

# Desenvolvimento de jogos com o Game Maker Studio 1.4

Prof. Me. Sérgio Carlos Portari Júnior

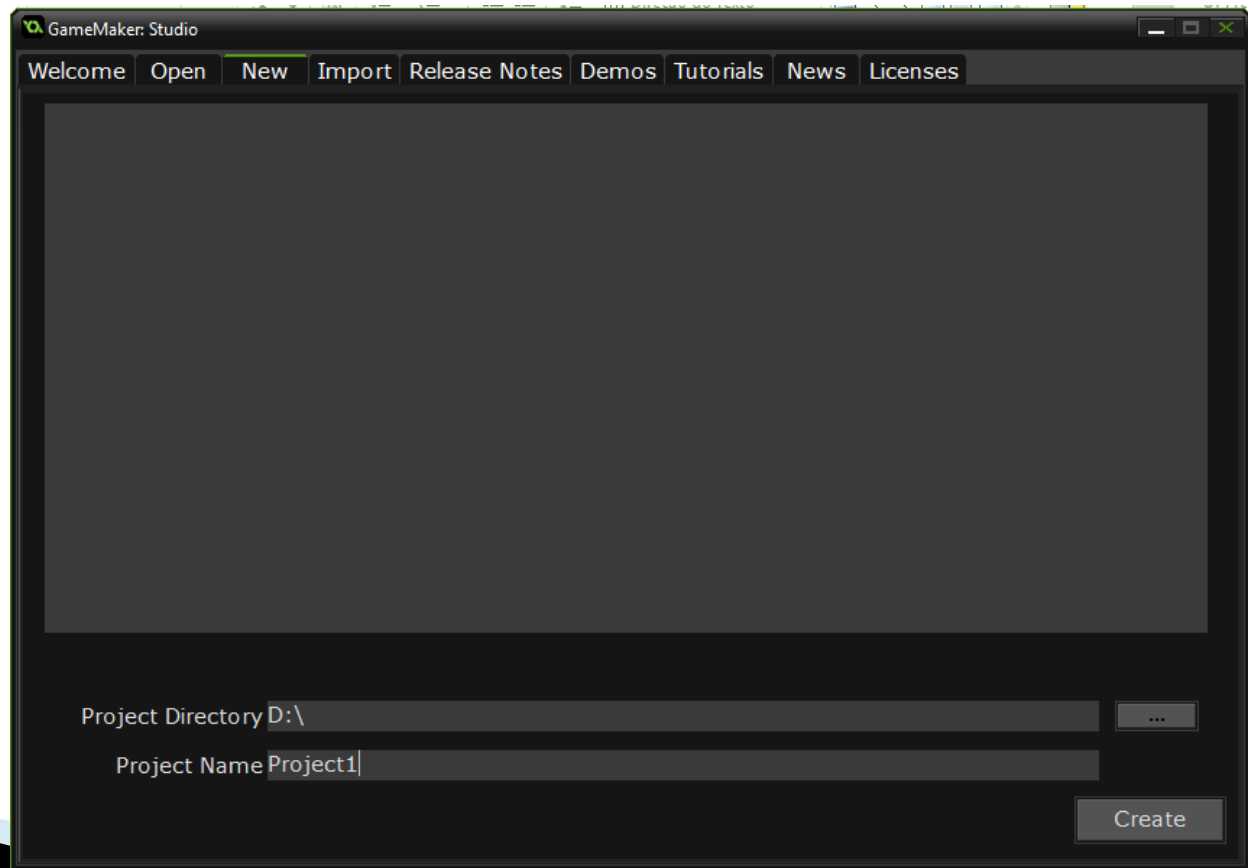
Maio / 2016

# Objetivo do minicurso:

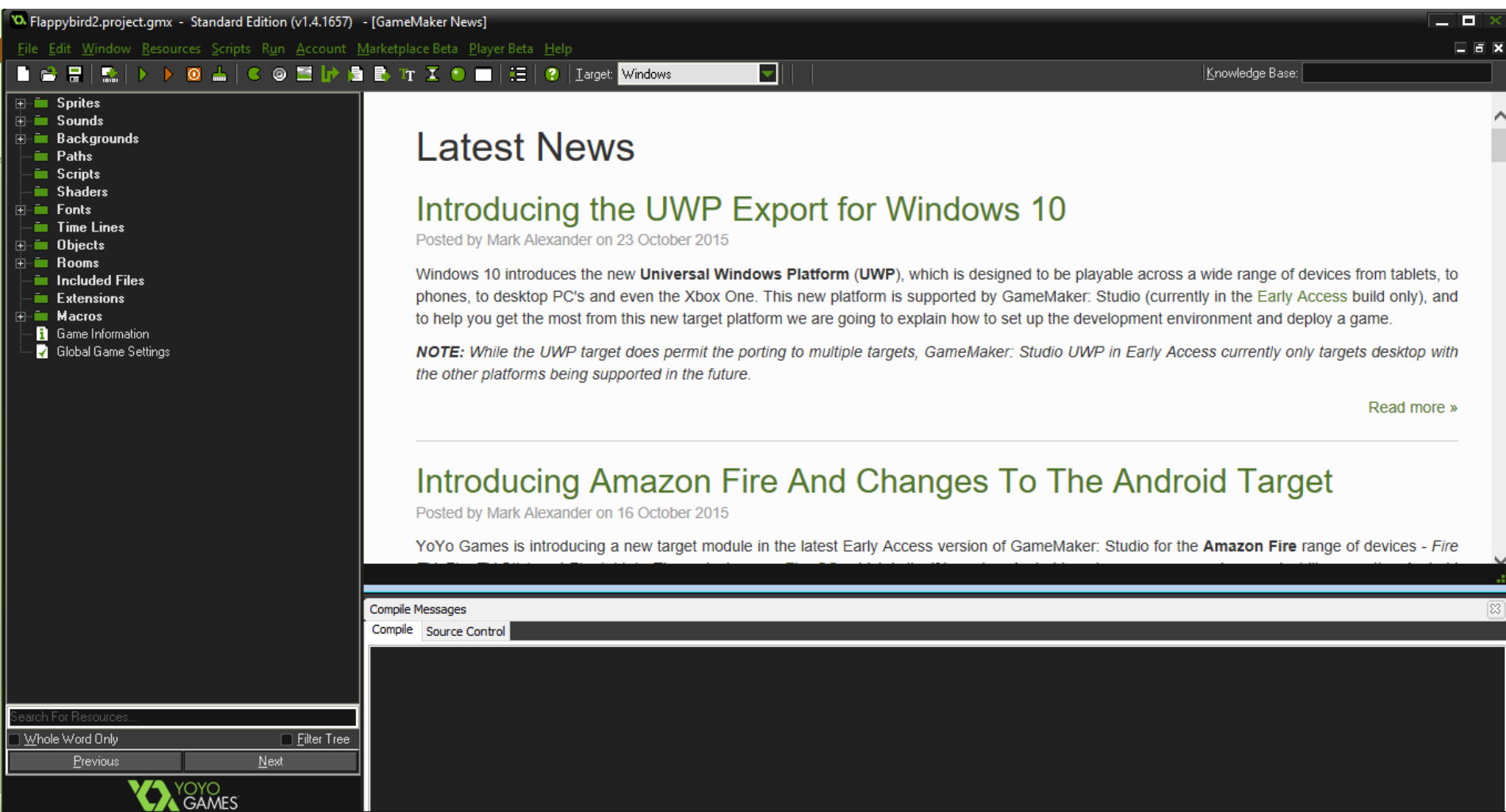
- ▶ Neste minicurso será apresentado o processo de planejamento de jogos 2D para serem desenvolvidos no GameMaker Studio, da Yoyo Games, na versão 1.4. Esta ferramenta permite a criação de jogos para diversos sistemas operacionais, incluindo para dispositivos móveis Android ou iOS, além de jogos padronizados para distribuição em lojas virtuais, como Steam. No minicurso, o aluno terá uma visão geral da ferramenta e grau de esforço para criação de jogos, explorando as funcionalidades básicas do GameMaker Studio, realizando um aprendizado na prática, onde completará o desenvolvimento de um jogo básico e terá, ao final, conhecimento para que possa evoluir, criando jogos mais sofisticados sozinho.

# Para começar...

- ▶ Clique em NEW e escolha pasta e nome do projeto



# Apresentação da interface do GameMaker Studio 1.4

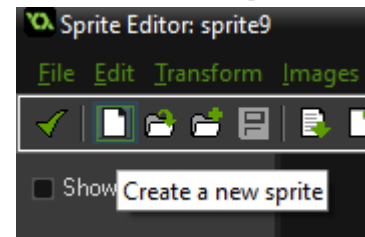
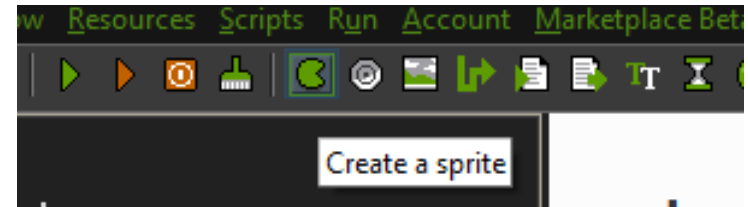


# Sprites

- ▶ Definição de sprite?
- ▶ Sprites – Não existe um termo em português para traduzir Sprite, mas seu significado em inglês é
- ▶ *“um gráfico de computador que pode ser movido na tela, manipulado como uma entidade única”*

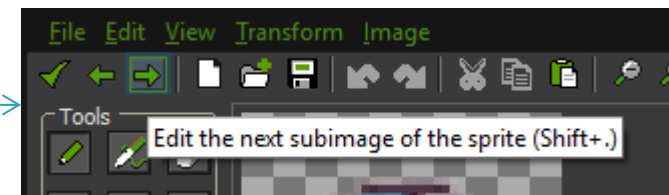
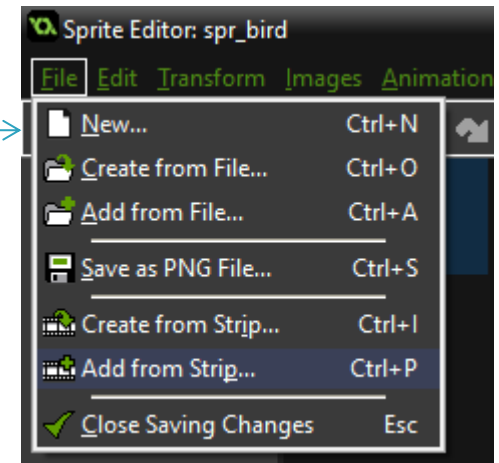
# Sprites

- ▶ Vamos criar um Sprite sendo um quadrado de cor qualquer para visualizarmos objetos que não terão sprites associados (será explicado um pouco adiante).
- ▶ O nome dele é SPR\_visualiza.
- ▶ Clique em Edit Sprite,
- ▶ Clique em Create New
- ▶ Coloque 32x32 nas dimensões . Dois cliques na image 0. Preencha o quadrado com uma cor (Fill, ou Atalho F). Selecione a cor, clique na imagem. Agora aperte V (verde em cima) para salvar, e novamente V (verde) para criar a Sprite.
- ▶ Clique em Centralizar a origem (Center abaixo, que deve ficar 16x16). Clique no Ok.



# Sprite – Criar animação (bird)

- ▶ Criar uma nova Sprite; Tamanho (40x40).
- ▶ Clique em Edit Sprite,
- ▶ Menu File – Add from Strip.
- ▶ Usar as imagens 1, 2, 3, 2.
- ▶ Tamanho 40x40, centro 20x20.
- ▶ Dê dois cliques em uma imagem
- ▶ Aperte o Zoom + (Lupa)
- ▶ Acerte a posição das imagens, todas ao centro, corte as arestas de outras imagens, se necessário. Aperte as setas para trocar de imagens e visualizar os centros.



# Sprites (pipe down)

- ▶ Criar Sprite vazia para o cano debaixo (64 x 600) (fica embaixo com a boca para cima)
- ▶ Clique em add from strip,
- ▶ Selecione o cano que aponta para baixo com 64,340.
- ▶ Ele vai perguntar o que fazer, utilize Streech.
- ▶ Depois **apague** a primeira criada **em branco** para ficar só com esta do cano.
- ▶ Utilize a origem na parte de cima do cano (aprox. 32 x 24)



# Sprites (pipe up)

- ▶ Criar Sprite vazia para o cano de cima (64 x 600) (fica em cima, com a boca para baixo)
- ▶ Clique em add from strip,
- ▶ Selecione o cano que aponta para baixo com 64,340.
- ▶ Ele vai perguntar o que fazer, utilize Streech.
- ▶ Depois **apague** a primeira criada **em branco** para ficar só com esta do cano.
- ▶ Utilize a origem na parte debaixo do cano (aprox. 32 x 580)

# Sprites – (floor)

- ▶ Criar o Sprite para o piso
- ▶ Create from Sprite,
- ▶ Tamanho 300x32 (deixar sobrar um pouco acima em branco).
- ▶ Origem no centro.

# Sprites – (signs)

- ▶ Vamos criar 3 sprites informativos
  - Get Ready! Cria um Sprite, tamanho 200 x 60. Centro no Centro
  - TAP – Cria um Sprite com tamanho 130x120. Centro no centro
  - FlapyStart – criar um Sprite para o início, 220x60, centro no centro

# BackGround (fundo de tela)

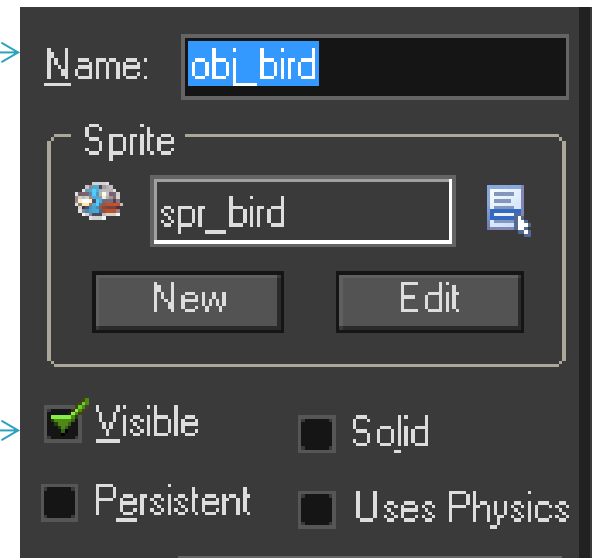
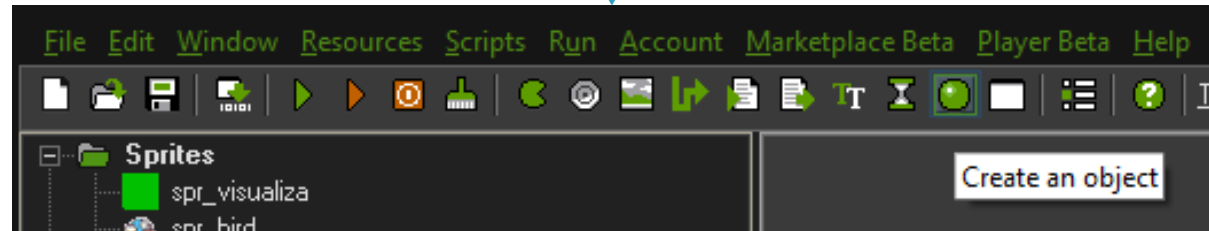


- ▶ Inserir o Background, 288 x 512

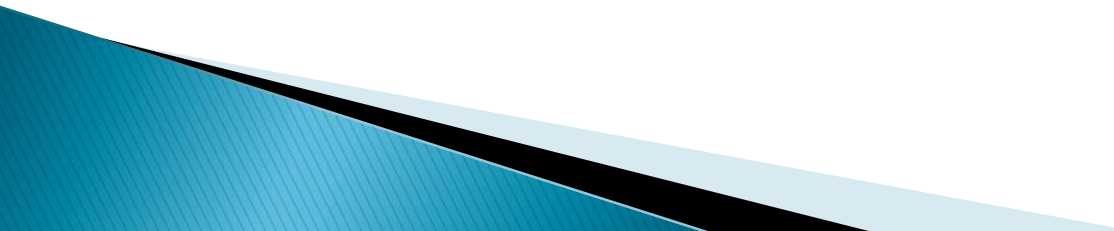


# Objetos (bird)

- ▶ Novo Objeto.
- ▶ Nome obj\_flappybird.
- ▶ Adicione a Sprite Bird -> spr\_flappybird.
- ▶ Deixe apenas a propriedade Visible
- ▶ ativada
- ▶ Aperte OK
- ▶ embaixo

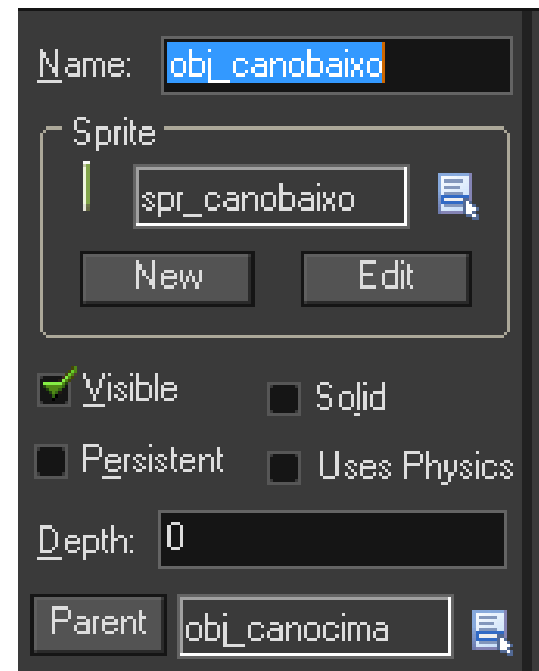


# Objetos (pipe up)

- ▶ Crie um novo objeto, chamado `obj_cano_cima`
  - ▶ Coloque a sprite do cano com a boca do tubo para cima (`spr_canocima`).
  - ▶ Deixe o Visible marcado
  - ▶ Aperte OK embaixo
- 

# Objetos (pipe down)

- ▶ Crie um novo objeto, chamado obj\_canobaixo
- ▶ Coloque a sprite do cano com a boca do tubo para baixo.
- ▶ No parent deste objeto, selecione o obj\_canocima.
- ▶ (Assim podemos programar tudo
- ▶ no objeto cano\_cima e
- ▶ tudo que fizer para ele
- ▶ funcionará para
- ▶ o cano\_baixo.)



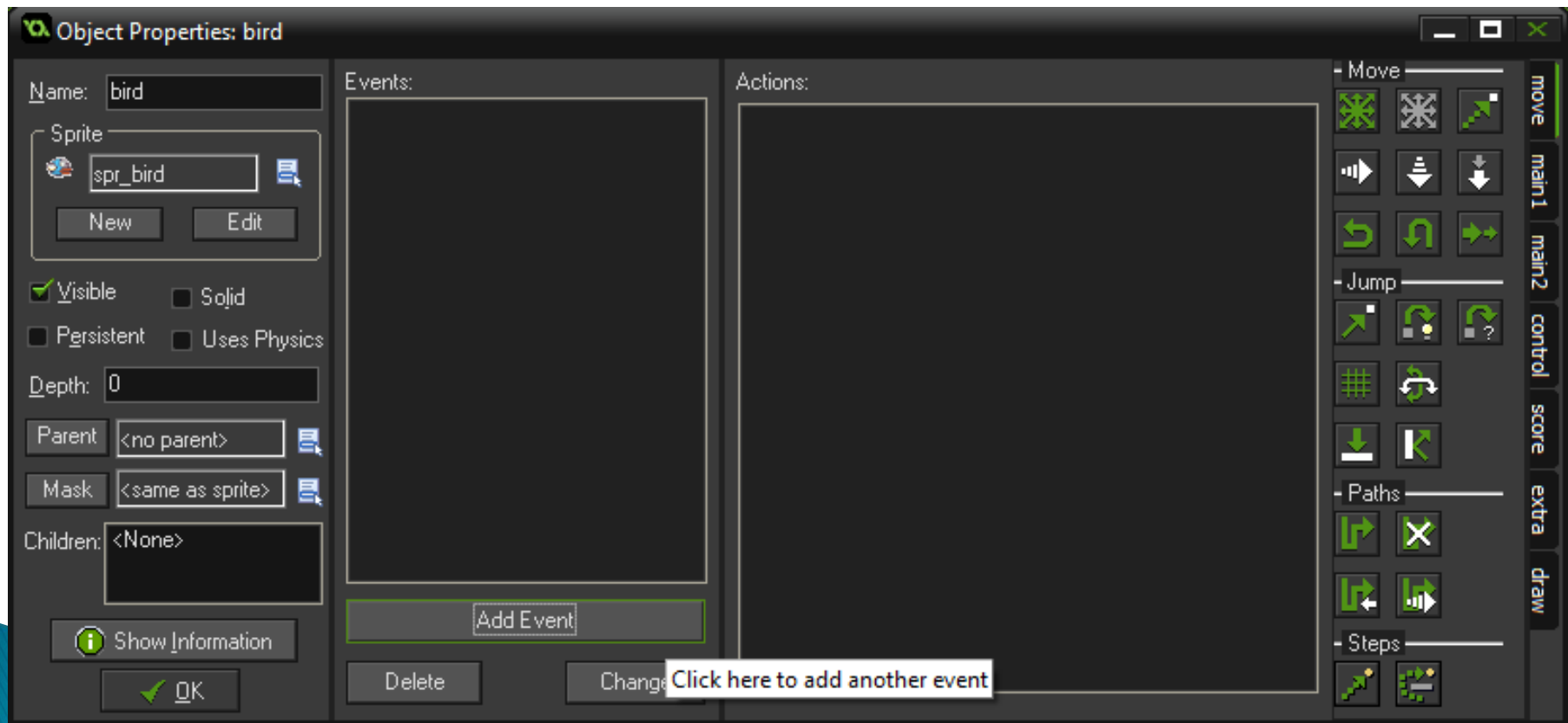
# Objetos (floor)

- ▶ Crie o obj\_piso,
- ▶ Coloque a Sprite spr\_piso
- ▶ Deixe a propriedade Visible marcada



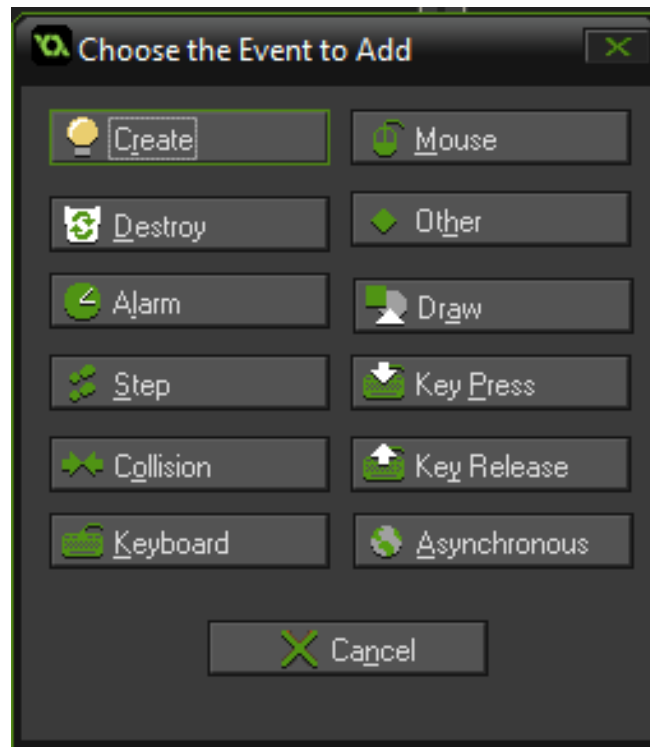
# Programando Eventos (bird)

- ▶ Dois cliques no objeto para abrí-lo.
- ▶ Clique em ADD EVENT



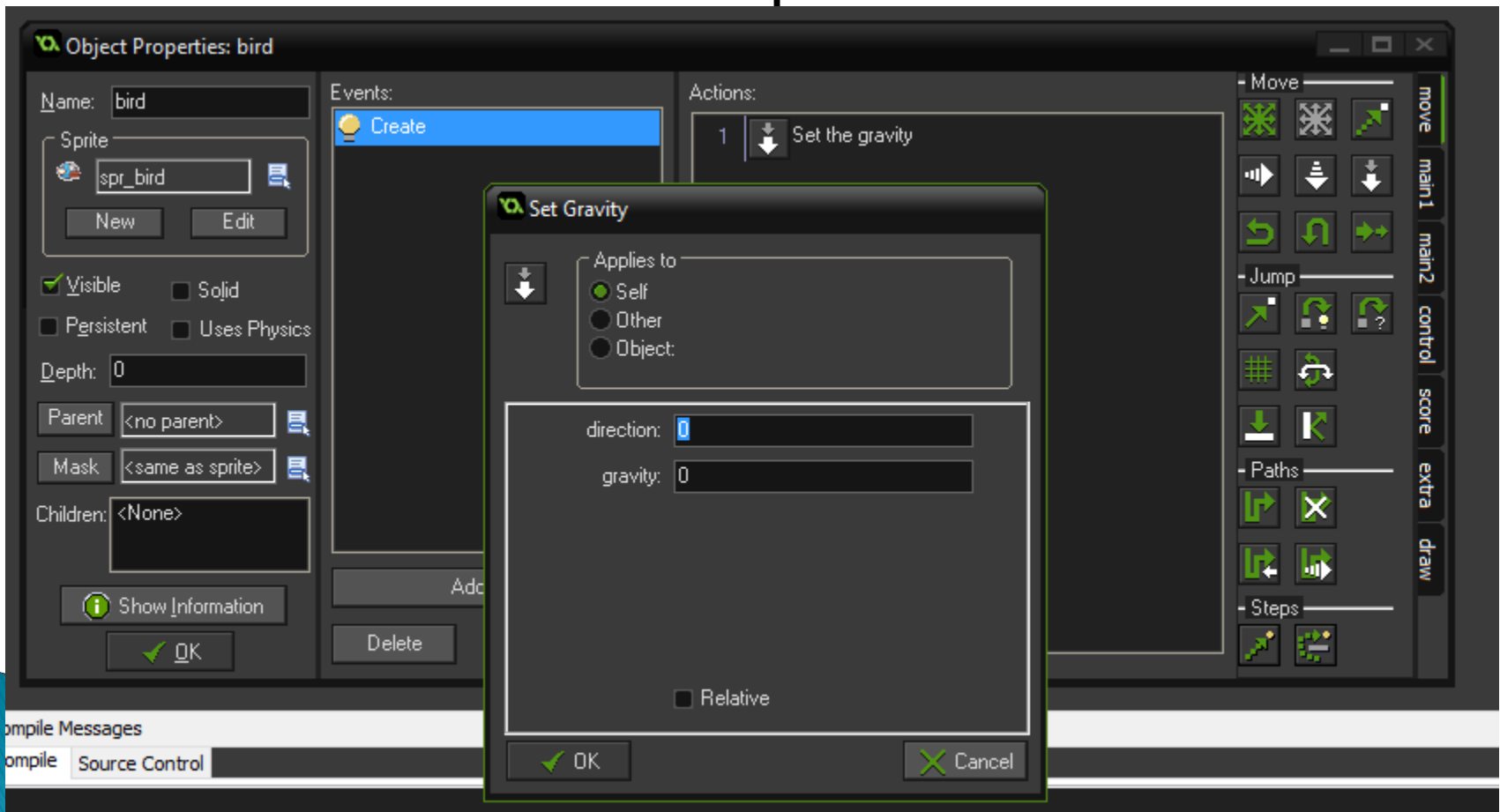
# Programando Eventos (bird)

- ▶ Selezione Create

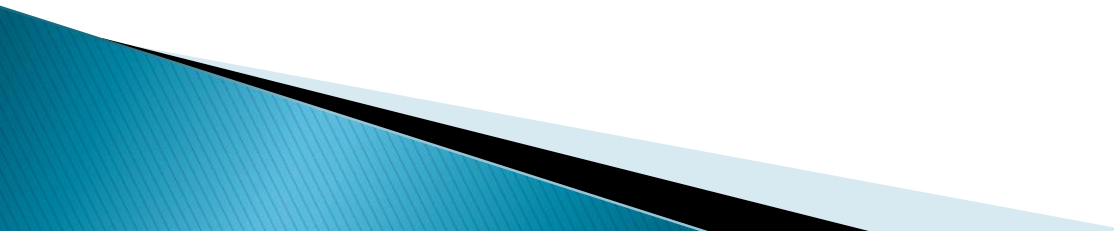


# Programando Eventos (bird)

- ▶ No create, adicione Set Gravity (menu Move),
- ▶ Clicando e arrastando para Actions.

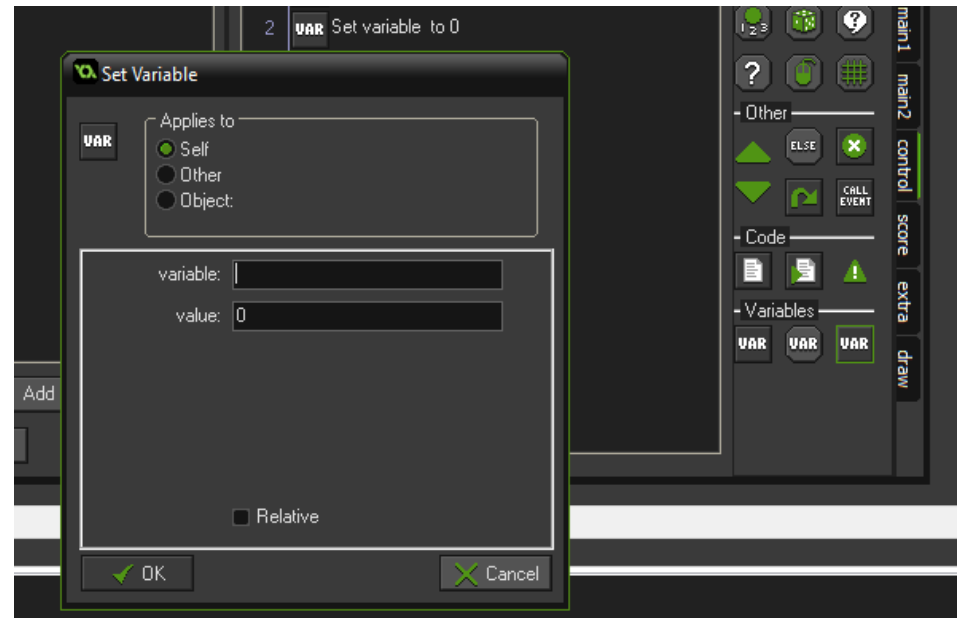


# Programando Eventos (bird)

- ▶ Insira para direção (direction) 270
  - ▶ Gravidade (gravity) 1.
  - ▶ Selecionado em Self e sem seleção do Relative. Aperte Ok.
  - ▶ O valor 1 indica a velocidade com o que o pássaro ira cair na direção 270.
  - ▶ Se quiser aumentar, aumente o valor (5 ou 6). Se quiser que ele caia mais lento, aproxime o valor do zero,(0,5, 0,3), etc.
  - ▶ Se colocar valor negativo, ele irá subir ao invés de cair.
- 

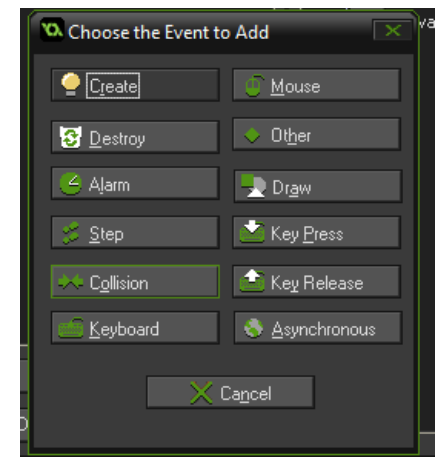
# Programando Eventos (bird)

- ▶ Vamos criar uma variável chamada angulo\_descida.
- ▶ No menu Control, pegue e arraste o Set Variable para Actions.
- ▶ Coloque o nome angulo\_descida
- ▶ Valor 0 (zero).



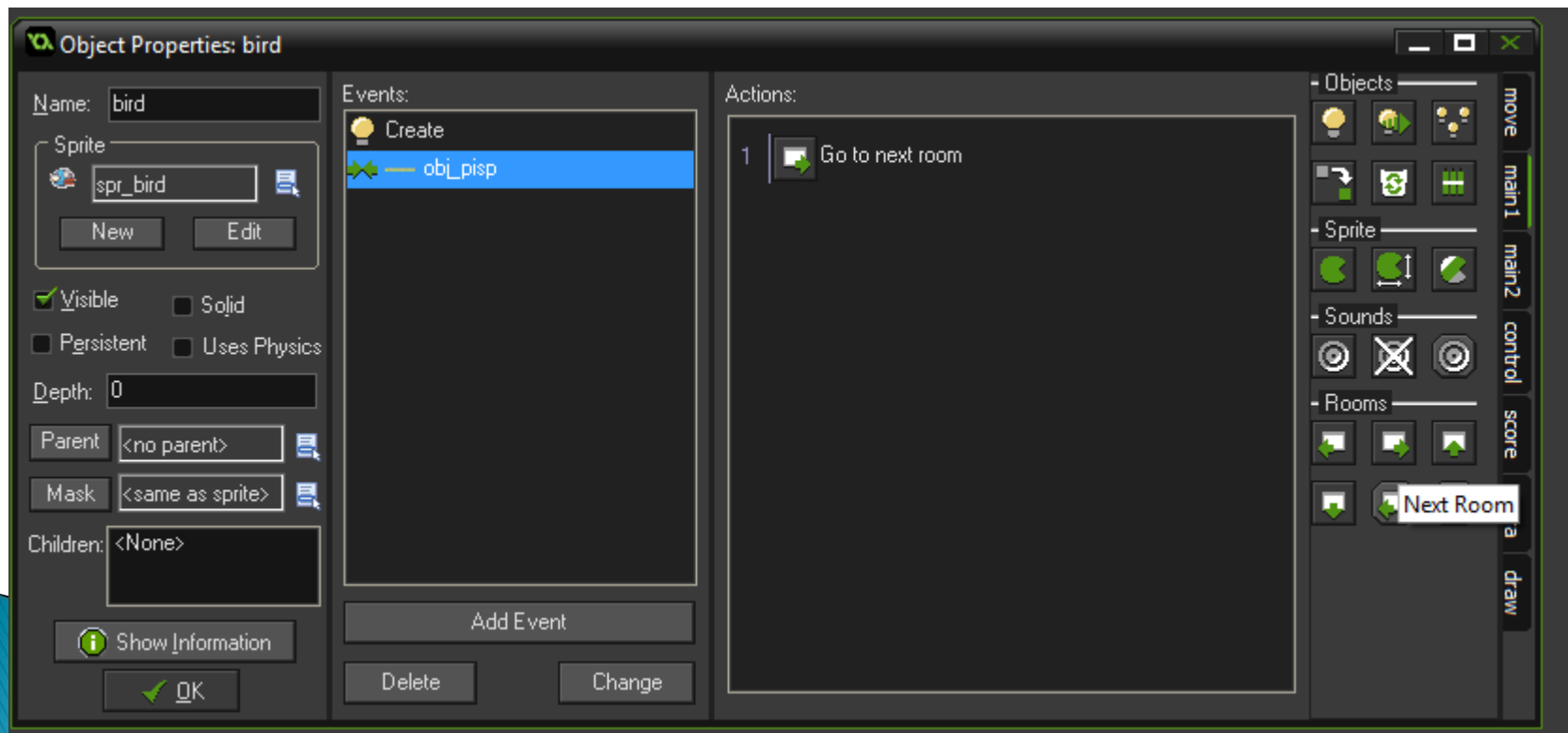
# Programando Eventos (bird)

- ▶ Crie um evento de colisão com o obj\_cano\_cima.
- ▶ Adicione um evento de Next Room. (menu Main1).
- ▶ Isso fará com que sejamos levados para a próxima room quando tiver um contato entre o pássaro e os canos. (já já falamos da room)



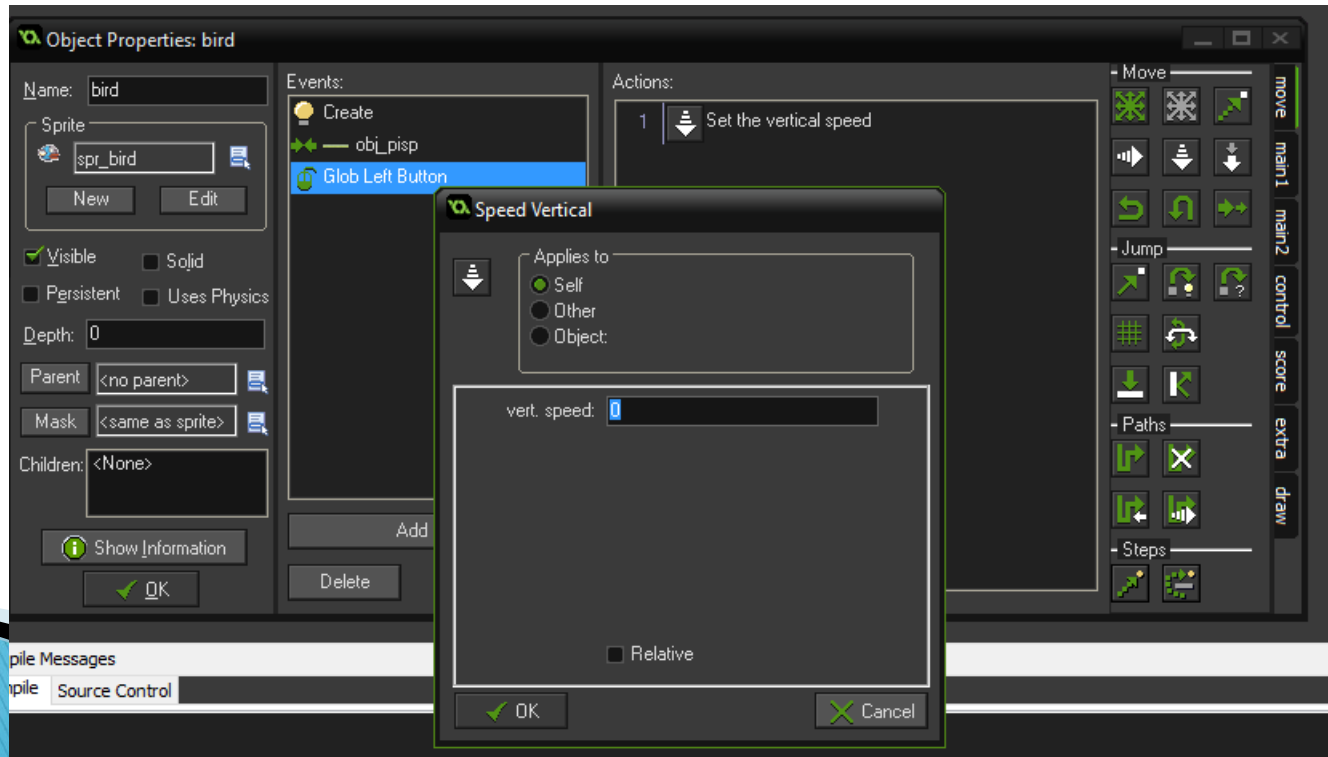
# Programando Eventos (bird)

- ▶ Crie um evento de colisão com o obj\_piso.
- ▶ Adicione um evento de Next Room (menu Main1). (idem do anterior)



# Programando Eventos (bird)

- ▶ Crie um evento Mouse, Global Mouse, GlobalLeftButton.
- ▶ Nas Actions, adicione um Set Vertical Speed para -10 em Self. (menu move).





# Programando Eventos (bird)

- ▶ Adicione um evento Step e vamos programar os eventos possíveis. (Add Event -> Step -> Step)
- ▶ Adicione um Test Variable (menu Control).
- ▶ O primeiro teste é saber se a velocidade vertical chegou a zero, se ela chegar iremos modificar o sprite. Então no TestVariable, coloque em variable: **vspeed**, em Value 0 e operação **equal to**.

# Programando Eventos (bird)

- ▶ Adicione um Start Block e um End Block (menu Control).
- ▶ Entre os blocks Start e End, adicione um Transform Sprite (menu Main1) e coloque 1 e 1 nos x scale e y scale, ângulo 0 e **no mirroring** em mirror.
- ▶ Logo depois do Transform Sprite, e antes do End Block, coloque um **Exit This Event** (menu control) para que ele encerre a execução das comparações que seguirão abaixo.

# Programando Eventos (bird)

- ▶ Vamos agora fazer uma outra comparação, com valores da velocidade vertical diferente de zero.
- ▶ Então vamos adicionar logo abaixo do EndBlock um novo teste de variável, (menu Control) e comparando a variável vspeed com value -7 e less than na operação.
- ▶ Colocamos como no teste anterior um Block Start e um Block End.
- ▶ Colocamos outro Transform Sprite (menu Main1), com valores xscale e yscale em 1, porém o valor do ângulo será colocado 20, para ele inclinar nosso pássaro.
- ▶ Coloque um Exit This Event (menu control) abaixo deste Transform Sprite e antes do Block End.

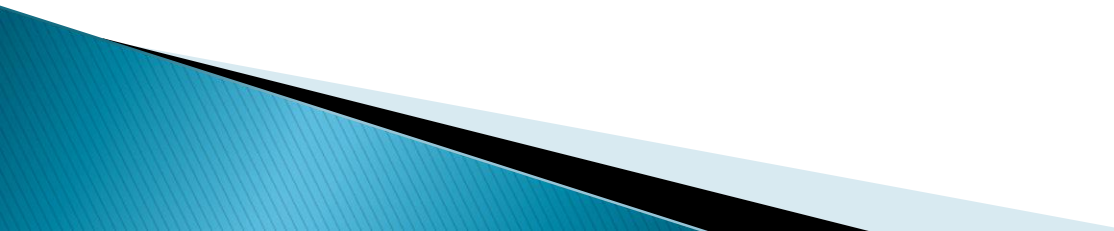
# Programando Eventos (bird)

- ▶ Repetiremos os passos agora para criar a mesma estrutura, com vspeed menor que -5, colocando ângulo 10 (é possível fazer ctrl-c e ctrl-v, mas atente ao block end que poderá ser deslocado para baixo, aí clique e arraste ele para a posição correta).
- ▶ Repetiremos os passos agora para criar a mesma estrutura, com vspeed menor que -2, colocando ângulo 5 (é possível fazer ctrl-c e ctrl-v, mas atente ao block end que poderá ser deslocado para baixo, aí clique e arraste ele para a posição correta).

# Programando Eventos (bird)

- ▶ Agora faremos a parte positiva. Teremos as mesmas comparações para vspeed positiva (2, 5 e 7) usando **greather than** ao invés de less than.
- ▶ Nestes casos, os ângulos estarão com o sinal negativo.
- ▶ Então vspeed greather than 2, angle -5, vspeed greather than 5, angle -10 e vspeed greather than 7, angle -20,

# Programando Eventos (pipe up)

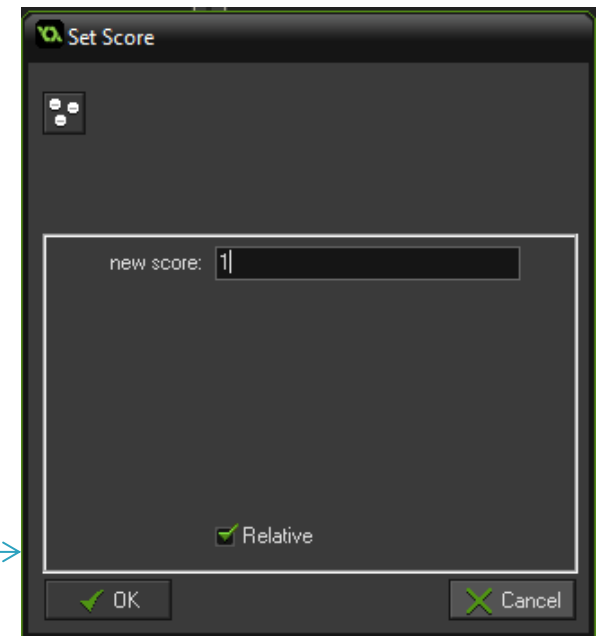
- ▶ Adicionar um evento Create
  - ▶ Adicione um Set Variable (menu control) e crie uma variável **hspeed** com value -5.
  - ▶ Adicione um evento Step. Neste evento vamos criar a engine (motor) que irá fazer com que os canos sejam criados e apagados sozinhos conforme iremos jogando.
  - ▶ Os canos irão sendo colocados na direita da fase, passarão da direita para esquerda, e quando saírem da fase serão apagados.
- 

# Programando Eventos (pipe up)

- ▶ No Step, adicione em Actions um Test Variable (menu control), coloque x em variable, -10 em value, **less than** na comparação.
- ▶ Adicione um Start Block e um End Block (menu control).
- ▶ Dentro deste bloco, vamos adicionar o comando para destruir o objeto (Destroy Instance – menu Main1).

# Programando Eventos (pipe up)

- ▶ Quando isso acontecer, iremos adicionar 1 ponto no Score (placar) do jogo.
- ▶ Para isso, vá ao menu Score e adicione um Set Score, no value coloque 1 e marque a opção **relative** abaixo para que ele possa somar o score com o score anterior.





# Programando Eventos (pipe down)

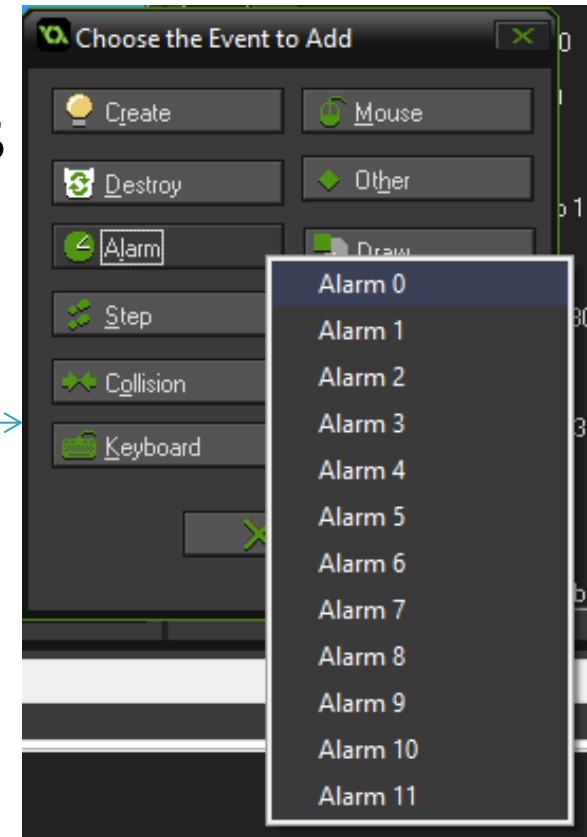
- ▶ Para o Cano\_baixo, iremos fazer a mesma coisa que o Step do cano\_cima faz, com uma excessão, iremos retirar a contagem do Score, senão serão adicionados 2 pontos quando ele passar pelo obstáculo. Selecione todas actions do step do cano\_cima;
- ▶ Faça ctrl+c, clique ok, abra o obj cano\_baixo, adicione Step, em actions, ctrl+v. clique no set score da action e aperte delete.

# Onde a mágica acontece? (control)

- ▶ Vamos criar um objeto chamado **obj\_contole**. Adicione o **SPR\_visualiza**. Não desative a propriedade Visible.
- ▶ Criar um evento Create neste objeto.
- ▶ Neste evento, vamos adicionar um Set Alarm (menu Main2).
- ▶ Em Number of steps coloque 60 e deixe in alarm em **Alarm0**.
- ▶ Assim criaremos um timer de 60 passos (+ ou - 2 segundos).

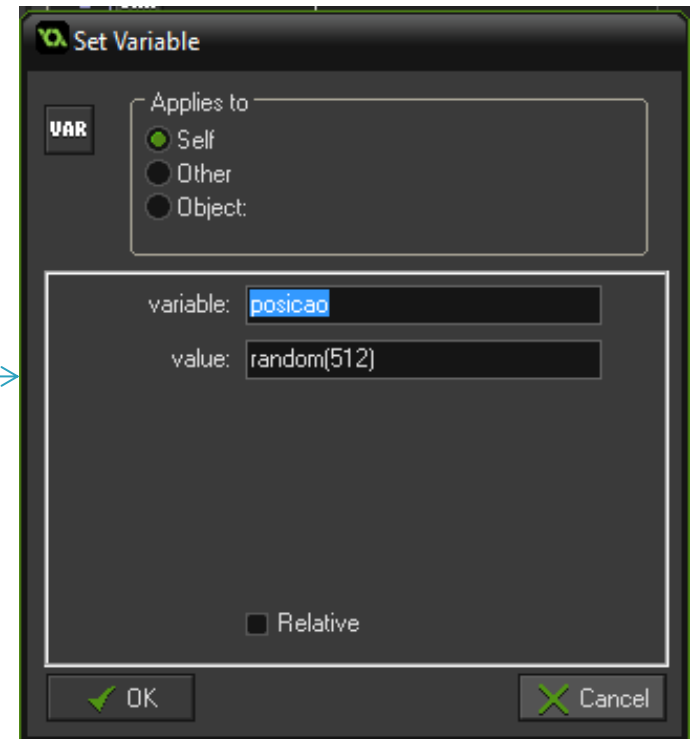
# Onde a mágica acontece? (control)

- ▶ Ainda no Create, crie um Set Variable (menu control) chamado canos e coloque o valor 0
- ▶ Adicione um evento Alarm0.
- ▶ Aqui que iremos fazer os canos serem desenhados.



# Onde a mágica acontece? (control)

- ▶ Coloque em Actions um Set Variable (menu control) com o nome **posicao** e atribua em value: random(512).
- ▶ Será sorteado um valor de 1 a 512 e colocado nesta variável.



# Onde a mágica acontece? (control)

- ▶ Adicione outra Set Variable. Nome **espalhar**, valor **70**, que será utilizada para espalhar os canos. Quanto maior o valor, mais fácil, quanto menor, mais difícil.
- ▶ Agora vamos testar a posição.
- ▶ Coloque um **Test Variable** (menu control)
- ▶ Em Variable coloque **posicao**, em valor coloque **110** e operação **less than**.
- ▶ Isto testa quando a posição sorteada for menor que 110, qual ação iremos fazer em seguida, para que ele não saia da tela na parte inferior. Coloque um abre bloco e um fecha bloco.

# Onde a mágica acontece? (control)

- ▶ Colocaremos um Set Variable, com variable posicao e value 110 para manter em 110 caso ela seja menor do que 110.
- ▶ Faremos a mesma coisa agora para a parte de cima, comparando com um Test Variable, quando a Variavel posicao estiver com valor maior que 380 (value 380, operação greather than).
- ▶ Colocaremos um Set Variable, com variable posicao e value 380 para manter em 380 caso ela seja maior do que 380.

# Onde a mágica acontece? (control)


- ▶ Vamos colocar um Set Alarm 0 com stpes 60 no Alarm0 para darmos início ao processo de checagem novamente dentro de 60 passos, analisando tudo de novo (criando um loop).
- ▶ Vamos criar uma instância do objeto obj\_cano\_cima. No Menu Main1, adicione um Create Instance.
- ▶ Em object, selecione o obj\_cano\_cima.
- ▶ Em x, coloque room\_width (para ser criado no começo da nossa room, no lado direito).
- ▶ E no y coloque posicao-espalhar , que são nossas duas variáveis que irão posicionar as alturas dos canos.

# Onde a mágica acontece? (control)

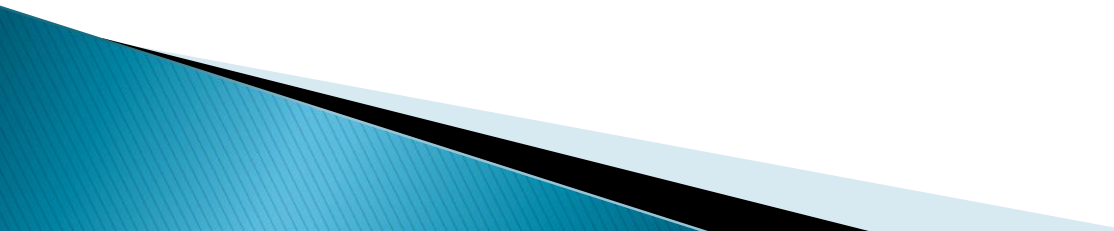
- ▶ Repita o mesmo processo para o `obj_cano_baixo`, adicionando uma `Create Instance` nova, em `object`, `obj_cano_baixo`, em `x: room_width` e em `y, posicao+espalhar`.
- ▶ Adicione um `Evento Draw`. Neste evento iremos desenhar os pontos na tela.
- ▶ (Opcional) Adicione um `Set Color` (menu `Draw`) nas `Actions` para escolhermos a cor que os pontos terão quando desenhados. Selecione a cor que desejar.



# Onde a mágica acontece? (control)

- ▶ (Opcional) Podemos criar uma configuração de fonte. Aperte o botão para criar Fonte (o TT na parte de cima). Selecione o nome para fnt\_placar, selecione Arial e tamanho 24. Clique em Ok.
  - ▶ (apenas se criou a Font) Agora adicione a font no evento Draw. Selecione a fnt\_placar.
  - ▶ No menu Score, adicione Draw Score para que seja mostrado o placar. Em x coloque 124, em y coloque 60. Em Caption apague o texto de deixe em branco, sem nada, para mostrar os pontos como no jogo original no meio da tela.
- 

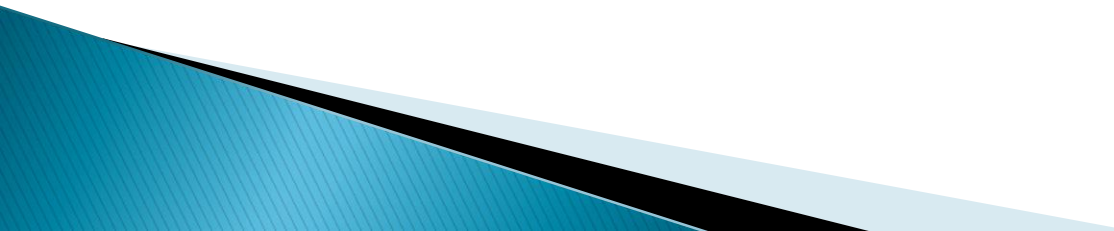
# Criando os últimos objetos (start)

- ▶ Adicione um novo objeto, obj\_flappystart. Coloque o spr\_flappystart associado a ele.
  - ▶ Adicione um evento Mouse, Global Mouse, Global left button.
  - ▶ No menu Main1, coloque em Actions um NextRoom. Quando o usuário clicar ele irá começar o jogo.
  - ▶ Crie dois objetos chamados obj\_getready e obj\_tap apenas para inserir as sprites respectivas, sem nenhum evento ou ação.
- 

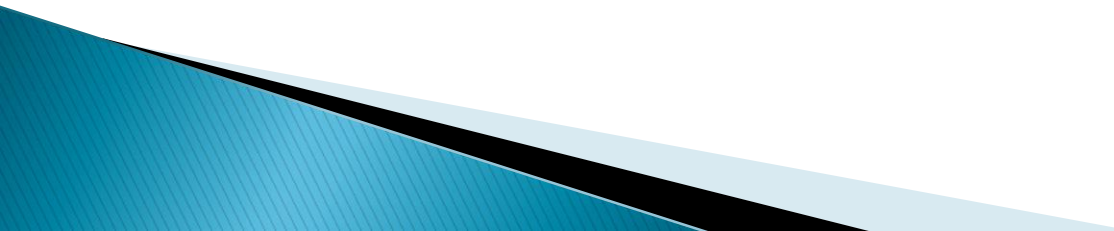
# Criando os últimos objetos (end)

- ▶ Vamos criar um objeto chamado `obj_gameover`, que será responsável por reconfigurar o jogo quando perdermos e mostrará o placar final.
- ▶ Coloque o `SPR_visualiza` no Sprite.
- ▶ Adicione um evento Mouse, Global Mouse, Global left button.
- ▶ No menu `Main2`, coloque em Actions um `Restart Game`. Quando o usuário clicar ele irá recomeçar o jogo.

# Criando os últimos objetos (end)

- ▶ Adicione um evento Draw
  - ▶ No menu Draw, coloque em Actions um Draw Scaled Text. Em text, escreva o texto informando que o jogo terminou, por exemplo GAME OVER. Em x coloque 40, em y coloque 50.
  - ▶ No menu Score, adicione um Draw Score, x 90, y 80, Caption Placar final: para mostrar a pontuação final do jogo.
- 

# Rooms – O palco

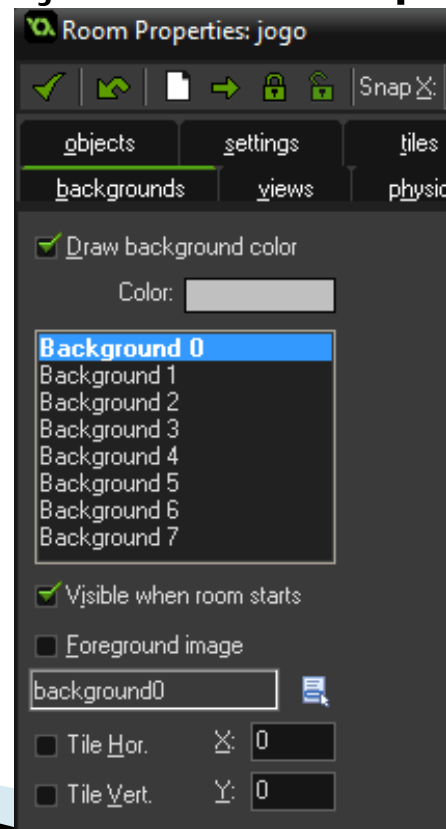
- ▶ Vamos criar as Rooms (onde o jogo acontece, o palco ou fase do jogo).
  - ▶ Adicione três novas Rooms.
  - ▶ O Game Maker trabalha com a ordem de cima para baixo. Portanto a primeira room (de cima) será a abertura do jogo, a segunda room (do meio) será o jogo em si e a terceira room (embaixo) será a game over.
  - ▶ Podem ser renomeados se desejar.
- 

# Room2 – O jogo

- ▶ Inicialmente, vamos configurar a room do jogo (do meio). Dê dois cliques nela para abrí-la.
- ▶ Clique no menu Views. Em View0, selecione o Enabled use of view (em cima) e visible when room starts (embaixo).
- ▶ Em View in Room, mantenha X e Y com zero, em W coloque 288 e H coloque 512 (dimensões de nosso background).
- ▶ Em Port on Screen, X e Y mantenha com zero, em W coloque 288, em H coloque 512.

# Room2 – O jogo

- ▶ Selecione a aba Background.
- ▶ Selecione o Background (embaixo do quadro).  
Remova as seleções das opções Tile Vert. e Tile Horz.



# Room2 – O jogo

- ▶ Settings, podemos colocar o nome da room, (jogo)
- ▶ Width é a largura da tela(288) e Height a altura da tela (512).
- ▶ Em cima, na tela, SnapX preencha com 10, Snapy com 10.



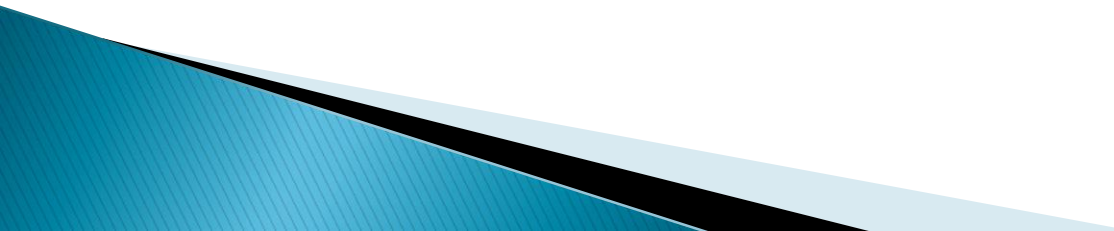
# Room2 – O jogo

- ▶ Selecione a aba Objects.
- ▶ No quadrado, se clicar com o mouse, poderá escolher qual objeto selecionará para adicionar ao jogo.
- ▶ Selecione o obj\_flappybird e, com o mouse, dê um (**E SOMENTE UM**) clique dentro da room onde ele deverá aparecer quando o jogo iniciar.
- ▶ Clique na parte branca do Background (nuvens) para *setar* a posição inicial do obj\_flappybird. (OBS não clique duas vezes para não adicionar 2 ou mais objetos. Dê um clique único).
- ▶ Pode definir a posição manualmente também se quiser. Position X, coloque 40 e em Y 340.

# Room2 – O jogo

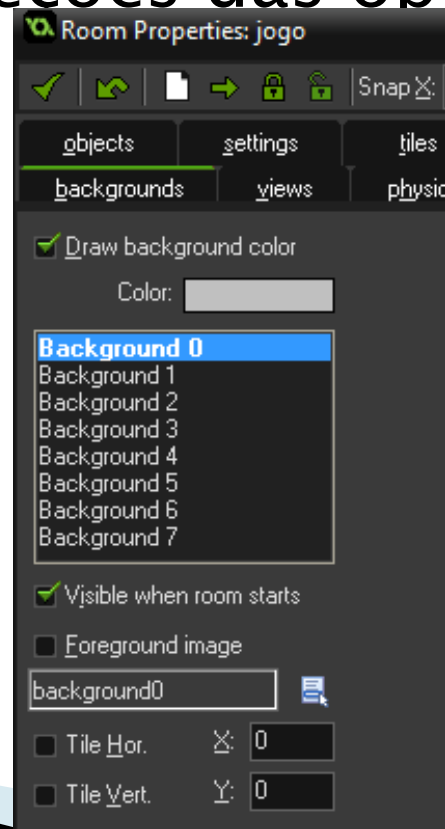
- ▶ Adicione também um obj\_controle em qualquer local (ele não será visível durante o jogo).
- ▶ Adicione um objeto Piso na parte debaixo. Em Position X coloque 140, Y 500.

# Room1 – Abertura

- ▶ Selecione a primeira room, inicial. Repita os mesmos passos
  - ▶ Clique no menu Views. Em View0, selecione o Enabled use of view (em cima) e visible when room starts (embaixo).
  - ▶ Em View in Room, mantenha X e Y com zero, em W coloque 288 e H coloque 512 (dimensões de nosso background).
  - ▶ Em Port on Screen, X e Y mantenha com zero, em W coloque 288, em H coloque 512.
- 

# Room1 – Abertura

- ▶ Selecione a aba Background.
- ▶ Selecione o Background (embaixo do quadro).  
Remova as seleções das opções Tile Vert. e Tile Horz.



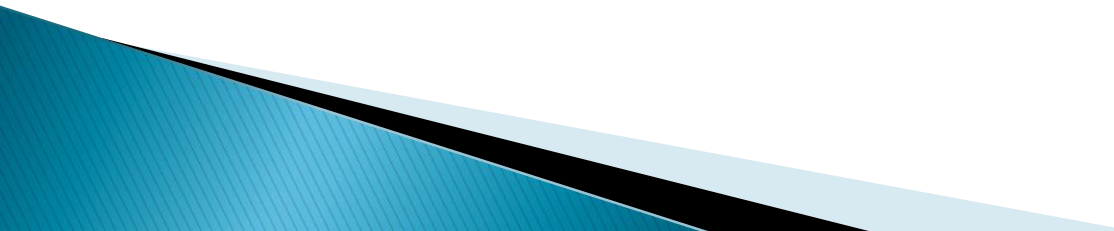
# Room1 – Abertura

- ▶ Settings, podemos colocar o nome da room, (jogo)
- ▶ Width é a largura da tela(288) e Height a altura da tela (512).
- ▶ Em cima, na tela, SnapX preencha com 10, Snapy com 10.

# Room1 – Abertura

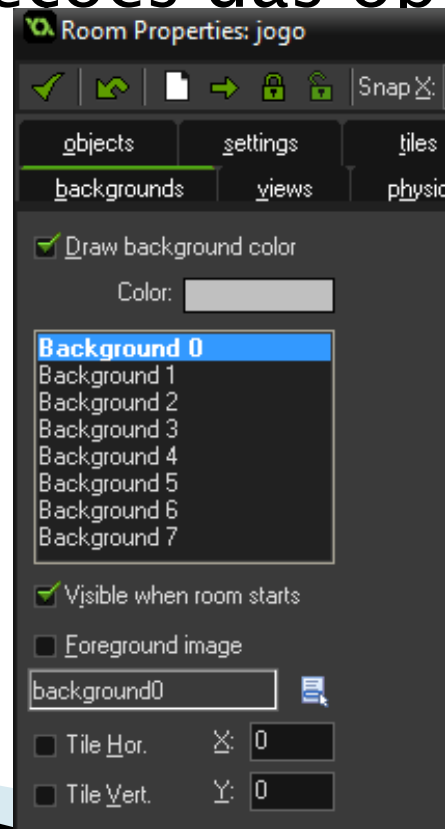
- ▶ Adicione o obj\_flapystart em qualquer local da room. (centro de preferência)
- ▶ Adicione os objetos sinais que mostram o obj\_tap e o obj\_getready.

# Room3 – Game Over

- ▶ Selecione a primeira room, inicial. Repita os mesmos passos
  - ▶ Clique no menu Views. Em View0, selecione o Enabled use of view (em cima) e visible when room starts (embaixo).
  - ▶ Em View in Room, mantenha X e Y com zero, em W coloque 288 e H coloque 512 (dimensões de nosso background).
  - ▶ Em Port on Screen, X e Y mantenha com zero, em W coloque 288, em H coloque 512.
- 

# Room3 – Game Over

- ▶ Selecione a aba Background.
- ▶ Selecione o Background (embaixo do quadro).  
Remova as seleções das opções Tile Vert. e Tile Horz.





# Room3 – Game Over

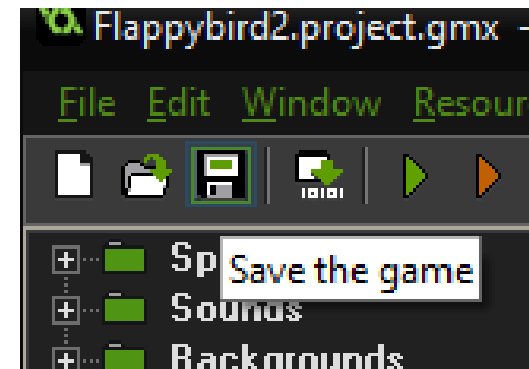
- ▶ Settings, podemos colocar o nome da room, (jogo)
- ▶ Width é a largura da tela(288) e Height a altura da tela (512).
- ▶ Em cima, na tela, SnapX preencha com 10, Snapy com 10.

# Room3 – Game Over

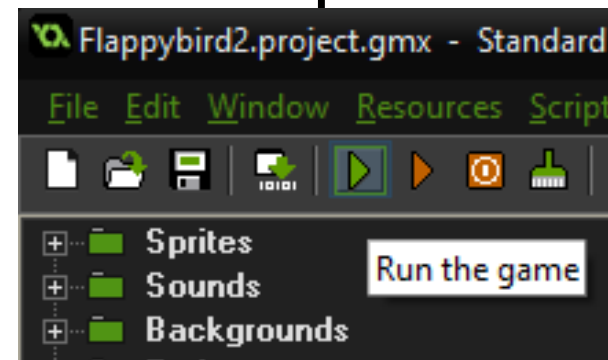
- ▶ Adicione um obj\_tap
- ▶ Adicione o obj\_gameover.

# Pronto!

- ▶ Pronto, seu jogo está pronto .
- ▶ Vamos salvar o jogo, apertando o 3º botão – desenho de um disquete.



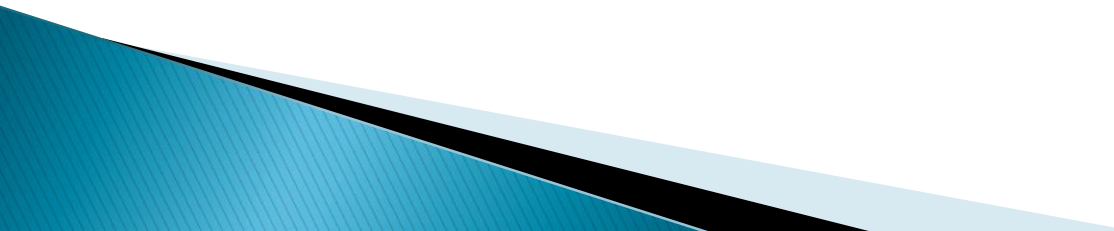
- ▶ Agora Vamos testar? Apertaremos o botão Run this Game, 5º botão, uma seta para frente verde.



# BUG – Passar por cima???

- ▶ Ainda temos um ajuste a fazer no flappybird globalleft button.
- ▶ Vamos colocar um Test Variable (menu Control). Coloque em variable: **obj\_flappybird.y** e em Value 40. Na Operation, coloque Less Than.
- ▶ Abra e feche um novo bloco de comandos.
- ▶ Dentro do bloco coloque um Set Variable (Menu Control) e em variable: **obj\_flappybird.y** e value coloque 40.
- ▶ Assim, se o pássaro atingir a altura máxima da tela ele não sairá da área do jogo. Testa!

# BONUS – Som

- ▶ Podemos agora inserir áudio em nosso jogo.
  - ▶ Clique no botão adicionar som. Dê o nome `snd_bird`. Aperte o botão para selecionar arquivos e selecione o arquivo `flap.wav` e selecione OK.
  - ▶ Repita o processo a selecione o som `smack.wav`. Dê o nome de `snd_bateu`.
  - ▶ Mais uma vez, repita o processo, selecione o som `bading.wav` e dê o nome de `snd_pontos`.
- 

# BONUS – Som

- ▶ Abra o obj\_flappybird. Na colisão com o obj\_canocima, adicionar o som snd\_bateu, clicando no botão Main1, arrastando o PlaySount e selecionando o snd\_bateu.
- ▶ Deixe o loop em false.
- ▶ Faça o mesmo para a colisão com o obj\_piso.
- ▶ E no evento GlobalLeft Button do mouse do obj\_flappybird, vamos colocar um teste de som, para saber se já está sendo tocado o som do pássaro e só acioná-lo se não tiver sido iniciado ainda. Coloque um Check Sound, selecione o snd\_bird e acione o botão NOT abaixo. adicione um som e coloque o snd\_bird.

# BONUS – Som

- ▶ Abra o obj\_canocima, no evento Step, depois de destruir a instancia dos canos, adicione um som (do menu Main1) e coloque o som snd\_pontos.

# Finish Him!

- ▶ É isso aí. Existe muitas outras possibilidades. Existem muitas outras propriedades e objetos a serem descobertos
- ▶ Pra quem gosta de programar (EM C) é possível criar um jogo com o GameMaker apenas com programação (sem clicar e arrastar). É possível ter um controle maior sobre variáveis, comandos, blocos, etc.
- ▶ File – Create Application cria o setup do jogo.
- ▶ Espero que tenham gostado! Até o próximo!
  - ▶ *Prof. Sérgio Portari – [portari@gmail.com](mailto:portari@gmail.com)*