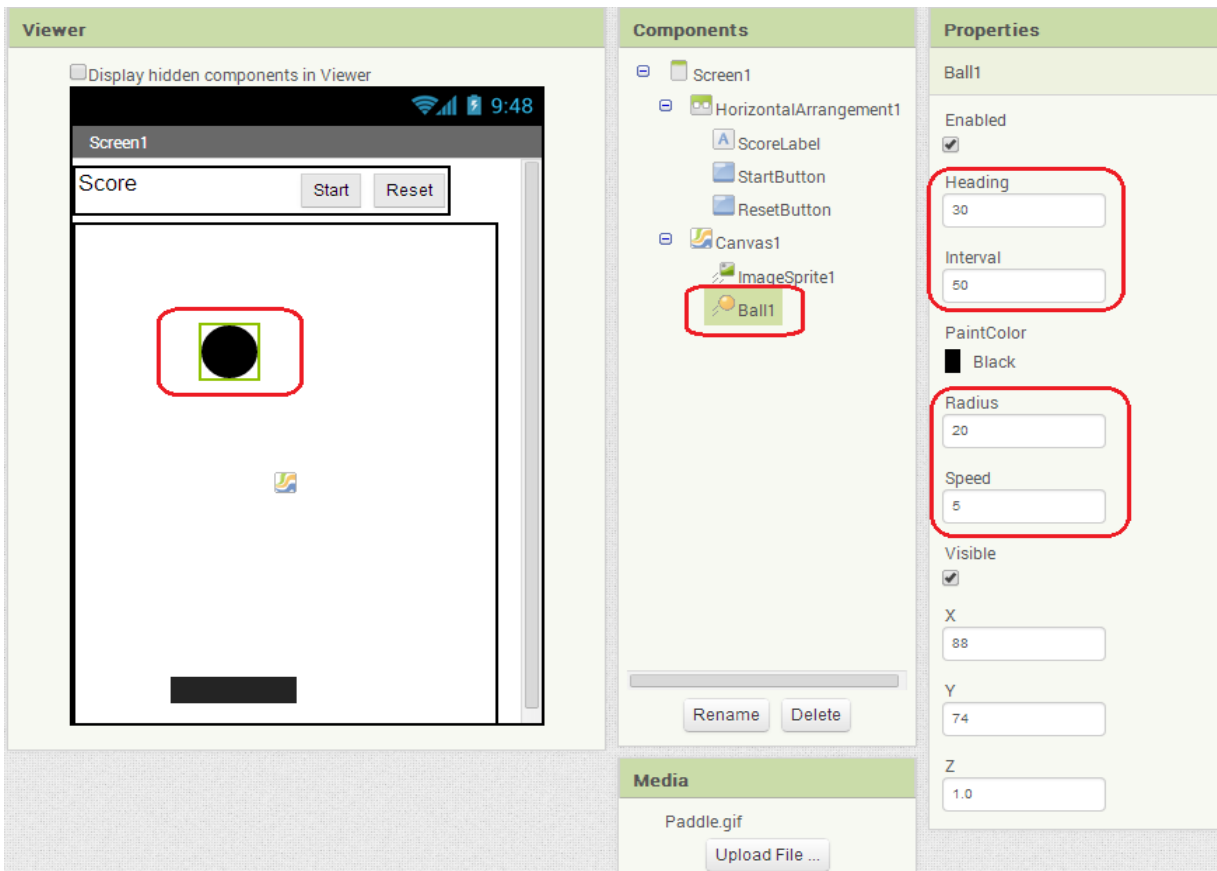
5.- Desde la paleta **Basic**, arrastra un componente **Canvas** y establece el ancho a **300** píxeles y la altura a **390** píxeles en el panel de **Propiedades**. Puedes cambiar el color de fondo del **Canvas** a cualquier color que te guste o incluso poner una imagen de fondo.

6.- En La paleta de **Drawing and Animation**, arrastra un **Ball** y un componente **imageSprite** al **Canvas** en la pantalla **(1)**. Selecciona el componente **imageSprite** **(2)** y cambia su imagen a la paleta **paddle.gif** **(3)**.





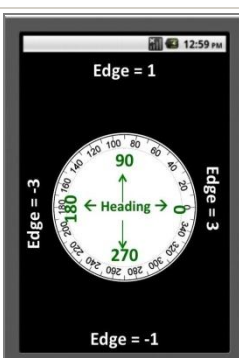
7.- Selecciona **Ball1** en el panel **Components** y en las **Propiedades** establece , el intervalo (**Interval**) a **50**, el radio (**Radius**) a **20**, y la velocidad (**Speed**) a **5**. Puedes cambiar el color de la bola (**PaintColor**) a cualquier color que te guste.



Segunda parte. CONSTRUIR: Editor de bloques

En la esquina superior derecha del **Designer**, haz clic en el botón de **Blocks**. Espera unos momentos mientras los bloques se cargan en el editor.

Para el juego **Pong**, hay tres hechos que debemos conocer :



Quando la bola alcanza una arista. Este evento tiene una propiedad, **borde (edge)**, cuyo valor es el límite que se alcanzó. Si **Edge = 1**, significa que la pelota alcanza elborde inferior.

App Inventor asigna valores numéricos a los bordes de un Canvas de la siguiente manera:

arriba = 1, derecha = 3, abajo = -1, izquierda = -3. También asigna valores de dirección a los objetos animados, se determinan por en un círculo completo, como una brújula, con valores entre 0 y 360 grados, se dice que un objeto que se mueve hacia la parte superior de la pantalla al tener un rumbo o ángulo de 90 grados.





Cuando la bola choca con la pala (o cualquier otra bola). Este evento tiene una propiedad, **otros (other)**, cuyo valor es la **pelota o otros sprites** (paleta) con la que chocó. Para esta aplicación, no nos preocuparemos por el valor de esta propiedad. Siempre será la paleta.

```
when Ball1 .CollidedWith
other
do
```

Cuando se arrastra el ImageSprite. Este evento tiene seis propiedades. Las más importantes para nuestra aplicación son **CurrentX** y **CurrentY**. Éstas representan la ubicación de la paleta cuando el usuario deja de arrastlarla.

```
when ImageSprite1 .Dragged
startX, startY, prevX, prevY, currentX, currentY
do
```

y dos **procedimientos** para las **bolas** y los **sprites** que haremos servir:

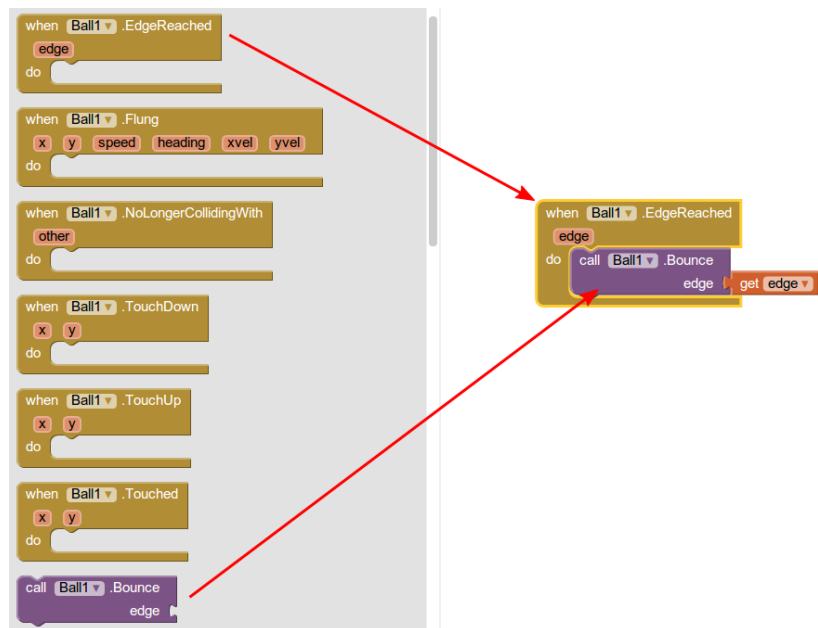
Rebotar la bola. Este procedimiento tiene un conector de **entrada, edge**, que es el borde con el que se desata que la pelota rebote. Teniendo en cuenta el rumbo actual de la bola, rebotará automáticamente con un ángulo fijo establecido por el (360 - ángulo).

```
call Ball1 .Bounce
edge
```

Mover la bola/Sprite a (x, y) dentro del Canvas. Bolas y sprites se pueden mover a cualquier punto en elCanvas. Los **conectores X e Y** representan el **punto (x, y)** donde queremos que se mueva.

```
call ImageSprite1 .MoveTo
x
y
```

1.- Desde la paleta **Screen1**, abre la paleta **Ball1** y arrastra el bloque **when Ball1.EdgeReached** sobre el área de trabajo, así como el bloque llamada **call Ball1.Bounce**. Una vez hecho esto, inserta el **get edge** en el conector del bloque **call Ball1.Bounce** **edge** que está pidiendo un valor **límite del rebote**, es una función integrada dentro de los sprites.



2.- Para controlar el arrastre del usuario de la paleta se utiliza el bloque **when ImageSprite1.Dragged**. Debido a que la pala **sólo puede moverse horizontalmente**, no verticalmente, su **coordenada Y siempre va a ser la misma**. Sin embargo, su **coordenada X** irá cambiado al valor que tiene por su ubicación. Por lo tanto, utilizamos solo un bloque para obtener **CurrentX** en la paleta de coordenadas.

```
when ImageSprite1 .Dragged
startX, startY, prevX, prevY, currentX, currentY
do
  call ImageSprite1 .MoveTo
  x: get currentX
  y: ImageSprite1 . Y
```



3.- Por último, tenemos que controlar las **colisiones** entre la **pelota y la paleta**. Lo que sucede es que la pelota **rebota** en la paleta de partida en la **dirección inversa**. Aquí es donde se utiliza la expresión (360 ángulo) para **invertir la dirección** de la pelota:

```
when Ball1 .CollidedWith  
other  
do  
  set Ball1 . Heading to 360 - Ball1 . Heading
```

Prueba el **funcionamiento de la aplicación**.

Ahora añadiremos más **jugabilidad**:

1.- Para **iniciar el juego**, para comenzar a **moverse**, la **velocidad** de la bola, la **partida**, la **posición** y propiedades de **intervalo** se deben **establecer** y deben estar **habilitadas**.

Ten en cuenta cómo la propiedad del **ángulo del movimiento** se establece en un valor **aleatorio** entre **225** y **315**, esto hará que la pelota se mueva hacia abajo, recordemos que un ángulo de 90 es hacia arriba y 0 es hacia el borde derecho.

Las **propiedades de velocidad y el intervalo** controlan la **velocidad** con la que la **bola** se mueve. En este caso se mueve **5 píxeles cada 10 milisegundos** (cambiaremos en el **Designer** las condiciones iniciales de **intervalo** y **radio** de la bola a **10 milisegundos** y **10 píxeles**).

Finalmente posicionamos la bola en la **parte superior central** del **Canvas**:

```
when ButtonStart .Click  
do  
  set Ball1 . Enabled to true  
  set Ball1 . Interval to 10  
  set Ball1 . Heading to random integer from 225 to 315  
  set Ball1 . Speed to 5  
  call Ball1 . MoveTo  
    x Canvas1 . Width / 2  
    y Ball1 . Radius
```

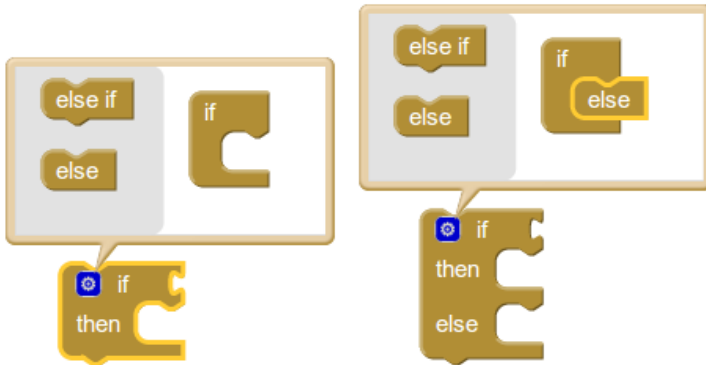
El **botón de reinicio**, simplemente vuelve a colocar la bola en la parte superior central del Canvas

```
when ButtonReset .Click  
do  
  call Ball1 . MoveTo  
    x Canvas1 . Width / 2  
    y Ball1 . Radius
```

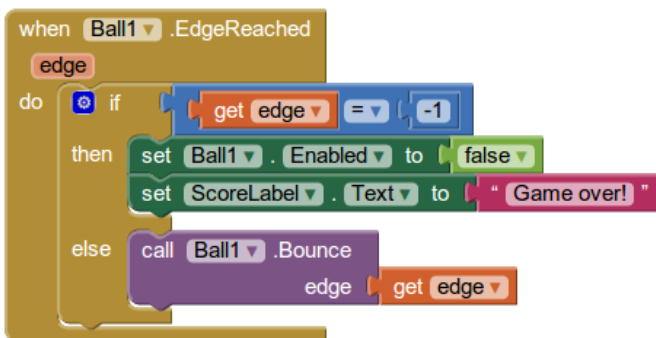
Comprueba el **funcionamiento de los botones**.



Ahora añadiremos la posibilidad de **registrar la puntuación** y la de **perder la partida**. Si no golpeamos con la paleta a la bola, tocará el **borde inferior**, habremos **perdido la partida**. Dentro del bloque **when Ball1.EdgeReached** debemos añadir algunos nuevos bloques para este evento, si toca el borde inferior se comportará de una manera y si toca los laterales o superior de otra:

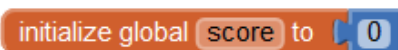


De la paleta de **Matemáticas**, arrastra un bloque **(=) igual** y colócalo en el conector de "if". Arrastra el **get edge** a la primera área en blanco. En la segunda área en blanco "-1" (por el borde inferior). Esto comprobará si el borde al que se llegó es el **borde inferior** del Canvas. Si es el inferior, establecerá la condición de "false" en su **habilitación** para la bola y establecerá en la etiqueta de puntuación un **¡¡ GAME OVER !!**. Si **no es el inferior**, llamará a **mover la bola**:



Comprueba si puedes **perder la partida**.

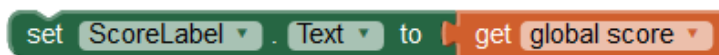
Ahora registraremos la puntuación. Definiremos una **variable " Score"** y la iniciaremos a 0.



Cada vez que golpeemos con la paleta debemos **sumar un punto a la variable** de la puntuación:

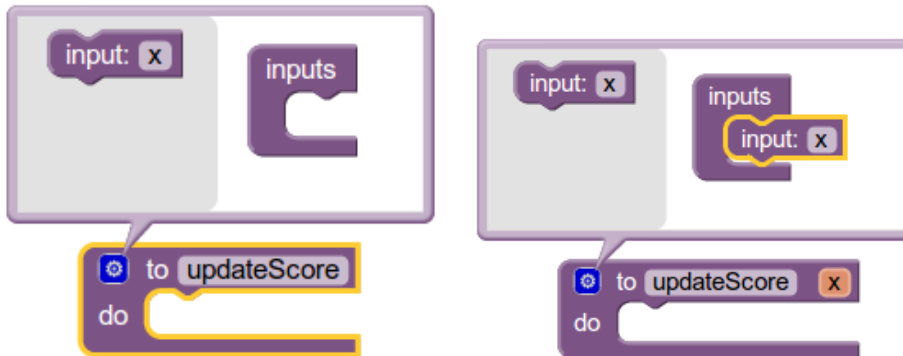


Y hacer que la presente la etiqueta de la puntuación:



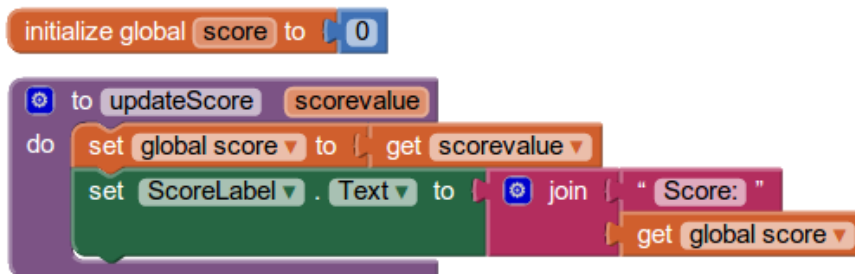


Crearemos un **procedimiento** para actualizar el **valor de la puntuación**, cada vez que sumemos un punto, cada vez que perdamos la partida, cada vez que empecemos de nuevo y cada vez que reiniciemos. Primero generamos el procedimiento, con una variable , con un bloque de **Procedures**:

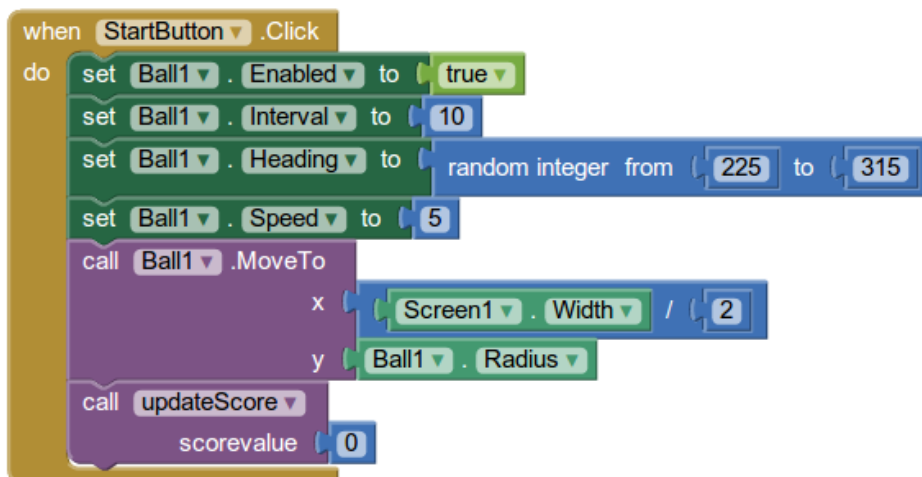


Cambia "x" a "scorevalue". Esto crea una **variable para el procedimiento** que se denomina "scorevalue". Este parámetro es una **variable temporal** que contiene un valor para el procedimiento. Se especifica el **valor** cuando se llama al **procedimiento**. Desde la paleta de **Variables** arrastra un **bloque set** que haga que la variable "score global" adopte el valor que entrega el **procedimiento "updateScore"**.

De la paleta **ScoreLabel**, arrastra un **set ScoreLabel.Text to**, colocaremos un bloque que cambie el valor inicial de "Score: " por el del valor que toma en cada momento más la palabra "Score", con el bloque "join":



Añadiremos la llamada al **procedimiento** en el bloque del botón de inicio **StartButton** y lo iniciaremos



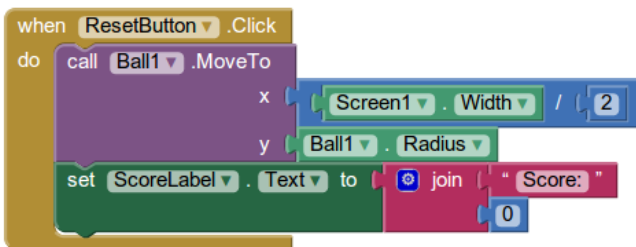
en "0".



También haremos una llamada al **procedimiento** cada vez que la bola **colisione** con la paleta, y sumaremos un **punto** a la puntuación:



Iniciaremos a "0" el marcador cuando activemos el botón de reinicio "ResetButton":



Comprueba el funcionamiento de la aplicación.

Tercera parte: AMPLIACIÓN. Añadir sonido, efectos y más posibilidades

Tendrás que mejorar esta aplicación, debes añadir algunas posibilidades:

1.- Añadir efectos de sonidos.

Si subes sonidos al servidor, añadiendo un componente de sonido Sound en el Designer, puedes hacer que se reproduzcan sonidos cuando se producen eventos como, golpear la bola, rebote de la bola o perder la partida. Deberás añadir estos bloques:



2.- Cuando se llegue a una puntuación determinada puedes cambiar la velocidad de la bola, el tamaño de la bola o de la paleta.

3.- Dar al jugador tres vidas para que puedan obtener tres intentos antes de "Game Over". Si se programan múltiples vidas, disminuir el marcador en 10 cada vez que el jugador pierde una vida.

4.- Investiga lo que sucede si se cambia el rango de números aleatorios para el comienzo de partida cuando se hace clic en el botón de inicio.

5.- Puedes añadir cualquier otro evento que puedes diseñar tú mismo.