

Tutorial Appinventor

Minigolf

Minigolf

Este ejemplo muestra cómo usar el “Fling” (deslizamiento) y los gestos “TouchUp”(tocar) y “TouchDown”(soltar).

Para jugar a este minijuego el jugador primero posiciona su pelota en soporte y entonces puede lanzarla hacia el agujero, la pelota rebotará contra los obstáculos y paredes. Por cada lanzamiento el contador de golpes aumentará. La puntuación total es el número de golpes utilizados.

Parte 1 Crea una nueva App y una pelota que responda a los lanzamientos

Vamos a construir esta aplicación por etapas, añadiendo una funcionalidad del juego cada vez. Accede al App Inventor y crea un nuevo proyecto con el nombre “Minigolf”. Cuando se abre la ventana de diseño observa que AppInventor automáticamente nombra la primera pantalla como “Screen1”, pero puedes cambiarlo.

En la pantalla de Propiedades (panel de la derecha). Desmarca la “Checkbox” llamada “Scrollable” para que la pantalla no se mueva mientras la app está en marcha. Las pantallas en las que se puede hacer “scroll” (deslizar y que se mueva) no tienen una altura determinada. Al no permitir el scroll en nuestra App tendremos que definirla.

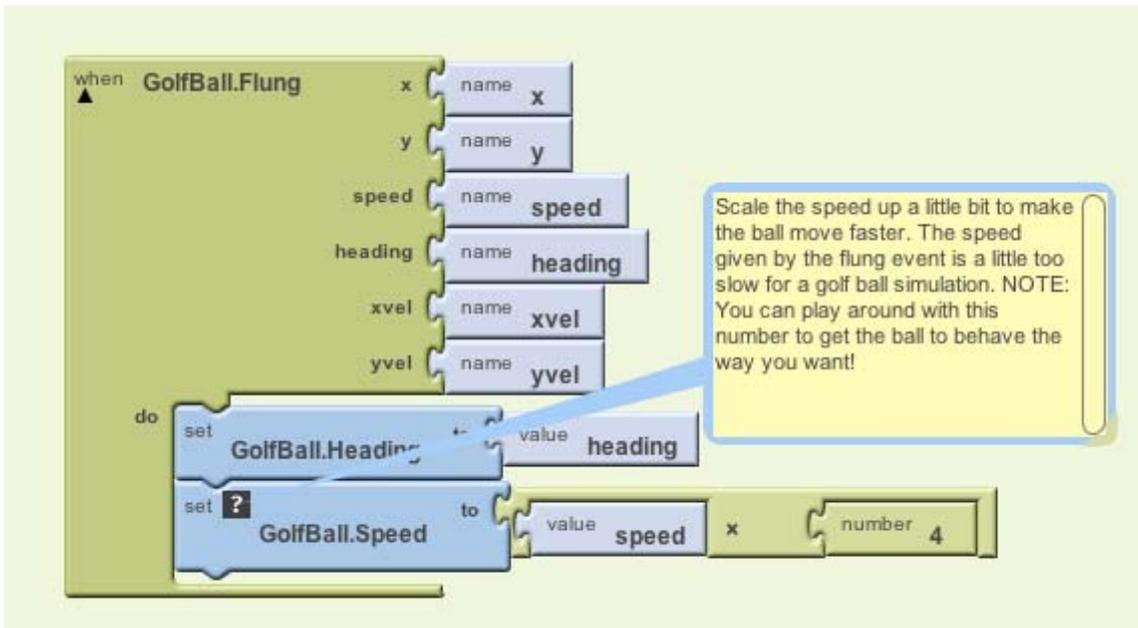
Añade los siguientes componentes en el designer:

Component Type	Palette Group	Nombre	Objetivo	Properties
Canvas1	Basic	Canvas1	El canvas será el campo de golf	Height: 300 Width: FillParent BackgroundColor: Verde o del color que quieras ;)
BallSprite	Animation	GolfBall	Esta es la pelota que hay que meter en el agujero	Radius = 10 Color: Blanco (típico) Speed: 0 Interval: 1 (ms) Z = 2 (así aparecerá por encima del campo z=0)
BallSprite	Animation	Hole	Este sera el agujero	Radius = 15 Color: negro (por ejemplo) Speed: 0
Clock1	Basic	Clock1	Controlará el movimiento de la pelota respecto del tiempo	Timer Always Fires Timer Enabled TimerInterval: 100

Abre el “Blocks Editor”

Programa el comportamiento de la pelota:

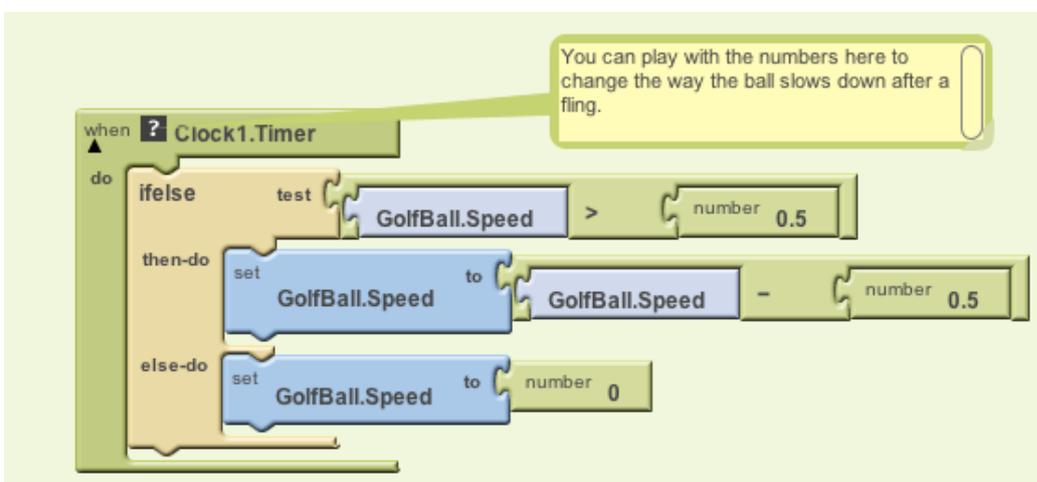
Usa el controlador de eventos GolfBall.Flung “lanzamiento de pelota” para mover la pelota de golf cuando es lanzada. Esencialmente, establece la velocidad de la pelota de golf y la dirige tal cual hayas hecho el gesto de lanzar. Puede ser que quieras aumentar un poco la velocidad ya que la velocidad del lanzamiento es inferior a como una pelota de golf debería moverse (pero depende de gustos). Puedes jugar con GolfBall.Speed.value multiplicándolo por el número que quieras hasta que encuentres la velocidad que más te guste.



Programa el comportamiento del reloj:

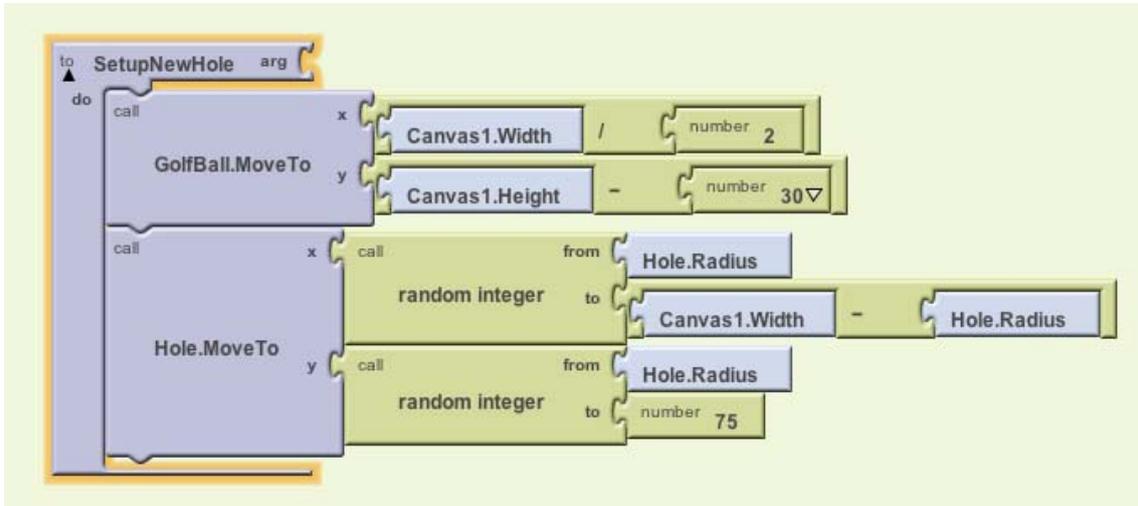
Usa el evento tiempo para reducir la velocidad de la pelota según vaya pasando el tiempo.

Cada momento que el reloj se activa reducirá un poco la velocidad de la pelota. Si la pelota no se está moviendo estos bloques no harán nada. Si no colocas el reloj y estos bloques la pelota estará moviéndose siempre ya que no habrá nada que la frene.



Programa un nuevo procedimiento llamado "SetupNewHole" (colocar un nuevo agujero):

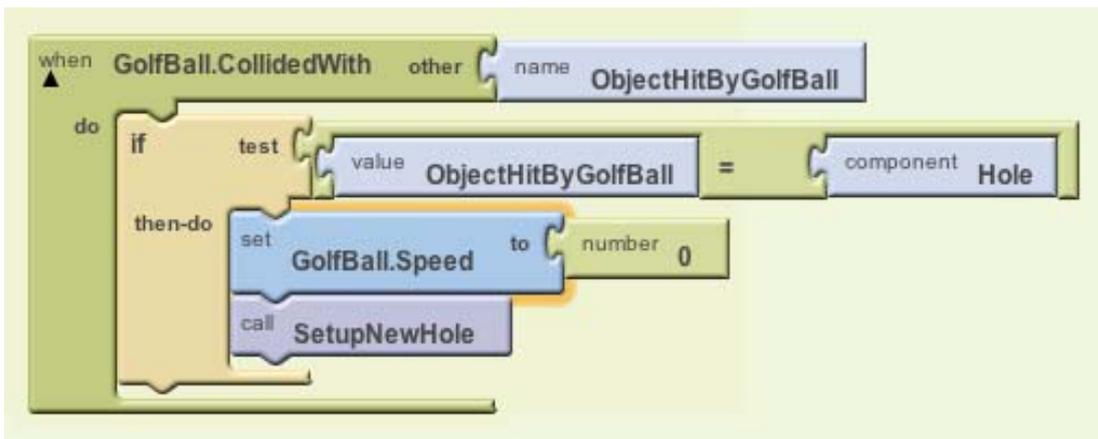
Este procedimiento entrará en acción cuando la pelota entra en el agujero. Su función será que cuando la pelota entre en el agujero devolverá la pelota al inicio y moverá el agujero a otro lugar aleatorio para jugar la siguiente partida.



Programa el comportamiento del Agujero:

Cuando la pelota golpea al agujero reduce la velocidad de la pelota a 0 y "llama" (activa) al bloque de arriba (SetupNewHole)

Nota: cuando arrastras el bloque "GolfBall.CollidedWith", la pieza nombre se llama "other" (otro). Puedes renombrarla clickando en ella y escribiendo el nombre que quieras. En este ejemplo la hemos llamado "ObjectHitByGolfBall" (objeto golpeado por la pelota de golf). Observa que el bloque "if-then" (si pasa xxx entonces haz yyy) comprueba si el elemento que choca con el agujero es la pelota, por eso usa el "component Hole".



¿Se atasca la pelota en el borde?

Esto es fácilmente solucionable con este pequeño bloque. Puedes encontrar el bloque “edge” (borde) debajo de “My definitions”.



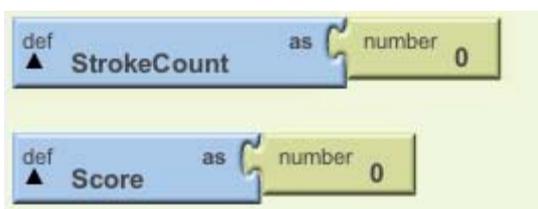
Parte 2: Mantener la puntuación

Los juegos son más divertidos si puedes ver qué tal lo estás haciendo. Por ello, vamos a añadir un contador que aumentará una unidad con cada lanzamiento de la pelota. De forma que el objetivo es conseguir la mínima puntuación. Mostraremos el número de golpes que el jugador tiene en el hoyo que está jugando y también en toda la partida.

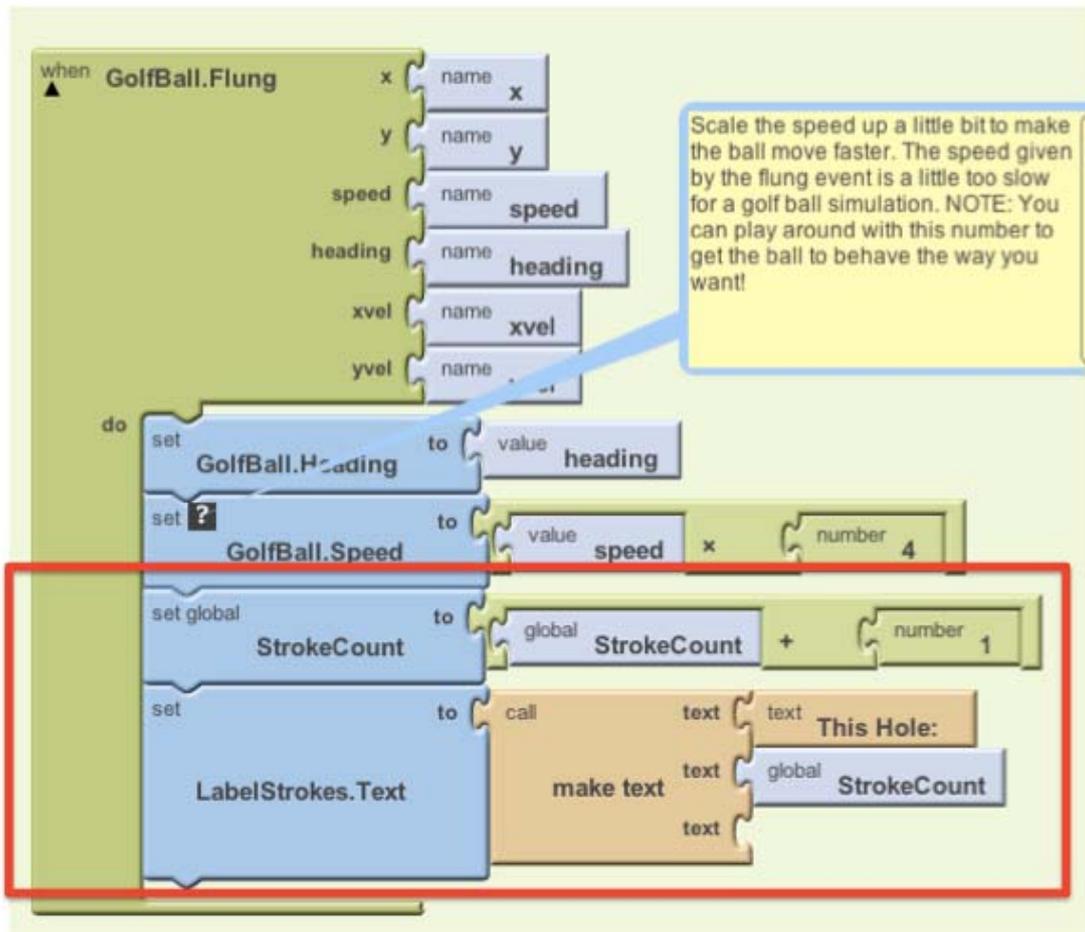
Para ello volveremos al Designer y crearemos los siguientes componentes:

Component Type	Palette Group	Nombre	Objetivo	Properties
Horizontal Arrangement	Screen Arrangement	HorizontalArrangement1	Contiene LabelScore y LabelStroke	Place at top of screen
Label1	Basic	LabelScore	Muestra los golpes de toda la partida	
Label2	Basic	LabelStroke	Muestra los golpes del hoyo actual	

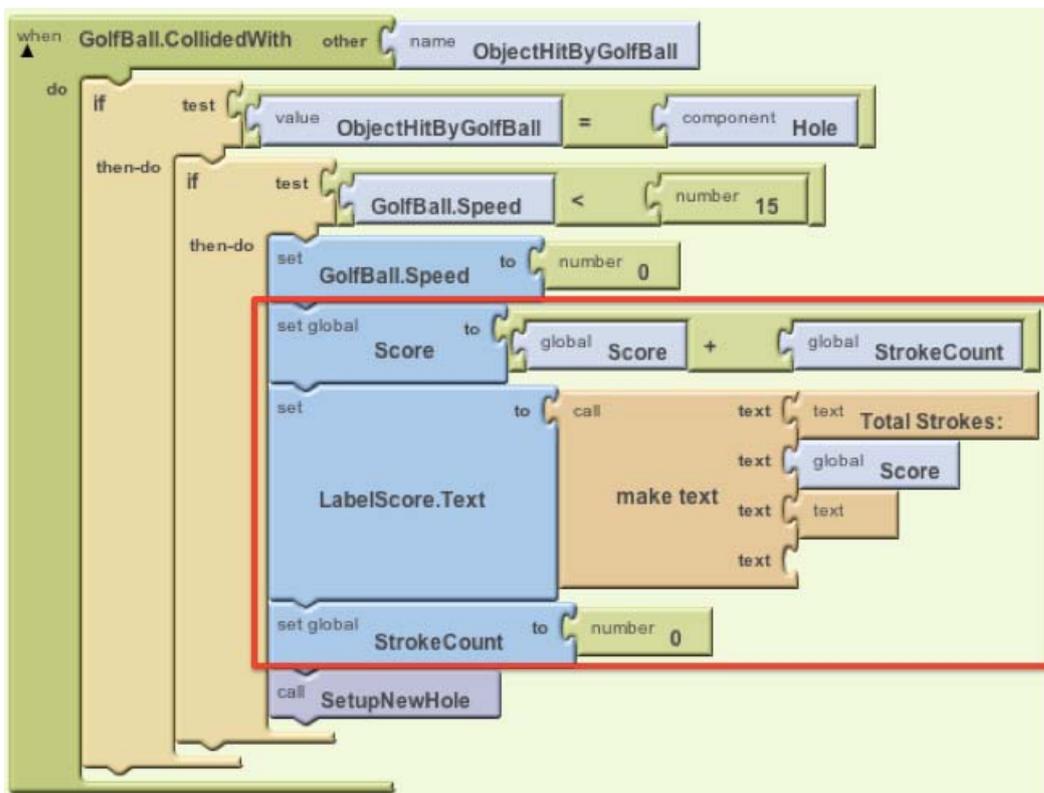
En el editor de bloques podemos programar actualizaciones a la puntuación. Primero, crearemos 2 variables globales llamadas “StrokeCount” (Contador de golpes) y “Score” (puntuación) y las inicializaremos a 0.



Entonces añadiremos al evento “GolfBall.Flung” los bloques que se encuentran recuadrados en rojo.



Y los siguientes bloques al evento que controla cuando la pelota entra en el agujero:



Con estos cambios deberías ver los contadores “Total Strokes” (golpes totales) y “This Hole” (este agujero) en la parte superior de la pantalla. Ambos aumentarán una unidad con cada lanzamiento de la pelota pero el contador “This Hole” se pondrá a 0 cuando la pelota entre en el hoyo.

Parte 3: Posicionar la pelota en el soporte usando los eventos TouchUp y TouchDown

¡Ya tenemos un juego que funciona! Pero ahora vamos a hacerlo más interesante y divertido. Primero colocaremos un soporte y dejaremos al jugador posicionar la pelota de golf en él antes de empezar a lanzarla.

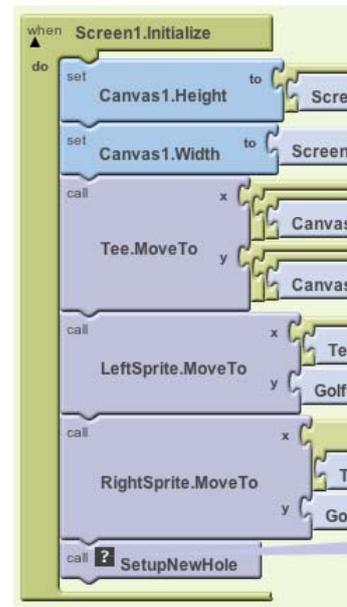
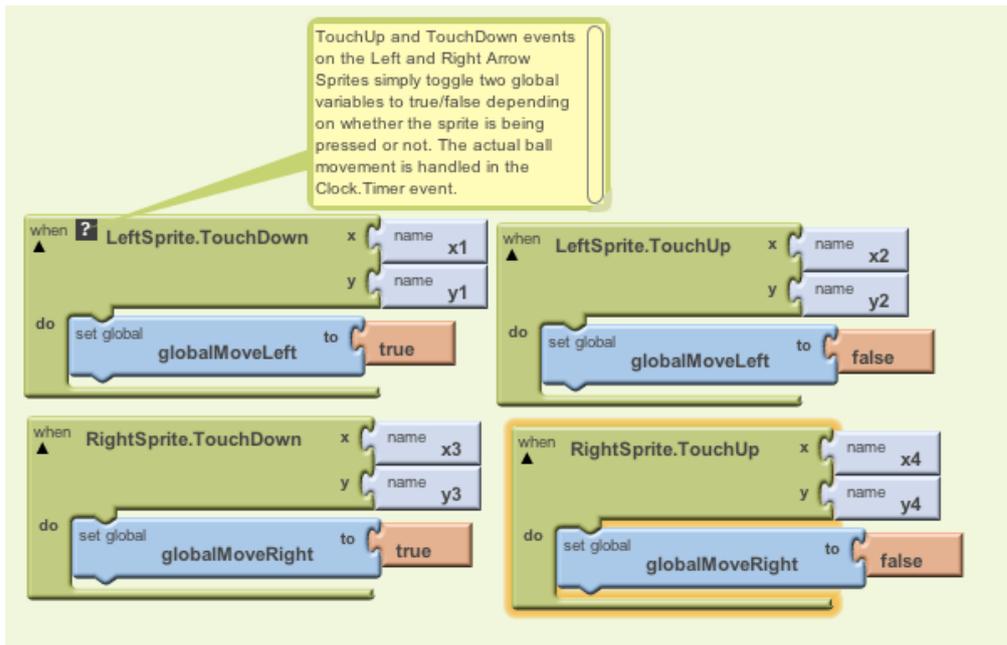
Vuelve al “Designer” y añade tres nuevos componentes:

Component Type	Palette Group	What You’ll Name It	Purpose	Properties
ImageSprite	Animation	Tee	Un area rectangular en la que el jugador puede colocar la pelota antes de lanzarla.	Sube la imagen del soporte (ctrl+click para descargar).
ImageSprite	Animation	LeftSprite	Esta es una flecha que el jugador usará para mover la pelota hacia la izquierda en el “Tee” (soporte).	Sube la imagen de la flecha (ctrl+click para descargar)
ImageSprite	Animation	RightSprite	Esta es una flecha que el jugador usará para mover la pelota hacia la derecha en el “Tee” (soporte).	Sube la imagen de la flecha (ctrl+click para descargar)

Programa el tamaño del canvas y la colocación de la pelota:

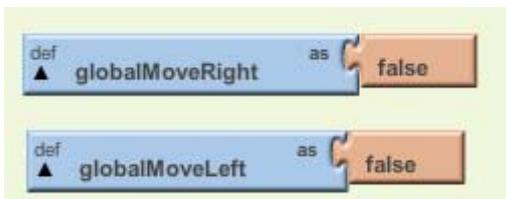
Primero programa la disposición de los componentes anteriores en pantalla. Lo mejor para acomodarse a todos los tamaños de pantalla es colocar los objetos en la pantalla en relación con el tamaño de la pantalla de forma que en pantallas más grandes las imágenes se verán más grandes y si las pantallas son pequeñas las imágenes se verán pequeñas. De forma que siempre se verá bien independientemente del tamaño y resolución de la pantalla.

Los bloques de abajo muestran cómo crear la pantalla dinámicamente para que todo esté donde tenga que estar y tenga el tamaño que tiene que tener. Empezamos haciendo que el canvas sea del tamaño de la pantalla y después colocamos cada objeto en relación al tamaño del canvas. Intenta comprender estos bloques antes de continuar.



Posicionar la pelota de golf en el Tee (soporte) usando TouchUp y TouchDown en las imágenes de las flechas:

Para controlar esto crearemos 2 variables globales más que se activarán cada vez que se pulsa su flecha.

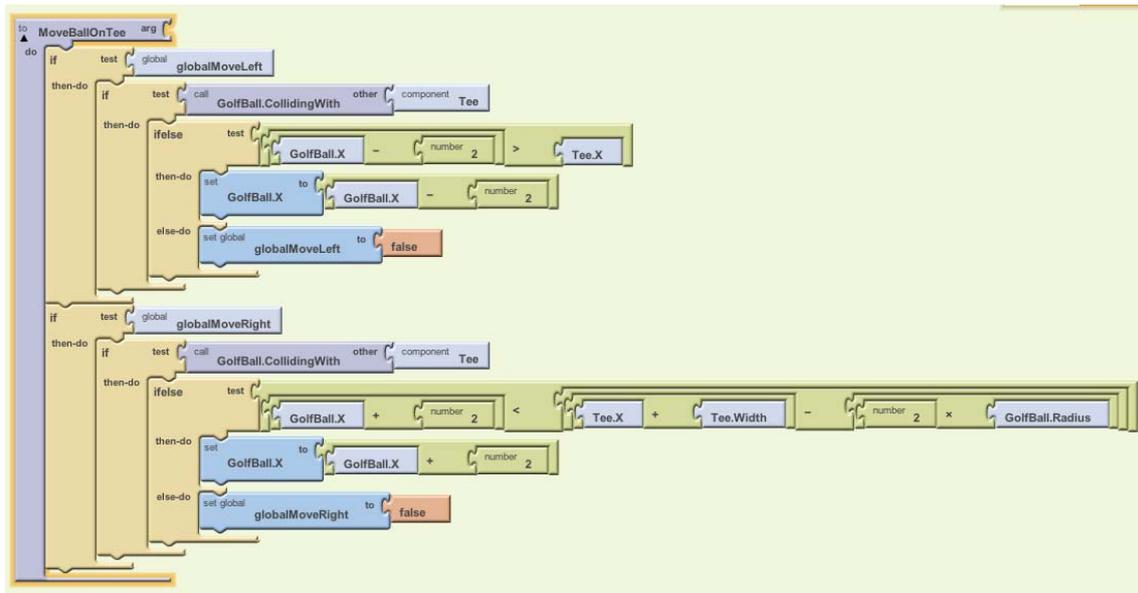


Programar el comportamiento de las flechas izquierda y derecha:

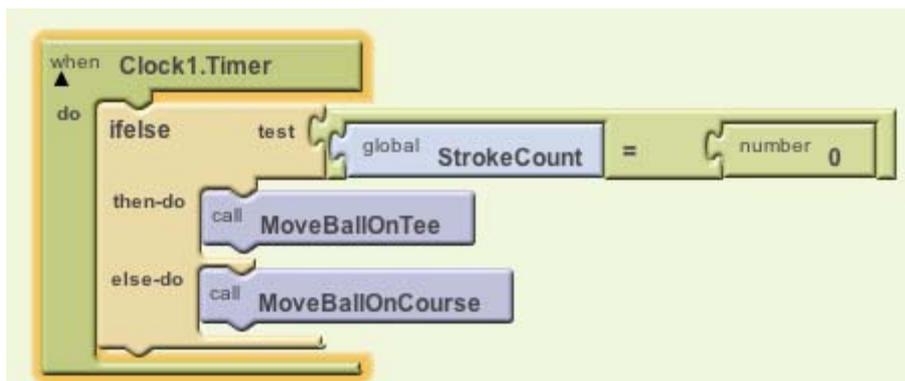
Las flechas izquierda y derecha son “sprites” así que tienen la habilidad de saber cuándo las pulsamos. Los siguientes bloques modifican las variables globales que marcan qué flecha se está pulsando.

Método para mover la pelota en el Tee:

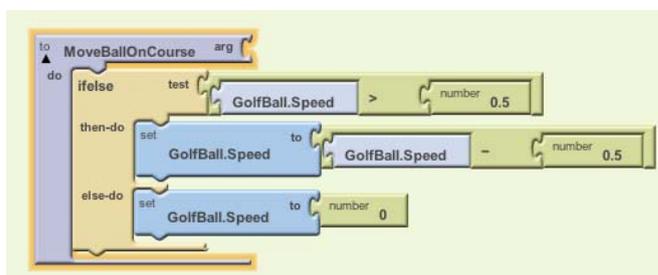
Creo un nuevo “procedure” (método) llamado “MoveBallOnTee” que hace que la pelota de golf se mueva a la derecha o a la izquierda dependiendo de las variables globales definidas anteriormente. Aunque la matemática aquí parezca complicada es simple. Si la pelota tiene que moverse a la izquierda primero hay que comprobar que moviendo la pelota 2 píxeles a la izquierda no se saldrá del “Tee”; y lo mismo si se mueve hacia la derecha.



En cada partida el jugador puede posicionar la pelota en cualquier lugar del “tee” antes de lanzar la pelota. Para programar esto, primero tienes que asegurarte de que la pelota no ha sido lanzada ya. Si el número de lanzamientos = 0 sabemos que se puede mover la pelota.



Los bloques dentro de “Clock1.Timer” se han movido a un nuevo “procedure” (método) llamado “MoveBallOnCourse”.



Permite el “reset” (reinicio) del juego:

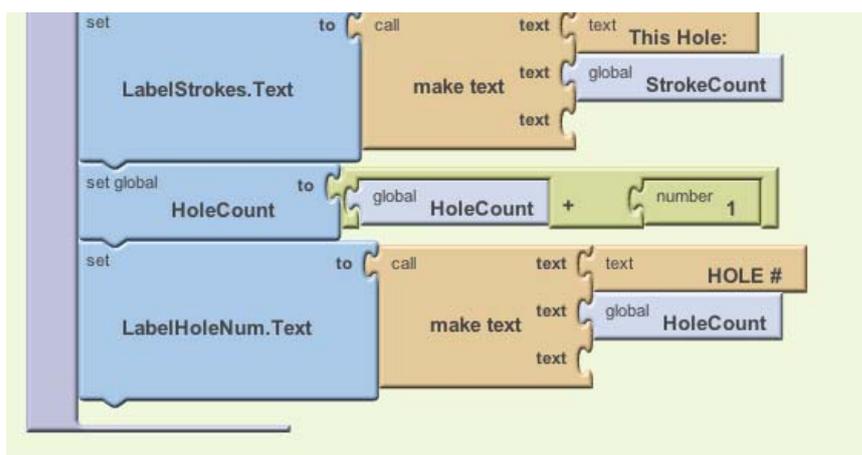
El juego funciona muy bien pero podría mejorar si permitimos que puedan reiniciar el juego. También sería buena idea dar al jugador unas pocas instrucciones para que sepan jugar a nuestro juego. Mientras el usuario juega le daremos una indicación de cuantos hoyos ha completado. Para ello añadiremos los siguientes componentes en el Designer:

Component Type	Palette Group	Nombre	Objetivo	Properties
Horizontal Arrangement2	Screen Arrangement	Horizontal Arrangement2	Contains the NewGame button and the HoleNum label	
Button	Basic	ButtonNewGame	Resets the game to Hole #1 with a score of 0.	Text: "New Game"
Label3	Basic	LabelHoleNum	Displays the current hole number, increments by one each time a hole is completed.	Text = "Hole # 1" Font: bold, 28, blue
Label4	Basic	LabelInstruct	Displays instructions	Text = "Usa las flechas para colocar la bola en el Tee. Golpea la pelota lanzándola con el dedo."

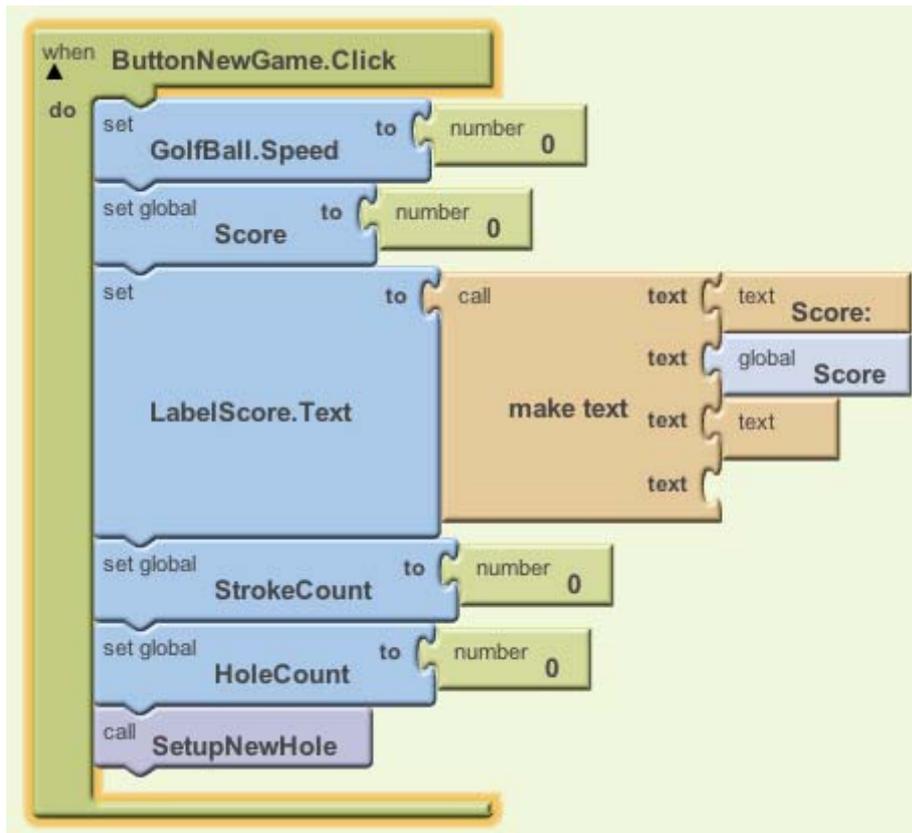
Define una nueva variable global para tener un seguimiento del número de hoyo:



Añade los siguientes bloques al método "SetupNewHole"



Programa el botón "New Game" (juego nuevo), el cual es muy simple. Cuando se pulsa el botón tenemos que resetear un montón de variables y por último llamaremos a "SetupNewHole" (para que genere un nivel aleatorio).



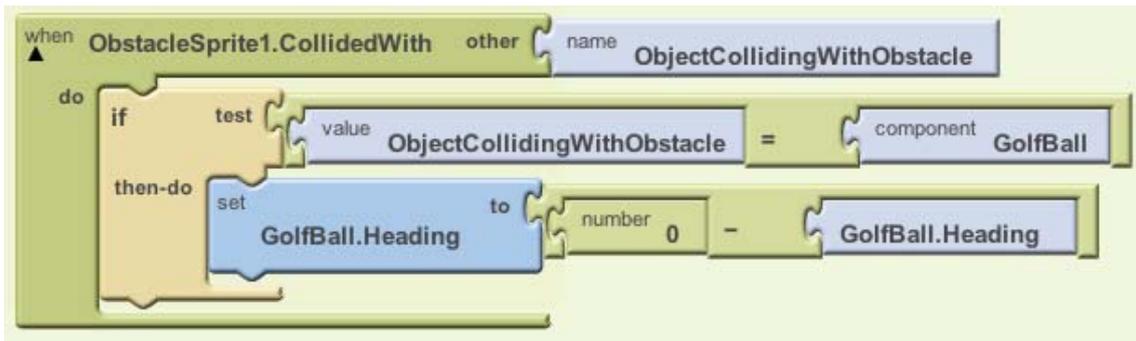
Parte 4: Introduce un obstáculo

La mayoría de minigolfs tienen obstáculos y además hacen el juego más divertido. Vamos a añadir un obstáculo rectangular simple que se colocará aleatoriamente en algún lugar entre el Tee y el hoyo. Cada partida el obstáculo se moverá (al igual que lo hace el hoyo).

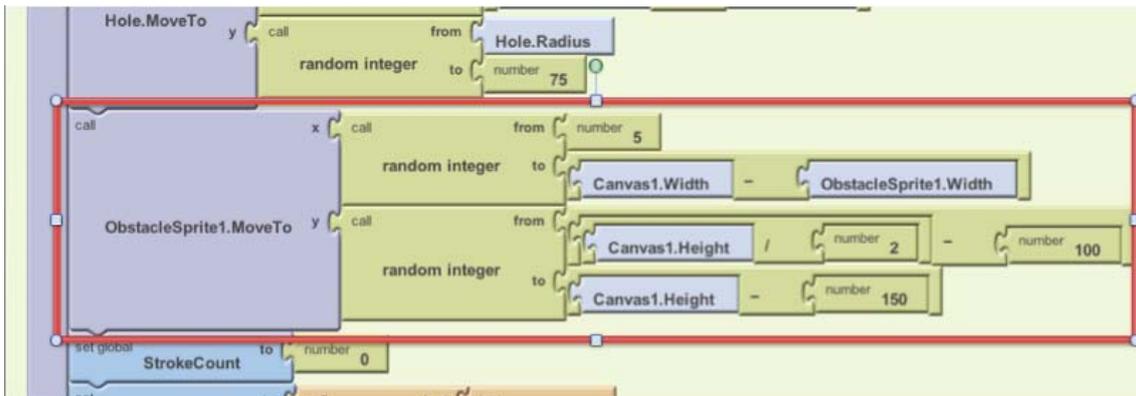
Añade el siguiente componente en el Designer:

Component Type	Palette Group	Nombre	Objetivo	Properties
ImageSprite	Animation	ObstacleSprite1	Esta imagen estará en cualquier lugar entre la pelota de golf y el hoyo. Su objetivo sera hacer más difícil que lleguen al hoyo.	Sube el obstáculo (rectángulo) (ctrl+click)

Programa el comportamiento del obstáculo en el blocks editor. Primero, crea el procedimiento por el cual cuando la pelota golpee al obstáculo. Para ello modificaremos la dirección de la pelota de la inicial a 0-la inicial para que siempre rebote de una forma natural.



Cada vez que se resetee el juego el obstáculo se colocará aleatoriamente. Añade este bloque al “procedure” (método) SetupNewHole:



Y ya está ;). Comparte la apk con tus amigos y prueba el juego para ver que todo funciona como tiene que funcionar. En este juego se utiliza muchísimo la aleatoriedad, es algo normal para que que el juego siempre sea diferente y los jugadores no se aburran así que intenta introducirla en tus juegos.

Parte 5: retos

Aquí hay 3 retos que pueden hacer tu juego mejor.

Reto 1: programa la colisión entre la bola y el hoyo para que la bola solo entre si lleva poca velocidad (como en la vida real).

Reto 2: Hay un pequeño “bug” (error) cuando la pelota golpea los lados verticales del obstáculo. Piensa en cómo solucionar este problema para que la pelota rebote en la dirección esperada.

Reto 3: Limita el número de hoyos por juego. Mantén la cuenta del número de hoyos y acaba el juego después de un número. (Un minigolf típico tiene 18 hoyos).

Resumen:

El proyecto al final queda tal que así:

A la derecha podemos ver el panel de Componentes

Abajo podemos ver el Blocks Editor resultante

