

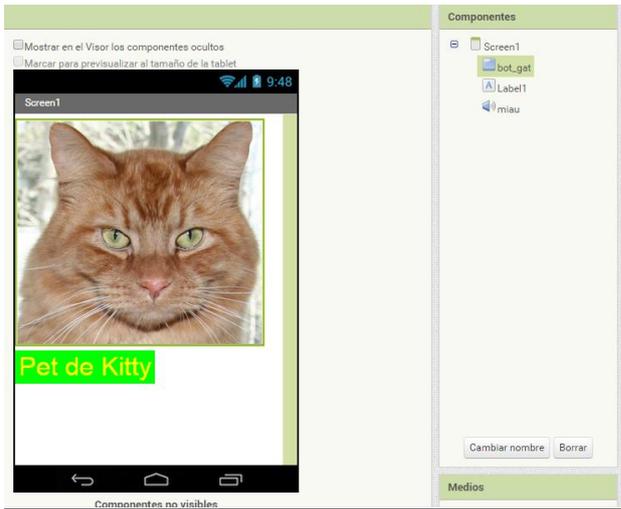
ACTIVITAT 1: PRÉMER SOBRE UN BOTÓ EMET UN SO

COMPONENTS: Botó, etiqueta, imatge,so

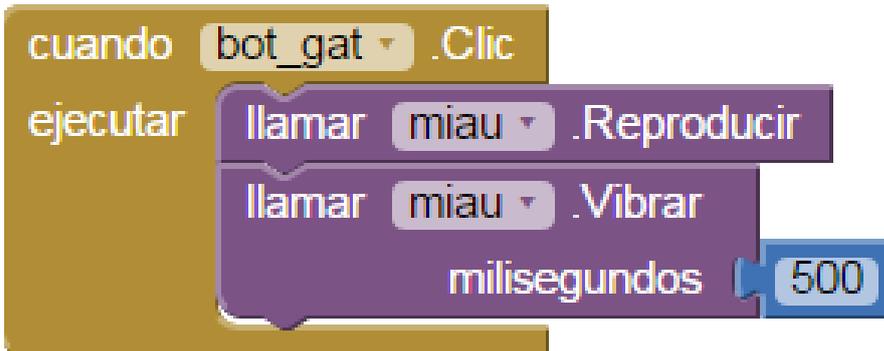
ESDEVENIMENT: Al prémer sobre el botó s'emeta un so.

PROPIETATS DELS COMPONENTS: 500 milisegons de so i vibració del telèfon de 500 ms.

DISSENY:



BLOCS:



ACTIVITAT 2: PASSAR D'UN CANVAS A UN ALTRE MITJANÇANT UN BOTÓ

COMPONENTS:

3 Screens i en cada screen el següents components:

2 Botons, 1 etiqueta, 1 so

ESDEVENIMENT:

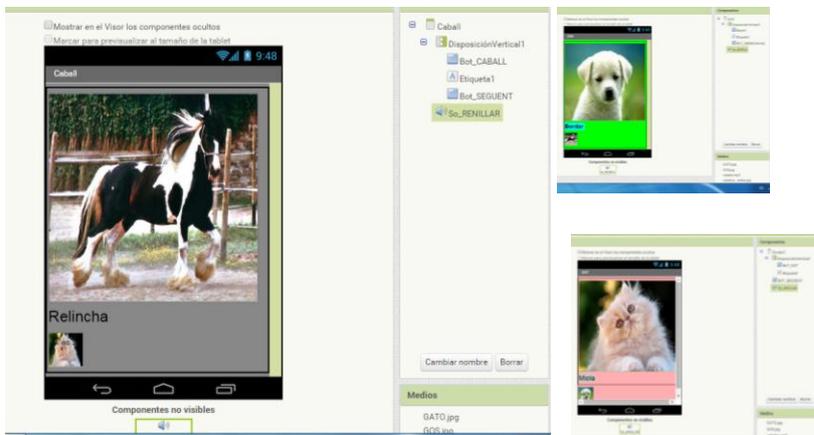
Al prémer el botó imatge emet un so.

Al prémer botó endavant s'obre l'screen següent.

PROPIETATS DELS COMPONENTS:

500 ms de so.

DISSENY:



BLOCS:



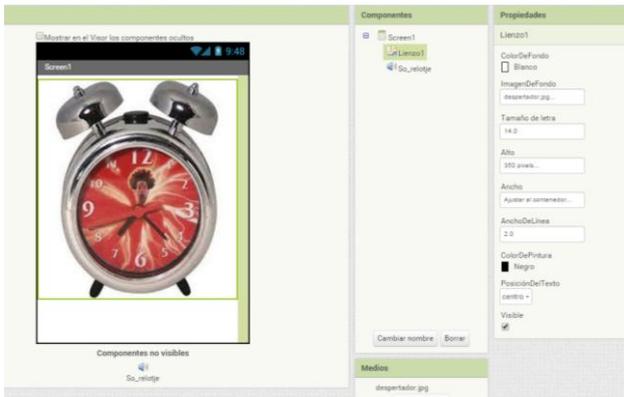
ACTIVITAT 3: PRÉMER SOBRE UNA IMATGE EMET UN SO

COMPONENTS: Canvas, imatge, so

ESDEVENIMENT: Al prémer sobre la imatge s'emeta un so.

PROPIETATS DELS COMPONENTS: 500 milisegons de so i vibració del telèfon de 500 ms.

DISSENY:



BLOCS:



ACTIVITAT 4: SHAKE

COMPONENTS: Canvas, acceleròmetres, so.

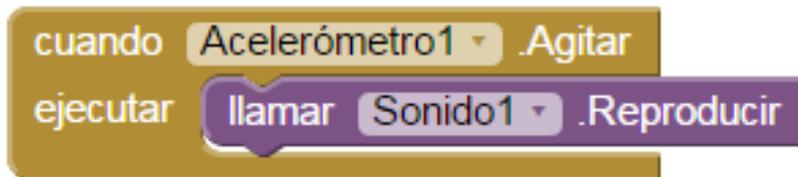
ESDEVENIMENT: Quan sacsejo el mòbil sona un so.

PROPIETATS DELS COMPONENTS: Sensibilitat moderada del acceleròmetre i durant 400 ms

DISSENY:



BLOCS



ACTIVITAT 5: EL JOC DEL TALP

COMPONENTS:

- Canvas
- Spritimatge
- 4 etiquetes
- 2 botons
- rellotge

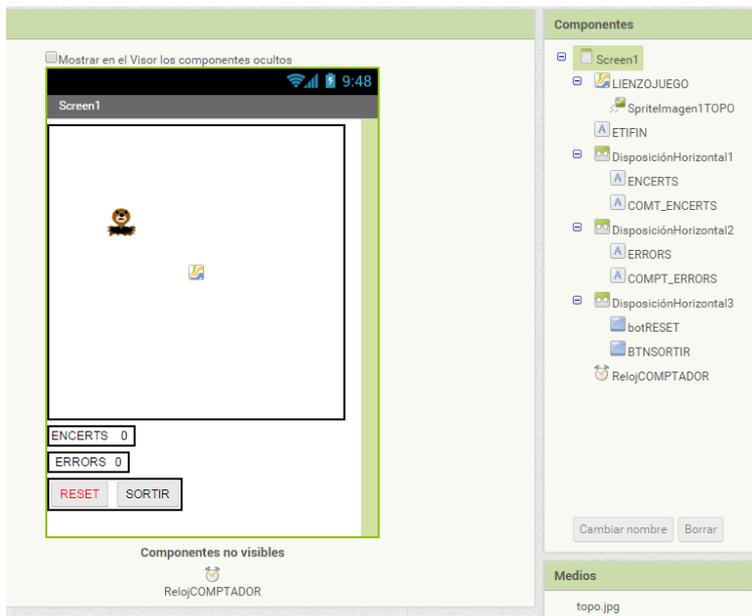
ESDEVENIMENTS:

1. En tocar damunt del talp etiqueta comptador encerts +1.
2. En tocar damunt del talp salta aleatòriament.
3. En tocar damunt del canvas etiqueta comptador errors +1 .
4. En tocar damunt botó reset comptador a 0.
5. Comptador de temps fa que la imatge es desplaci aleatòriament després d'un cert temps.

PROPIETATS DELS COMPONENTS:

Donar mida adequada a tots els components.

DISSENY:



BLOCS.

1.- En tocar damunt del talp etiqueta comptador encerts +1



Comprova el funcionament.

Què passa si prems només a sobre del canvas?

2.- En tocar damunt del canvas etiqueta comptador errors +1



Comprova què funcioni.

Què passa si ara toques el talp? Per què?

Per solucionar-lo hem de fer que quant toquis el canvas, **si** hi ha un spritimatge compte +1, i sinó no. Hem de fer servir una funció if.



Si el valor es cierto, ejecutar el primer bloque de instrucciones. Si no es cierto, ejecutar el segundo bloque de instrucciones.

Comprova.

```

cuando Lienzo1 .Tocar
ejecutar
  si touchedAnySprite
  entonces poner COMPTADOR_ENCERTS . Texto como COMPTADOR_ENCERTS . Texto + 1
  si no poner COMPTADOR_ERRORS . Texto como COMPTADOR_ERRORS . Texto + 1
  
```

3.- En tocar damunt del talp salta aleatòriament.

```

cuando Spritemagen1 .Presionar
ejecutar
  llamar Spritemagen1 .MoverA
  x entero aleatorio entre 0 y Lienzo1 . Ancho - Spritemagen1 . Ancho
  y entero aleatorio entre 0 y Lienzo1 . Alto - Spritemagen1 . Alto
  
```

4.- En tocar damunt botó reset comptador a 0.

```

cuando RESET .Clic
ejecutar
  poner COMPTADOR_ENCERTS . Texto como 0
  poner ERRORS . Texto como 0
  
```

5.- Comptador de temps fa que la imatge es desplaci aleatòriament després d'un cert temps.

```

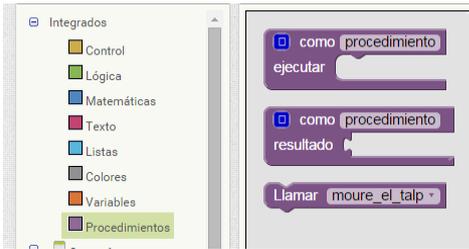
cuando Reloj1 .Temporizador
ejecutar
  llamar Spritemagen1 .MoverA
  x entero aleatorio entre 0 y Lienzo1 . Ancho - Spritemagen1 . Ancho
  y entero aleatorio entre 0 y Lienzo1 . Alto - Spritemagen1 . Alto
  
```

6.- Crear una funció "llamar a..." Procedimiento.

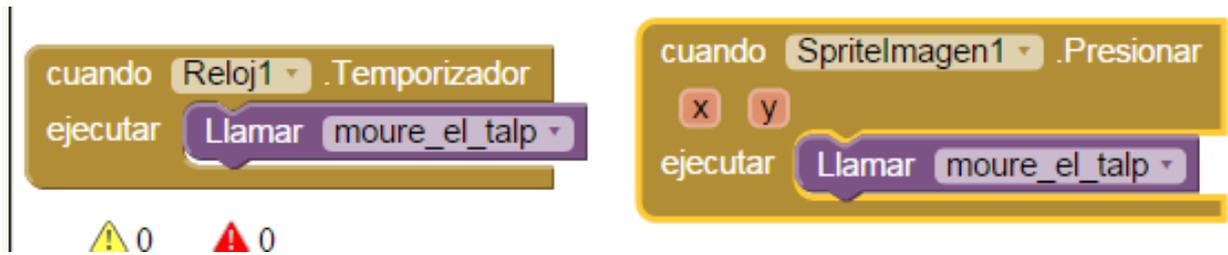
```

como moure_el_talp
ejecutar
  llamar Spritemagen1 .MoverA
  x entero aleatorio entre 0 y Lienzo1 . Ancho - Spritemagen1 . Ancho
  y entero aleatorio entre 0 y Lienzo1 . Alto - Spritemagen1 . Alto
  
```

En procedimientos se ha creat: llamar moure el talp.



Ara podem substituir:



7.- Sortir de l'aplicació.



ACTIVITAT 6: JOC DEL TALP. “ GAME OVER “

COMPONENTS:

- Canvas
- Spritimatge
- 5 etiquetes
- 2 botons
- relotje

ESDEVENIMENTS.

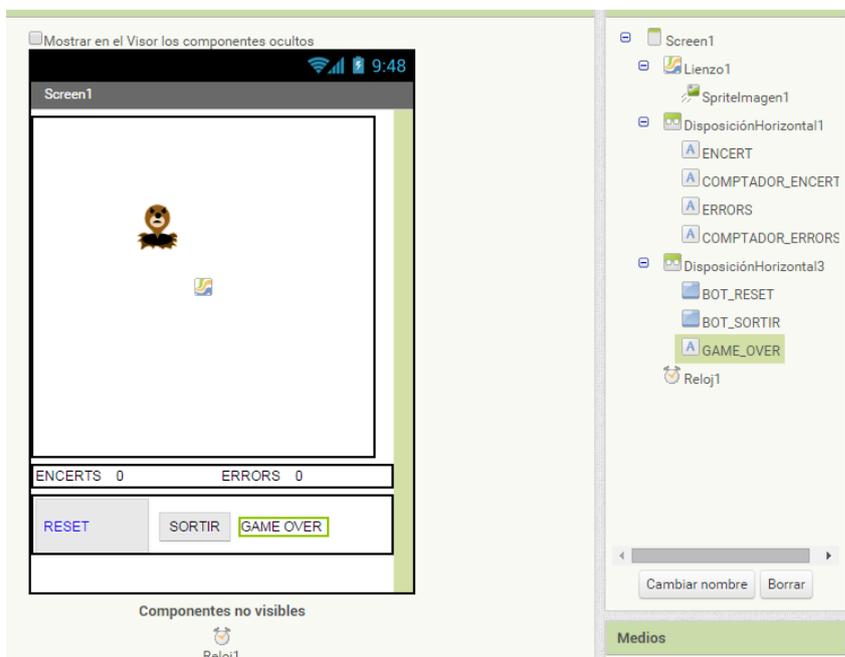
A partir d'un nombre d'errors, el talp desapareix.

A partir d'un nombre d'errors comptadors a 0 i surt un missatge “Game over”

PROPIETATS DEL COMPONENTS

Les mateixes que l'exercici 5.

DISSENY



1.- Afegir una etiqueta amb el nom “Game Over “ i fer-la no visible.

2.- programar que quan el nombre d'errors sigui superior a 10 es faci visible l'etiqueta Game Over.

```

cuando Lienzo1 .Tocar
  x y touchedAnySprite
  ejecutar
    si
      tomar touchedAnySprite
      entonces
        Llamar moure_el_talp
        poner COMPTADOR_ENCERTS . Texto como COMPTADOR_ENCERTS . Texto + 1
      si no
        poner COMPTADOR_ERRORS . Texto como COMPTADOR_ERRORS . Texto + 1
        si
          COMPTADOR_ERRORS . Texto = 10
          entonces
            poner GAME_OVER . Visible como cierto
          si no
            poner GAME_OVER . Visible como falso
  
```

3.- Programar que quan el comptador d'errors sigui 10 el comptador d'encert i d'errors sigui 0.

```

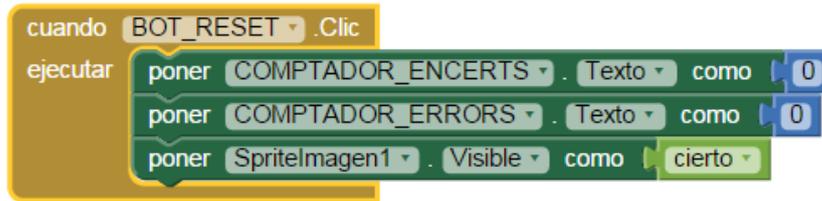
cuando Lienzo1 .Tocar
  x y touchedAnySprite
  ejecutar
    si
      tomar touchedAnySprite
      entonces
        Llamar moure_el_talp
        poner COMPTADOR_ENCERTS . Texto como COMPTADOR_ENCERTS . Texto + 1
      si no
        poner COMPTADOR_ERRORS . Texto como COMPTADOR_ERRORS . Texto + 1
        si
          COMPTADOR_ERRORS . Texto = 10
          entonces
            poner GAME_OVER . Visible como cierto
            poner COMPTADOR_ERRORS . Texto como 0
            poner COMPTADOR_ENCERTS . Texto como 0
          si no
            poner GAME_OVER . Visible como falso
  
```

4.- Programar que quan el comptador d'errors sigui 10 desaparegui el talp.

```

cuando Lienzo1 .Tocar
  x y touchedAnySprite
  ejecutar
    si
      tomar touchedAnySprite
      entonces
        Llamar moure_el_talp
        poner COMPTADOR_ENCERTS . Texto como COMPTADOR_ENCERTS . Texto + 1
      si no
        poner COMPTADOR_ERRORS . Texto como COMPTADOR_ERRORS . Texto + 1
        si
          COMPTADOR_ERRORS . Texto = 10
          entonces
            poner GAME_OVER . Visible como cierto
            poner COMPTADOR_ERRORS . Texto como 0
            poner COMPTADOR_ENCERTS . Texto como 0
            poner Spritelmagen1 . Visible como falso
          si no
            poner GAME_OVER . Visible como falso
  
```

5.- Programar que quan clic botó reset es faci visible el talp.



ACTIVITAT 7:JOC DEL TALP. VIBRACIÓ

COMPONENTS:

- Canvas
- Spritimatge
- 5 etiquetes
- 2 botons
- relotje
- So

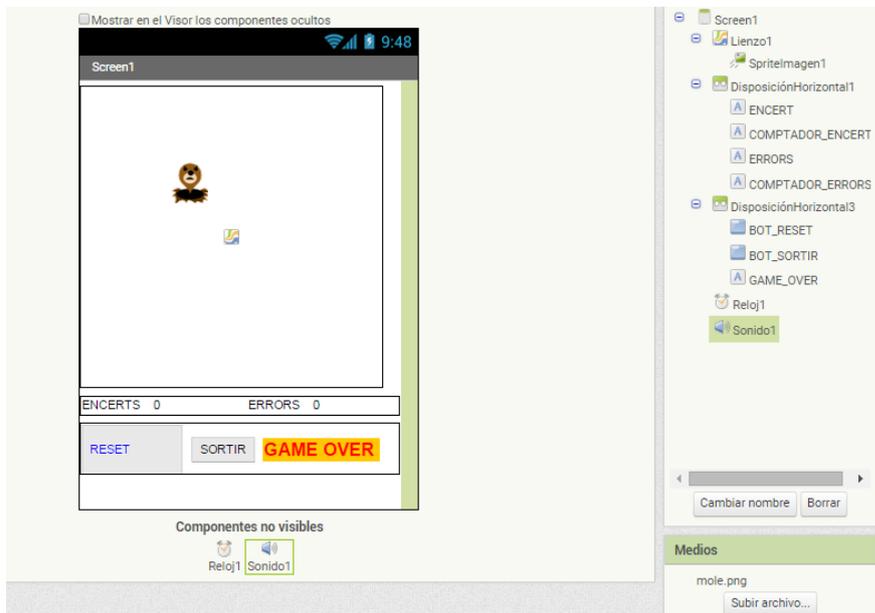
ESDEVENIMENT

Vibra el mòbil.

PROPIETATS DE L'ESDEVENIMENT

Quan toqui a sobre del talp vibra 500 ms.

DISSENY



PROGRAMACIÓ

```

cuando Lienzo1 .Tocar
  x y touchedAnySprite
  ejecutar
    si
      entonces
        tomar touchedAnySprite
        Llamar moure_el_talp
        Llamar Sonido1 .Vibrar
          milisegundos 500
        poner COMPTADOR_ENCERTS . Texto como COMPTADOR_ENCERTS . Texto + 1
      si no
        poner COMPTADOR_ERRORS . Texto como COMPTADOR_ERRORS . Texto + 1
        si
          COMPTADOR_ERRORS . Texto >= 10
          entonces
            poner GAME_OVER . Visible como cierto
            poner COMPTADOR_ERRORS . Texto como 0
            poner COMPTADOR_ENCERTS . Texto como 0
            poner Spritemagen1 . Visible como falso
          si no
            poner GAME_OVER . Visible como falso
  
```

PROGRAMA SENCER

```

como moure_el_talp
  ejecutar
    llamar Spritemagen1 .MoverA
      x entero aleatorio entre 0 y Lienzo1 . Ancho - Spritemagen1 . Ancho
      y entero aleatorio entre 0 y Lienzo1 . Alto - Spritemagen1 . Alto

cuando BOT_SORTIR . Clic
  ejecutar
    cerrar la aplicaci3n

cuando Reloj1 . Temporizador
  ejecutar
    Llamar moure_el_talp

cuando Spritemagen1 . Presionar
  ejecutar
    Llamar moure_el_talp

cuando BOT_RESET . Clic
  ejecutar
    poner COMPTADOR_ENCERTS . Texto como 0
    poner COMPTADOR_ERRORS . Texto como 0
    poner Spritemagen1 . Visible como cierto

cuando Lienzo1 .Tocar
  x y touchedAnySprite
  ejecutar
    si
      entonces
        tomar touchedAnySprite
        Llamar moure_el_talp
        Llamar Sonido1 .Vibrar
          milisegundos 500
        poner COMPTADOR_ENCERTS . Texto como COMPTADOR_ENCERTS . Texto + 1
      si no
        poner COMPTADOR_ERRORS . Texto como COMPTADOR_ERRORS . Texto + 1
        si
          COMPTADOR_ERRORS . Texto >= 10
          entonces
            poner GAME_OVER . Visible como cierto
            poner COMPTADOR_ERRORS . Texto como 0
            poner COMPTADOR_ENCERTS . Texto como 0
            poner Spritemagen1 . Visible como falso
          si no
            poner GAME_OVER . Visible como falso
  
```

ACTIVITAT 8: PAINT AMB VARIABLES

Les variables serveixen per guardar característiques . És una caixa que guarda un valor i l'actualitza.

Les variables tenen nom com, per exemple: mida del pinzell o color del pinzell.

TOMAR és LEER

PONER és ESCRIBIR

COMPONENTS:

1. Canvas
2. 9 botons
3. Càmera de fotos
4. 4 etiquetes

ESDEVENIMENT

En tocar damunt del canvas pintar cercles i línees de diferents colors.

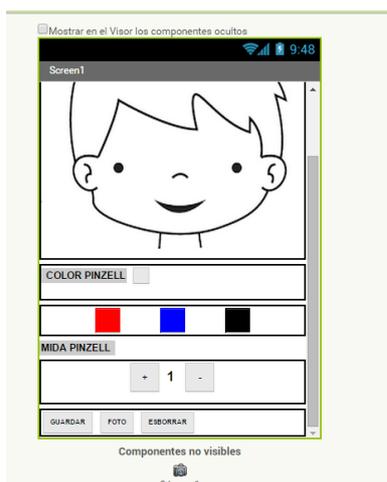
En tocar damunt del botó esborra,netejar el canvas.

En tocar damunt del botó fer foto activar la càmera del mòbil i posar la imatge en el canvas.

PROPIETATS DELS COMPONENTS:

seleccionar el tamany i el color dels botons.

DISSENY:



1.- Crear dues variables:

Una pel tamany del pinzell i l'altre pel color del pinzell.



2.- En tocar damunt del botó “ color? “ pintar damunt del canvas. Això s'ha de fer per cada color.

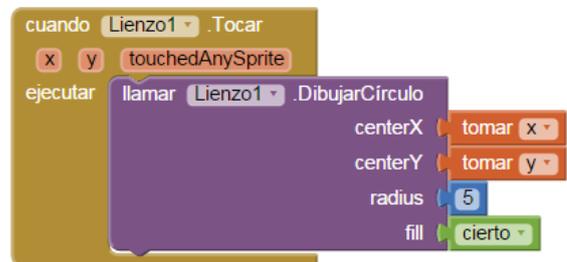


3.- En tocar damunt del botó identificació de color, canvia al mateix color.

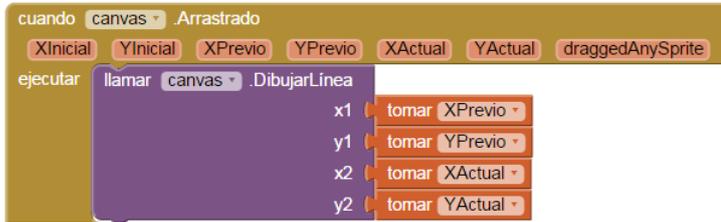


4.- En tocar damunt del canvas que dibuixi cercles de diàmetre 5.

Prova l'app



5.- En arrossegar el dit pel canvas dibuixa línies.



```
cuando canvas .Arrastrado
  ejecutar
    llamar canvas .DibujarLinea
      x1 tomar XPrevio
      y1 tomar YPrevio
      x2 tomar XActual
      y2 tomar YActual
```

Prova la aplicació.

6.- En tocar damunt del botó esborrar,neteja el canvas.



```
cuando BOTESBORRAR .Clic
  ejecutar
    llamar Lienzo1 .Limpiar
```

Prova l'aplicació

7.- Amb els botons + o - canviar la mida del pinzell. La mida serà entre 1 i 5.



```
cuando BOTMENYS .Clic
  ejecutar
    si
      tomar global MIDA_PINZELL > 1
      entonces
        poner global MIDA_PINZELL a tomar global MIDA_PINZELL - 1

cuando Bot_MES .Clic
  ejecutar
    si
      tomar global MIDA_PINZELL < 5
      entonces
        poner global MIDA_PINZELL a tomar global MIDA_PINZELL + 1
```

8.- Quan canviï la mida del pinzell que a l'etiqueta mida posi la mida.

```

cuando BOTMENYS .Clic
ejecutar
  si
    tomar global MIDA_PINZELL > 1
  entonces
    poner global MIDA_PINZELL a tomar global MIDA_PINZELL - 1
    poner ETIQ_MIDA . Texto como tomar global MIDA_PINZELL

cuando Bot_MES .Clic
ejecutar
  si
    tomar global MIDA_PINZELL < 5
  entonces
    poner global MIDA_PINZELL a tomar global MIDA_PINZELL + 1
    poner ETIQ_MIDA . Texto como tomar global MIDA_PINZELL
  
```

9.- En arrossegar el dit el canvas es dibuixen línees segon la mida del pinzell.

```

cuando Lienzo1 .Arrastrado
ejecutar
  llamar Lienzo1 .DibujarLinea
    x1 tomar XPrevio
    y1 tomar YPrevio
    x2 tomar XActual
    y2 tomar YActual
  poner Lienzo1 . AnchoDeLinea como tomar global MIDA_PINZELL
  
```

10.- En tocar damunt del botó foto,s'activa la càmera del mòbil.

```

cuando BOT_FOTO .Clic
ejecutar
  llamar Cámara1 .TomarFoto
  
```

11.- Una vegada feta la foto canviar el canvas

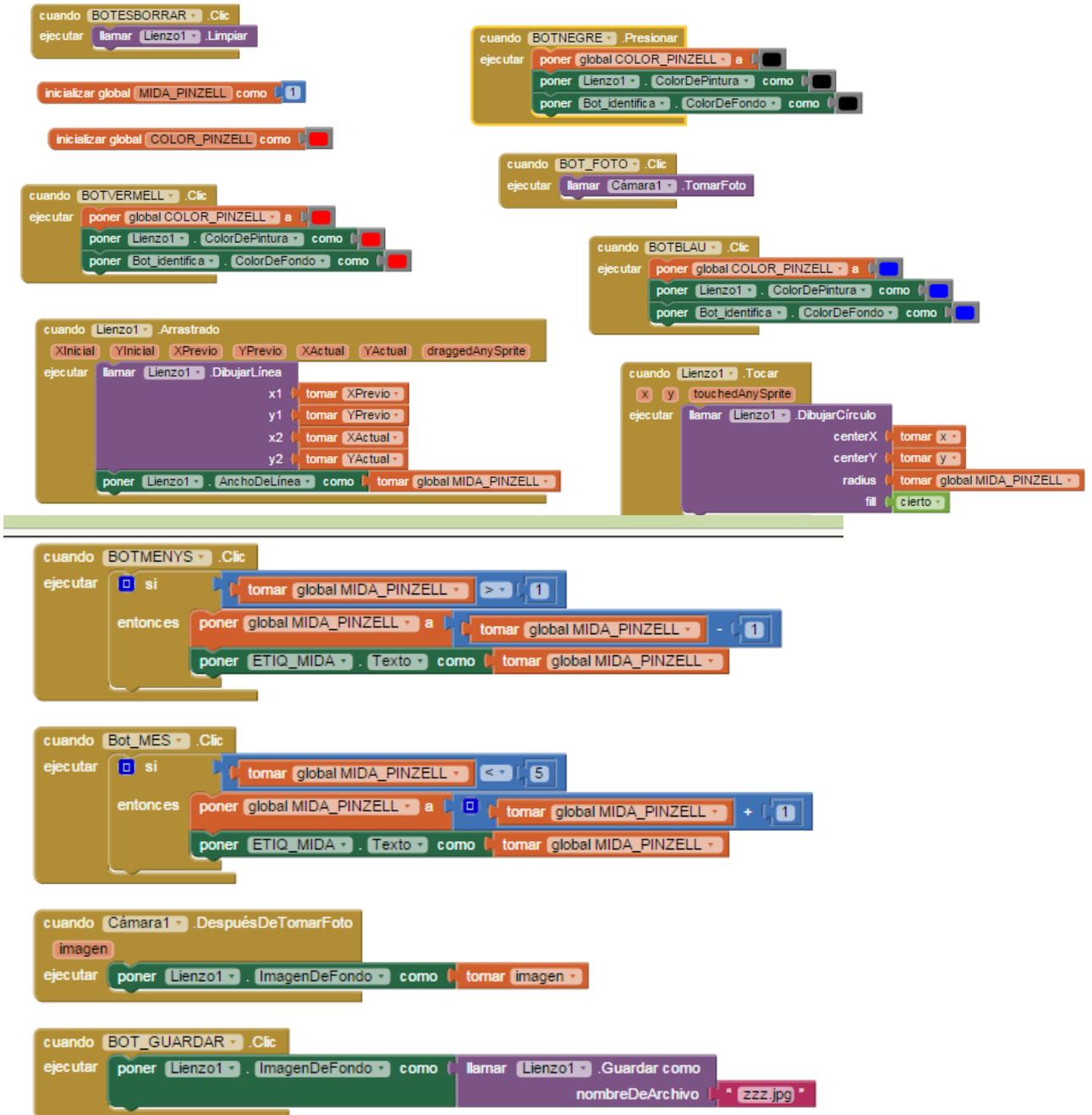
```

cuando Cámara1 .DespuésDeTomarFoto
  imagen
ejecutar
  poner Lienzo1 . ImagenDeFondo como tomar imagen
  
```

12.- En tocar damunt del botó Guardar, la imatge del canvas es guarda en el mòbil.



RESULTAT SENCER:



The image displays several blocks of code from the MIT App Inventor visual programming environment, organized into two sections separated by a horizontal line. The code is written in Spanish and controls a drawing application.

Top Section:

- cuando BOTESBORRAR . Clic:**
 - ejecutar: llamar Lienzo1 . Limpiar
- inicializar global (MIDA_PINZELL) como 1**
- inicializar global (COLOR_PINZELL) como [rojo]**
- cuando BOTVERMELL . Clic:**
 - ejecutar: poner global COLOR_PINZELL a [rojo]; poner Lienzo1 . ColorDePintura como [rojo]; poner Bot_identifica . ColorDeFondo como [rojo]
- cuando BOTNEGRE . Presionar:**
 - ejecutar: poner global COLOR_PINZELL a [negro]; poner Lienzo1 . ColorDePintura como [negro]; poner Bot_identifica . ColorDeFondo como [negro]
- cuando BOT_FOTO . Clic:**
 - ejecutar: llamar Cámara1 . TomarFoto
- cuando BOTBLAU . Clic:**
 - ejecutar: poner global COLOR_PINZELL a [blau]; poner Lienzo1 . ColorDePintura como [blau]; poner Bot_identifica . ColorDeFondo como [blau]
- cuando Lienzo1 . Arrastrado:**
 - ejecutar: llamar Lienzo1 . DibujarLinea (with parameters: x1: tomar XPrevio, y1: tomar YPrevio, x2: tomar XActual, y2: tomar YActual); poner Lienzo1 . AnchoDeLinea como tomar global MIDA_PINZELL
- cuando Lienzo1 . Tocar:**
 - ejecutar: llamar Lienzo1 . DibujarCirculo (with parameters: centerX: tomar x, centerY: tomar y, radius: tomar global MIDA_PINZELL, fill: cierto)

Bottom Section:

- cuando BOTMENYS . Clic:**
 - ejecutar: si (tomar global MIDA_PINZELL > 1) entonces poner global MIDA_PINZELL a (tomar global MIDA_PINZELL - 1); poner ETIQ_MIDA . Texto como tomar global MIDA_PINZELL
- cuando Bot_MES . Clic:**
 - ejecutar: si (tomar global MIDA_PINZELL < 5) entonces poner global MIDA_PINZELL a (tomar global MIDA_PINZELL + 1); poner ETIQ_MIDA . Texto como tomar global MIDA_PINZELL
- cuando Cámara1 . DespuésDeTomarFoto:**
 - imagen: poner Lienzo1 . ImagenDeFondo como tomar imagen
- cuando BOT_GUARDAR . Clic:**
 - ejecutar: poner Lienzo1 . ImagenDeFondo como llamar Lienzo1 . Guardar como nombreDeArchivo " zzz.jpg "

ACTIVITAT 9: JOC DE FUTBOL

COMPONENTS:

Canvas

1 imatge: porteria futbol.

so

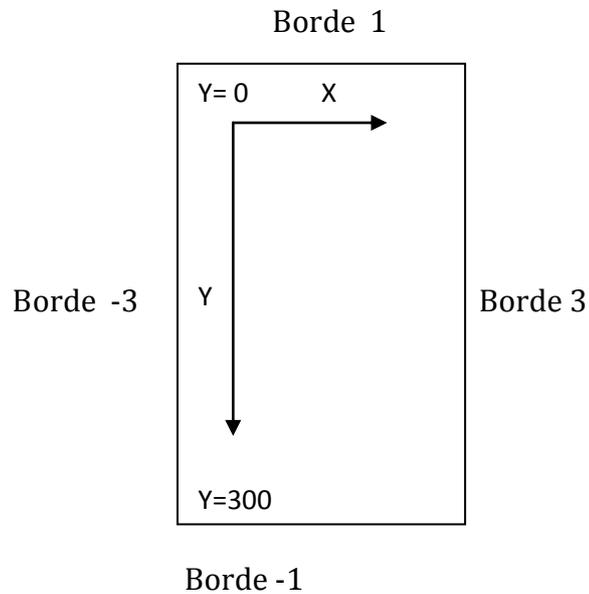
Botó reset

Botó pausa

sensor orientació

Temporitzador

4 Etiqueta



ESDEVENIMENTS:

1.- Moure porteria de manera aleatòria en l'eix X. Fer un "procedimiento".

2.- Moure la pilota amunt. "Lanzar pilota en direcció +90"

Lanzar(Inclinación): 0 grados cuando el dispositivo esta en horizontal, aumenta hasta 90 grados (amunt) cuando el dispositivo está apoyado en vertical sobre su lado izquierdo, y disminuye hasta -90 grados está apoyado en vertical sobre su lado derecho.

3.- Quan col·lisió porteria i pilota, pilota torna a baixar. La porteria es mou.

4. Quan pilota toca el borde 1, torna a baixar i la porteria es mou.

5.- Si pilota i porteria col·lisionen, Comptador de gols +1 i so de gol.

4.- Si la pilota col·lisió "borde 1". Comptador d'errors +1.

5.- Pilota es mou de forma aleatòria pel canvas.

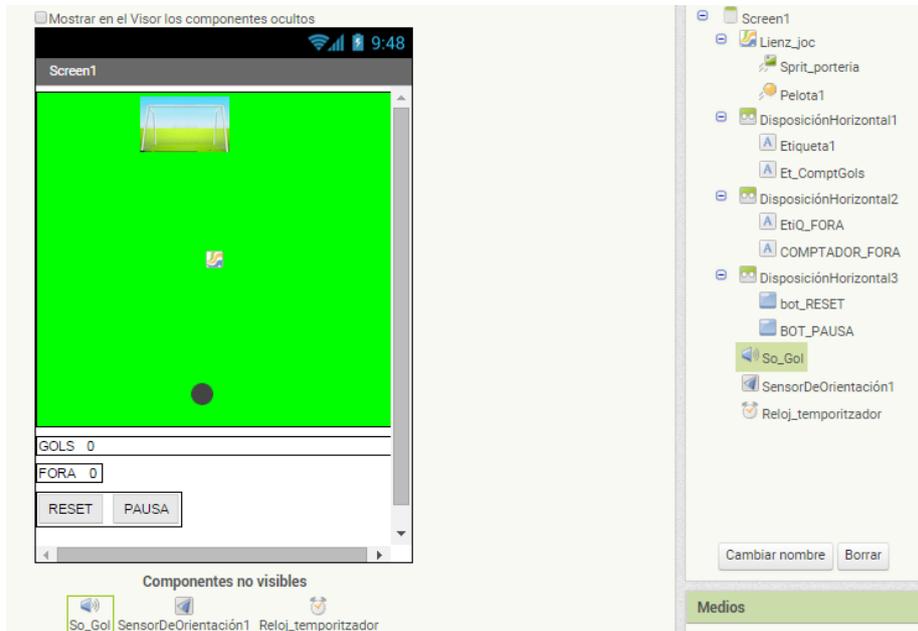
6.- Botó reset etiquetes comptadors a 0.

7.- Botó pausa aturar el joc i reanudar-lo.

PROPIETATS COMANDAMENTS

Pilota:velocitat 10,radi 10,dirección90 , intervalo100

DISSENY:



1. Programar un procediment moure la porteria horitzontalment :



ó



2.- Porteria col·lisió amb la pilota, la pilota torna a baixar i la porteria es desplaça horitzontalment.

```

cuando Sprit_porteria .EnColisióCon
    otro
    ejecutar
        llamar Pelota1 .MoverA
            x 150
            y 290
        llamar MOURE_PORTERIA
    
```

3.- Porteria col·lisió amb la pilota, Comptador de gols +1 i so de gol.

```

cuando Sprit_porteria .EnColisióCon
    otro
    ejecutar
        llamar Pelota1 .MoverA
            x 150
            y 290
        llamar MOURE_PORTERIA
        poner Et_ComptGols .Texto como Et_ComptGols .Texto + 1
        llamar So_Gol .Reproducir
    
```

4.- La pilota, quan el mòbil està horitzontal va cap amunt. Si s'inclina a dreta o esquerra dirigint la pilota.

```

cuando SensorDeOrientaci3n1 .CambioEnOrientaci3n
    acimut tono lanzar
    ejecutar
        poner Pelota1 .Direcci3n como tomar lanzar + 90
    
```

5.- Si la pilota toca el "borde 1", es mou cap abaix , la porteria es mou i el comptador de fores +1.

```
cuando Pelota1 .TocarBorde
borde
ejecutar
  si
  entonces
    llamar Pelota1 .MoverA
      x 150
      y 290
    llamar MOURE_PORTERIA
    poner COMPTADOR_FORA . Texto como COMPTADOR_FORA . Texto + 1
```

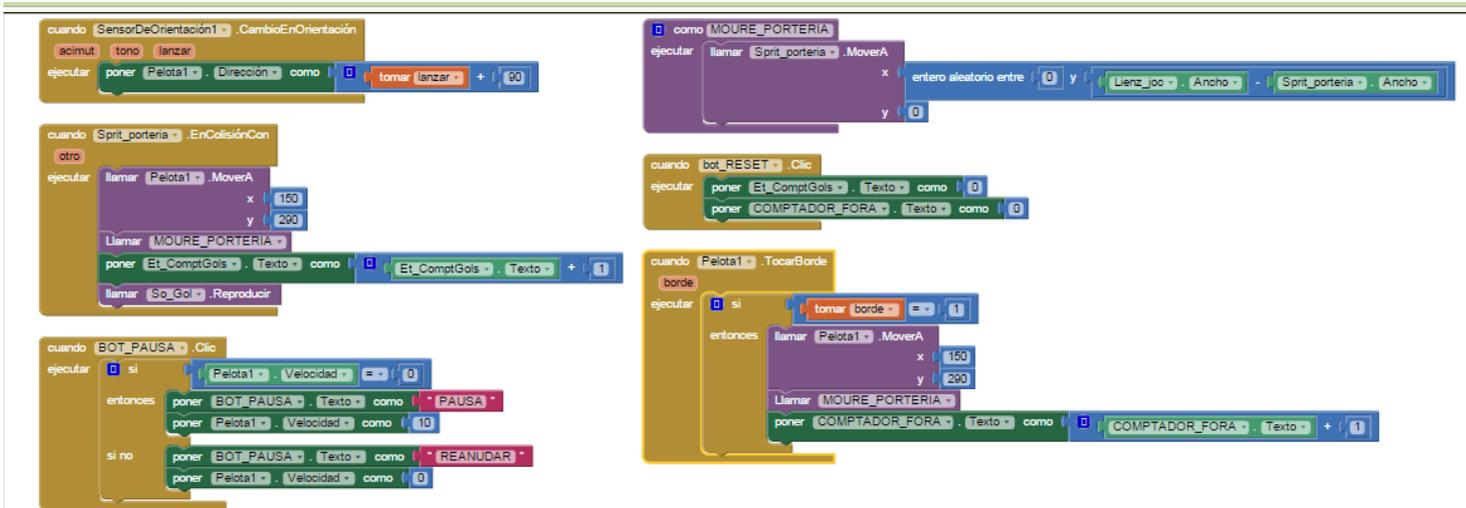
6.- En tocar damunt del botó reset els comptadors a 0.

```
cuando bot_RESET .Clic
ejecutar
  poner Et_ComptGols . Texto como 0
  poner COMPTADOR_FORA . Texto como 0
```

7.- Si la velocitat de la pilota és 10, en tocar damunt del botó pausa s'atura el joc i al botó pausa surt el text REANUDAR.

```
cuando BOT_PAUSA .Clic
ejecutar
  si
  entonces
    poner BOT_PAUSA . Texto como " PAUSA "
    poner Pelota1 . Velocidad como 10
  si no
    poner BOT_PAUSA . Texto como " REANUDAR "
    poner Pelota1 . Velocidad como 0
```

PROGRAMA SENCER:



The image displays a collection of Scratch code blocks for a game, organized into several event-driven blocks:

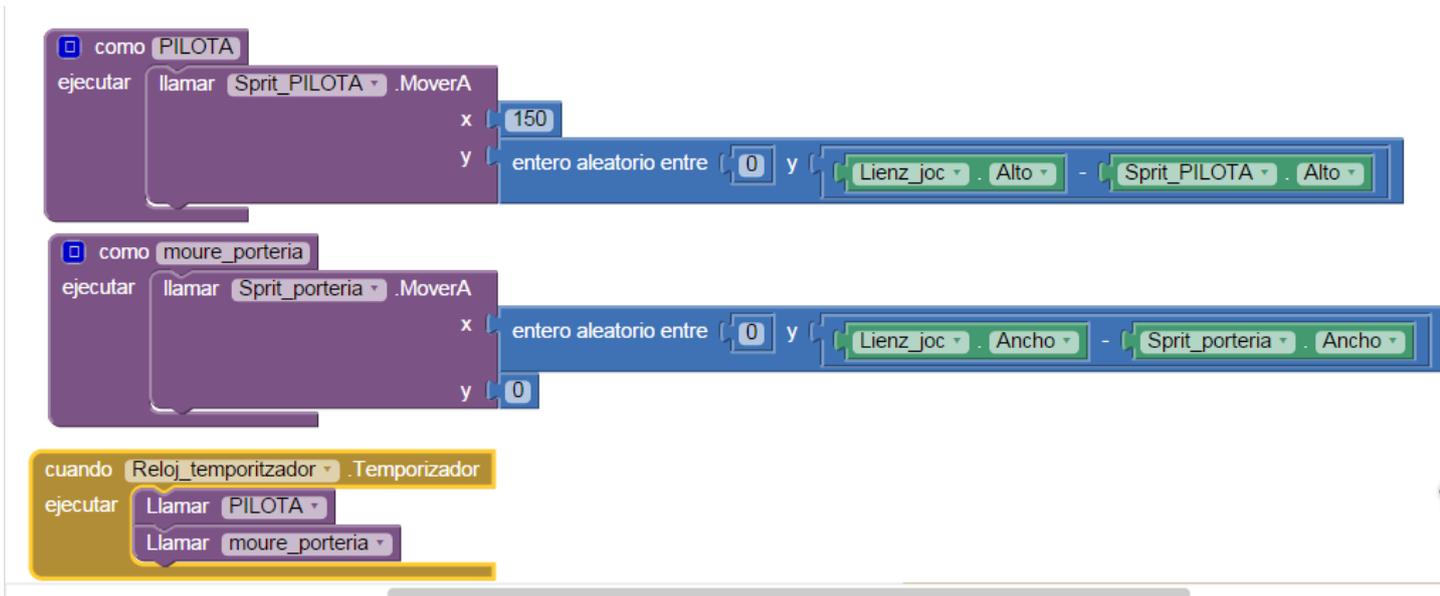
- when orientation changes:** A block for "SensorDeOrientación1" with the event "CambioEnOrientación". The action is to set "Pelota1" direction to "tomar lanzar" + 90.
- when collision:** A block for "Sprite_porteria" with the event "EnColisiónCon". The actions are: call "Pelota1" "MoverA" with x=150 and y=200; call "MOURE_PORTERIA"; set "Et_ComptGols" text to "Et_ComptGols" + 1; and call "So_Gol" "Reproducir".
- when button clicked:** A block for "BOT_PAUSA" with the event ".Clic". It contains an "if" block: if true, set "BOT_PAUSA" text to "PAUSA", set "Pelota1" velocity to 10; if false, set "BOT_PAUSA" text to "REANUDAR" and set "Pelota1" velocity to 0.
- when ball touches border:** A block for "Pelota1" with the event ".TocarBorde". It contains an "if" block: if true, set "tomar borde" to 1, call "Pelota1" "MoverA" with x=150 and y=200, call "MOURE_PORTERIA", and set "COMPTADOR_FORA" text to "COMPTADOR_FORA" + 1.
- when sprite clicked:** A block for "MOURE_PORTERIA" with the event ".Clic". The action is to call "Sprite_porteria" ".MoverA" with x="entero aleatorio entre 0 y Lienz_joc - Ancho" and y=0.
- when button clicked:** A block for "bot_RESET" with the event ".Clic". The actions are to set "Et_ComptGols" text to 0 and "COMPTADOR_FORA" text to 0.

ACTIVITAT 10: JOC DE FUTBOL AMB IMATGE PILOTA.

Ara farem servir dues imatges en el canvas, una porteria i una pilota de futbol.

Amb el mateix disseny que l'exercici anterior, programa que la porteria es desplaci aleatòriament de dreta a esquerra. I la pilota amunt i a baix.

SOLUCIÓ:



The image shows three Scratch code blocks for a soccer game:

- Block 1:** A "when green flag clicked" block (yellow) with a "when green flag clicked" trigger. It contains an "execute" block (purple) with a "call sprite PILOTA" block. The "call sprite PILOTA" block has an "execute" block (purple) with a "call sprite Sprit_PILOTA" block. The "call sprite Sprit_PILOTA" block has a "move" block (blue) with "x" set to 150 and "y" set to "random number between 0 and (Lienz_joc . Alto) - (Sprit_PILOTA . Alto)".
- Block 2:** A "when green flag clicked" block (yellow) with a "when green flag clicked" trigger. It contains an "execute" block (purple) with a "call sprite moure_porteria" block. The "call sprite moure_porteria" block has an "execute" block (purple) with a "call sprite Sprit_porteria" block. The "call sprite Sprit_porteria" block has a "move" block (blue) with "x" set to "random number between 0 and (Lienz_joc . Ancho) - (Sprit_porteria . Ancho)" and "y" set to 0.
- Block 3:** A "when green flag clicked" block (yellow) with a "when green flag clicked" trigger. It contains an "execute" block (purple) with a "call Relej_temporitzador" block. The "call Relej_temporitzador" block has a "call Relej_temporitzador" block with a "call Relej_temporitzador" block. The "call Relej_temporitzador" block has a "call Relej_temporitzador" block.

ACTIVITAT 11: SIMULACIÓ SMS. LECTOR I RECONEIXEMENT DE VEU.

Utilització de les TinyDB

Les TinyDB serveixen per anar a una base de dades. Va guardant parelles de valors.

COMPONENTS:

1 Campo de texto

2 botones

3 textoAvoz

2 reconocimiento de voz

1 reloj

1 enviartexto

1tinyBD

1 Notificador

Campo de texto: 

Es un campo que permite al usuario introducir texto. En la propiedad Texto se almacena el texto que ha introducido el usuario. Si el campo está vacío, la propiedad Pista, que aparece como un texto atenuado dentro del campo de texto, puede proporcionar alguna pista al usuario sobre lo que tiene que escribir.

La propiedad Multilínea determina si el texto debe estar compuesto por más de una línea. En el caso de un campo de texto de una sola línea, el teclado se ocultará automáticamente cuando el usuario pulse la tecla Hecho. Cuando se trate de un campo Multilínea, la aplicación tendrá que utilizar el método OcultarTeclado, o confiar en que el usuario pulse la tecla Atrás para ocultarlo.

La propiedad SóloNúmeros obliga a que únicamente puedan introducirse números.

Otras propiedades afectan a la apariencia de un campo de texto (PosiciónDelTexto, ColorDeFondo, etc.), y determinan si el campo es editable o no (Habilitado).

Es habitual utilizar los campos de texto junto a componentes Botón, de tal manera que el usuario pulsa el botón cuando ha terminado de introducir el texto en el campo.

Cuando no se quiere que sea visible el texto que está escribiendo el usuario, es más apropiado utilizar el componente CampoDeContraseña.

El componente Notificador: 

Muestra cuadros con alertas, mensajes y alertas temporales, y hace anotaciones en el registro de Android utilizando los siguientes métodos:

- MostrarDiálogoMensaje: presenta un mensaje que el usuario puede descartar pulsando un botón.
- MostrarDiálogoElección: muestra un mensaje y dos botones, para que el usuario pueda elegir entre dos respuestas, por ejemplo si o no, después de lo cual si activa el evento DespuésDeSelección.
- MostrarDiálogoTexto: permite al usuario escribir una respuesta al mensaje, después de lo cual se activa el evento DespuésDeEntradaDeTexto.
- MostrarAlerta: presenta una alerta que desaparece después de un breve periodo de tiempo.
- RegistrarError: anota un mensaje de error en el registro de Android.
- RegistrarInfo: escribe un mensaje de información en el registro de Android.
- RegistrarAviso: anota un mensaje de aviso o advertencia en el registro de Android.
- Se puede poner formato a los mensajes que aparecen en los cuadros de diálogo (pero no a las alertas) usando las siguientes etiquetas HTML: , <big>, <blockquote>,
, <cite>, <dfn>, <div>, , <small>, , <sub>, <sup>, <tt>. <u>
- También se puede utilizar la etiqueta font para definir, por ejemplo, el color con . Entre los nombres de colores disponibles están aqua, black, blue, fuchsia, green, grey, lime, maroon, navy, olive, purple, red, silver, teal, white, y yellow

Texto de voz: 

Componente para utilizar la función TextoAVoz para reproducir un mensaje.

Reconocimiento de voz: 

Es un componente que permite utilizar el reconocimiento de voz para convertir de voz a texto.

Reloj: 

Es un componente no visible, que permite utilizar un reloj, o un temporizador, y hacer cálculos de tiempos.

Enviar texto: 

Un componente que, cuando se hace una llamada al método EnviarMensaje, envía el mensaje de texto indicado en la propiedadMensaje al número de teléfono especificado en la propiedad NúmeroDeTeléfono.

Si la propiedad RecepciónHabilitada tiene el valor 1 **no** se recibirán mensajes. Si la propiedad RecepciónHabilitada tiene el valor 2 sólo se recibirán mensajes cuando la aplicación se esté ejecutando. Por último, si la propiedad RecepciónHabilitada tiene el valor 3, se recibirán los mensajes cuando la aplicación se esté ejecutando, **y cuando no lo esté** se pondrán los mensajes en cola y se mostrará al usuario una notificación.

Cuando llega un mensaje se activa el eventoMensajeRecibido, que indica el número que lo envía y el mensaje.

Una aplicación que incluya este componente recibirá los mensajes incluso cuando la aplicación no esté en primer plano (por ejemplo, cuando no es visible en la pantalla, e incluso cuando la aplicación no se está ejecutando pero sí está instalada en el teléfono. Si el teléfono recibe un mensaje de texto cuando la aplicación no está en la pantalla, el teléfono mostrará una notificación en la barra de notificaciones. Al hacer clic sobre la notificación se mostrará la aplicación. Como programador, seguramente desees ofrecer a tus usuarios la capacidad de controlar el estado de RecepciónHabilitada, de modo que puedan ignorar los mensajes de texto.

Si la propiedad VozDeGoogleHabilitada tiene el valor Verdadero, se podrán enviar los mensajes a través de una Wifi utilizando Google Voice, Para ello es necesario que el usuario disponga de una cuenta de Google Voice, y que dicha aplicación esté instalada en el teléfono. Esta funcionalidad solamente es posible en dispositivos que tengan instalada la versión de Android 2.0 (Eclair) o superior.

Para especificar el número de teléfono (por ejemplo, 650-555-1212) hay que poner en la propiedad NúmeroDeTeléfono una cadena de texto con el número (650-555-1212). Pueden incluirse guiones, puntos o paréntesis ((650)-555-1212)), pero el sistema los ignorará. No se permiten espacios.

Otro método para indicar un número de teléfono en una aplicación consiste en utilizar un componente SelectorNúmeroTfno, que permite al usuario seleccionar

un número de teléfono de entre todos los que tiene almacenados en la lista de contactos del teléfono.



es un componente no visible que puede almacenar datos.

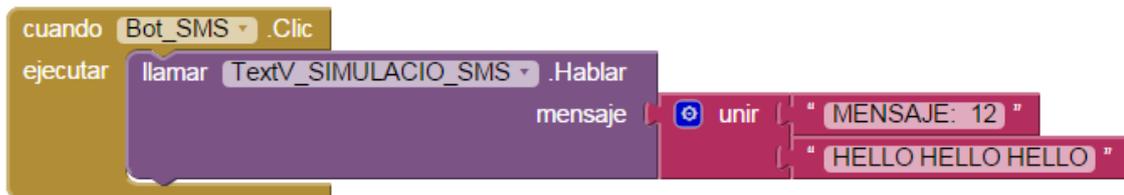
Las aplicaciones creadas con App Inventor se inicializan cada vez que se ejecutan: si una aplicación inicializa el valor de una variable y el usuario termina la ejecución de la aplicación, el valor de esa variable se perderá, y no estará disponible la próxima vez que se ejecute la aplicación. TinyDB es un almacén de datos *persistente* para la aplicación, lo que significa que los datos estarán disponibles cada vez que se abra la aplicación. Un ejemplo de uso puede ser un juego que guarda las puntuaciones más altas cada vez que se juega.

Los datos son cadenas de texto que se almacenan en *etiquetas*. Para almacenar datos, se debe de especificar la etiqueta que corresponde a ese dato. Posteriormente, los datos se pueden recuperar con esa misma etiqueta.

Solamente existe una base de datos para cada aplicación. Aunque se utilicen distintas TinyDBs en una aplicación, todas ellas apuntan al mismo almacén. Se deben utilizar distintas etiquetas para guardar diferentes datos, y no usar diferentes almacenes. TinyDB es única para cada aplicación, y no se puede utilizar para pasar datos a otras aplicaciones instaladas en el dispositivo, pero se sí puede utilizar para pasar datos entre pantallas en la misma aplicación.

Cuando TinyDB se utiliza con la aplicación Companion, todas las aplicaciones comparten la misma DB. Esto no ocurre cuando la aplicación se empaqueta como un archivo apk. Pero mientras se desarrolla, se debe ser cuidadoso en situaciones en las que las mismas etiquetas se usan para distintas aplicaciones.

1.-Quan clico botó sms entrant li dic a un texto de voz que digui el que jo introduirè.



2.- Después de hablar activar el temporizador.

```
cuando TextV_SIMULACIO_SMS .DespuésDeHablar
Resultado
ejecutar poner Reloj1 . TemporizadorHabilitado como cierto
```

3.- Activeu missatge “¿Quieres responder?”

```
cuando Reloj1 .Temporizador
ejecutar poner Reloj1 . TemporizadorHabilitado como falso
llamar TEXV_RESPOSTASMs_SIMULACIO .Hablar
mensaje unir "¿QUIERES RESPONDER?"
```

4.- Activa el micro per respondre.

```
cuando TEXV_RESPOSTASMs_SIMULACIO .DespuésDeHablar
Resultado
ejecutar llamar MICRO_SMS .ObtenerTexto
```

5.-Si el Micro rep SI, sortirà una notificació que digui RESPONDEMOS. En cas contrari la notificació serà NO RESPONDEMOS

```
cuando MICRO_SMS .DespuésDeObtenerTexto
Resultado
ejecutar si
mayúscula tomar Resultado = "SI"
entonces llamar Notificador1 .MostrarAlerta
aviso "Respondemos"
si no
llamar Notificador1 .MostrarAlerta
aviso "No respondemos"
```

6.- En tocar damunt del botó resposta s'obre un altre micròfon.

```
cuando Bot_RESPOSTA .Clic
ejecutar
  llamar MICRO_RESPOSTA .ObtenerTexto
  llamar TEXV_LEERESPUESTA .Hablar
  mensaje unir " DI TU NUEVO MENSAJE "
```

7.-La respuesta s'ha de convertir en text en el campo de texto. El missatge s'ha de guardar en una tinyBD.

```
cuando MICRO_RESPUESTA .DespuésDeObtenerTexto
  Resultado
ejecutar
  poner CampoDeTexto1 . Texto como tomar Resultado
  llamar TinyBD1 .GuardarValor
  etiqueta " MISSATGE "
  valorAGuardar tomar Resultado
```

PROGRAMA SENCER.

visor

```

cuando Bot_SMS .Clic
ejecutar
  llamar TextV_SIMULACIO_SMS .Hablar
  mensaje unir MENSAJE: 112 , DICE:HOLA HOLA HOLA

cuando TextV_SIMULACIO_SMS .DespuésDeHablar
  Resultado
ejecutar
  poner Reloj1 .TemporizadorHabilitado como cierto

cuando Reloj1 .Temporizador
ejecutar
  poner Reloj1 .TemporizadorHabilitado como falso
  llamar TEXV_RESPOSTASms_SIMULACIO .Hablar
  mensaje unir ¿QUIÉRES RESPONDER?

cuando TEXV_RESPOSTASms_SIMULACIO .DespuésDeHablar
  Resultado
ejecutar
  llamar MICRO_SMS .ObtenerTexto

cuando MICRO_SMS .DespuésDeObtenerTexto
  Resultado
ejecutar
  si mayúscula tomar Resultado = SI
  entonces
    llamar Notificador1 .MostrarAlerta
    aviso Responderemos
  si no
    llamar Notificador1 .MostrarAlerta
    aviso No respondemos
  
```

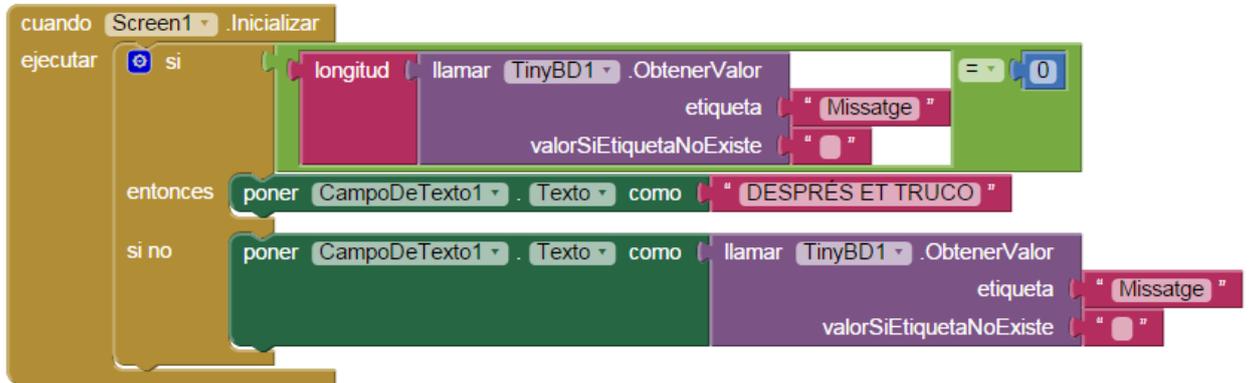
Mostrar Aviso

```

cuando Bot_RESPUESTA .Clic
ejecutar
  llamar MICRO_RESPUESTA .ObtenerTexto
  llamar TEXV_LEERESPUESTA .Hablar
  mensaje unir DI TU NUEVO MENSAJE

cuando MICRO_RESPUESTA .DespuésDeObtenerTexto
  Resultado
ejecutar
  poner CampoDeTexto1 .Texto como tomar Resultado
  llamar TinyBD1 .GuardarValor
  etiqueta Missatge
  valorAGuardar tomar Resultado
  
```

ACTIVITAT 12. AL OBRIR L'APLICACIÓ ANTERIOR RECUPEREM EL MISSATGE GUARDADO AMB L'ETIQUETA MISSATGE DE LA TINYBD.



Canvia el nom de la primera etiqueta " mensaje " i observa què passa: