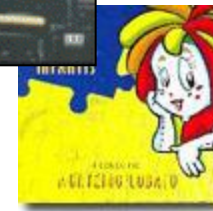
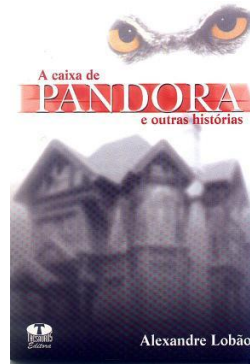


# Introdução à Criação de Jogos com GameMaker

Alexandre S. Lobão  
[contato@alexandrelobao.com](mailto:contato@alexandrelobao.com)

# Quem sou eu?



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

# Motivação e Objetivos

- ❧ Conhecer o processo de planejamento de jogos
- ❧ Ter uma visão geral de ferramentas disponíveis e grau de esforço para criação de jogos
- ❧ Explorar funcionalidades básicas do GameMaker
- ❧ Completar jogos básicos em tutoriais e exercícios
- ❧ Ter uma visão geral sobre funcionalidades avançadas
- ❧ Conhecer onde conseguir mais informações
- ❧ Criar um novo jogo (“projeto final”)
- ❧ Preparar os alunos para o curso avançado de criação de jogos

# Foco do curso

- 🐜 Entender conceitos básicos sobre programação de jogos
  - 🐜 Desenvolver as habilidades para criar jogos simples
  - 🐜 Passar conhecimento para que vocês possam evoluir, criando jogos mais sofisticados
- 
- 🐜 Muita teoria nos dias 1 e 2, para “dar bagagem”
  - 🐜 Muita prática nos dias 3 e 4, para “dar experiência”

# Como você irá aprender...

- 🐜 Baseado em suas próprias experiências
- 🐜 Trabalhando em grupos pequenos
- 🐜 Discussões abertas com toda a classe
- 🐜 Exercícios
  - 🐜 A cada dia, menos teoria e mais prática!

*Esteja preparado para trocar idéias!*

# Agenda – Dia 1

- 🐛 **Histórico:** A História dos Vídeo Games
- 🐛 **Visão Geral:** Componentes de um jogo e seu relacionamento
- 🐛 **Introdução ao GameMaker**
- 🐛 **Conceito:** Nomes de componentes, Sistema de Coordenadas
- 🐛 **Exercício:** “Simple Breakout” - Criando seu primeiro jogo
  - 🐛 Conceitos: Objetos, sprites, eventos, uso do mouse

# Agenda – Dia 2

- 🐜 **Como planejar um jogo**
- 🐜 **Eventos no GameMaker**
- 🐜 **Ações no GameMaker**
- 🐜 **Conceitos: Objetos, Instâncias e Variáveis**
- 🐜 **Exercício: River Raid**
  - 🐜 Conceitos: variáveis, uso de texto, background móvel (scrolling), uso do teclado, controle de energia e número de vidas, posição absoluta e relativa de objetos

# Agenda – Dia 3

- 🐜 **Conceitos Avançados no GameMaker**
- 🐜 **Gêneros de Jogos**
- 🐜 **Como planejar um jogo**
- 🐜 **Palavras dos Gurus:** “Como ser um desenvolvedor de jogos”, tradução da palestra de David Weller, gerente de programas das comunidades de desenvolvimento de jogos da Microsoft Game Technology Group
- 🐜 **Definição do projeto final** (individual por aluno)
- 🐜 **Exercício: Dungeon Dude**
  - 🐜 Conceitos: Uso avançado de sprites e variáveis, informação sobre o jogo (game info), criação de inimigos “inteligentes” (escolha de caminho), controle de objetos carregados, mudança de fases



# Agenda - Dia 4

- 🐜 **Visão Geral da versão registrada**
  - 🐜 Programa RegDemo
- 🐜 **Ferramentas para desenvolvimento de jogos**
- 🐜 **Palavras dos Gurus:** “Um guia para criar jogos de sucesso”, tradução da palestra de Bruce Shelley, do Ensemble Studios (criadores de sucessos como a série “Age of Empires”)
- 🐜 **Projeto Final**
- 🐜 **Próximos passos**

# Atividade

## 🐉 Apresentações e expectativas

### 🐉 Apresente-se

- 🐉 Nome

- 🐉 Profissão, Hobbies

- 🐉 Experiência com programação de jogos

### 🐉 Expectativas sobre o curso



# Introdução à Criação de Jogos com GameMaker Dia 1

Alexandre S. Lobão  
[contato@alexandrelobao.com](mailto:contato@alexandrelobao.com)

# Agenda – Dia 1

- 🐛 **Histórico:** A História dos Vídeo Games
- 🐛 **Visão Geral:** Componentes de um jogo e seu relacionamento
- 🐛 **Introdução ao GameMaker**
- 🐛 **Conceito:** Nomes de componentes, Sistema de Coordenadas
- 🐛 **Exercício:** “Simple Breakout” - Criando seu primeiro jogo
  - 🐛 Conceitos: Objetos, sprites, eventos, uso do mouse

# Histórico

- 1951: A Marufuku, fabricante de cartas de baralho, muda de nome para **Nintendo** Playing Card Company. Nintendo significa "deixe a sorte para os céus".
- 1954: David Rosen, um veterano de guerra americano, percebe o sucesso de máquinas mecânicas alimentadas por moedas nas bases militares americanas e funda sua empresa, exportando essas máquinas para o Japão. Depois ele resolve fabricar suas próprias máquinas com jogos, comprando fábricas de jukeboxes e caça-níqueis japonesas. Nasce, então, a Service Games, ou **SEGA**.

# Histórico

🐛 1958: O Brookhaven National Laboratories resolve abrir suas portas para visitação para mostrar que não havia testes para criar animais mutantes como temia a população, influenciada por filmes de ficção científica. Para evitar que as pessoas ficassem muito entediadas durante a visita, Willy Higinbotham cria um jogo interativo chamado "Tennis for Two" que simulava um ping-pong, considerado o primeiro videogame da História.

# Histórico

🐉 Willy Higinbotham e seu "Tennis for Two"



# Histórico

👾 1961: Steve Russell, estudante do MIT, cria Spacewar, primeiro jogo de computador da História.





# Histórico

- 1970: Nolan Bushnell, com a ajuda de Ted Dabney, consegue criar o primeiro arcade da História, uma versão do Spacewar a que ele dá o nome de Computer Space. A Nutting Associates compra a máquina de Bushnell e o contrata.
- 1971: A Nutting Associates lança o Computer Space, com um visual futurista. As pessoas acham muito difícil de jogar e o jogo torna-se um fracasso.

# Histórico

## 🚗 O Computer Space

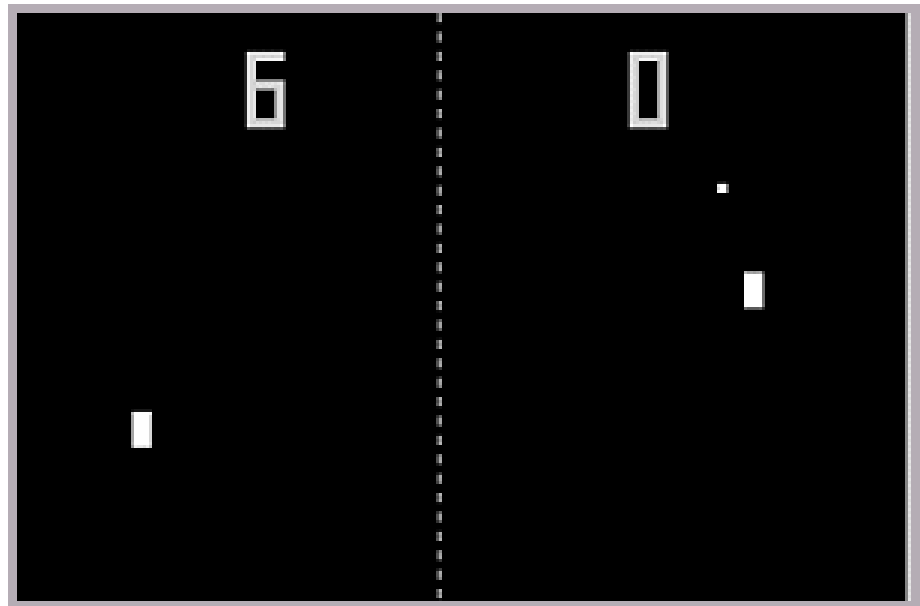


# Histórico

- ❧ 1972: Com o fracasso do Computer Space, Bushnell resolve tentar fazer jogos mais fáceis de se jogar. Apesar do sinal verde da Nutting, ele pede um terço de todo o lucro. Com a negativa, ele se demite e, junto com Ted Dabney, resolve fundar sua própria empresa, originalmente chamada de Syzygy. Como este nome já está sendo utilizado por outra empresa, resolve mudar o nome para **Atari** (o “cheque” do jogo japonês Go).

# Histórico

- 1972: Bushnell contrata Al Alcorn, que programa um jogo simples de ping-pong. Nasce o “Pong”, assim batizado pois “ping-pong” já era registrado.



# Histórico

👾 1972: A Magnavox apresenta seu Odyssey, criado pelo engenheiro Ralph Baer, numa convenção em Burlingame, California. Mais de 100.000 unidades são vendidas.



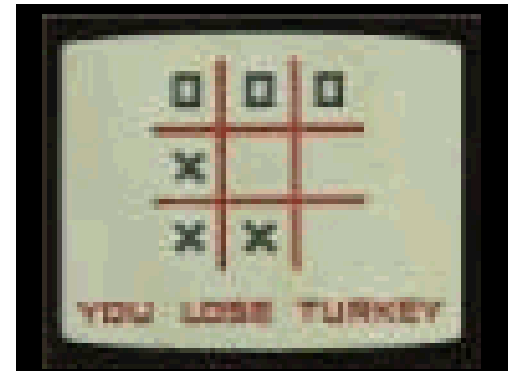
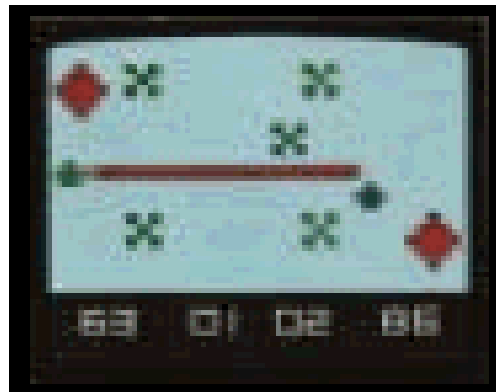
# Histórico

🎮 1973 a 1975: O sucesso do Pong faz com que surja uma enxurrada de “clones”. Empresas como Sega, Taito, Midway, Coleco e a própria Atari lançam diversas variações do jogo original.



# Histórico

- 1976: A Fairchild Camera & Instrument lança o Channel F. Nasce o primeiro console da História, o primeiro “programável” (com cartuchos).





# Histórico

- 1977: A história dos video-games mudaria para sempre depois que a Atari lança, no Natal, o Video Computer System (Atari VCS), mais tarde rebatizado de Atari 2600. O sistema era chamado internamente de Stella.





# Histórico

- 1978: A japonesa Taito fabrica o arcade “Space Invaders” e a Midway importa a máquina para os EUA. O jogo inova ao introduzir os “high-scores” (recordes) e se torna um imenso sucesso.



# Histórico

🎮 1978: A Magnavox lança o Odyssey2, cuja novidade é um teclado de membrana embutido no console.



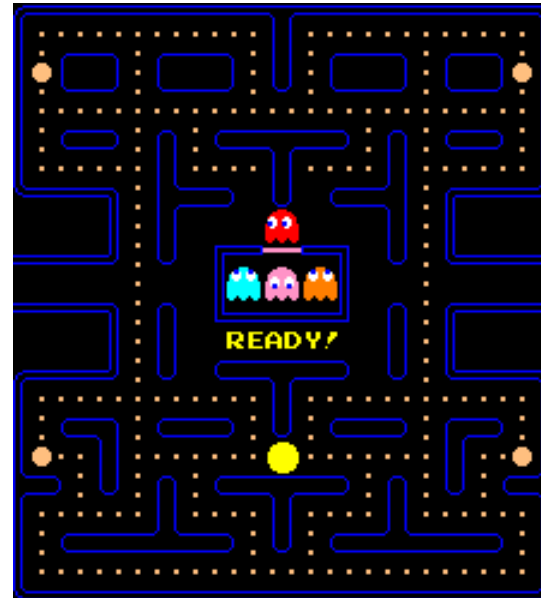
# Histórico

- 1980: A Mattel lança o Intellivision, mais avançado que o VCS. Apesar de ser a única competição séria que a Atari enfrenta, não evita que o remodelado e agora batizado de Atari 2600 domine o mercado, com mais de 25 milhões de consoles e 120 milhões de cartuchos vendidos, 200 jogos disponíveis de 40 fabricantes diferentes e lucro de US\$ 5 bilhões.



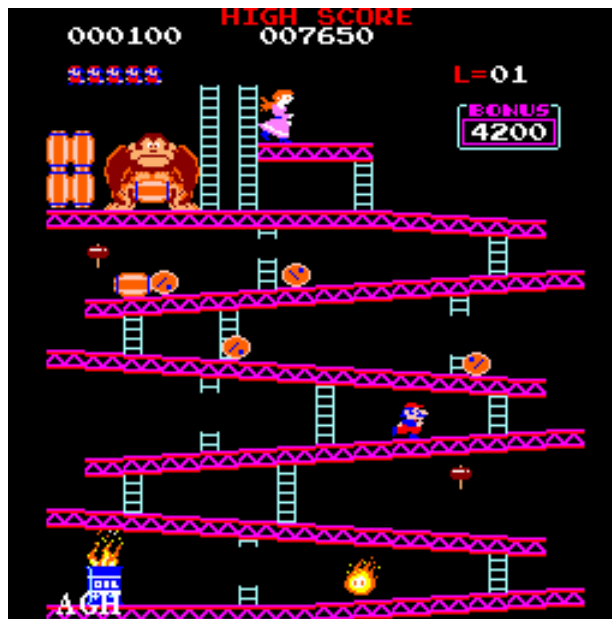
# Histórico

- ❧ 1980: A Namco lança o Pac-Man, o arcade de maior sucesso de toda a História. O jogo era inicialmente chamado de Puck-Man, mas foi alterado pois temia-se que vândalos mudassem o nome para Fuck-Man nos fliperamas, riscando o 'P'.



# Histórico

- 1981: Depois de inúmeros fracassos, a Nintendo consegue um grande sucesso: Donkey Kong, de Shigeru Miyamoto. O herói, inicialmente batizado de “jumpman”, acabou conhecido pelo seu apelido de Mario, dado pelos membros da Nintendo por sua semelhança com Mario Segali, seu chefe.



# Histórico

- ❌ 1982: A Coleco lança o Colecovision, um console muito superior ao Atari 2600. Nintendo, Konami e Sega são alguns dos desenvolvedores de jogos para o console.
- ❌ 1983: A Commodore lança o Commodore 64, computador melhor que qualquer console de então.
- ❌ 1983: A Nintendo lança seu console, o Famicom (Family Computer) no Japão.

# Histórico

👾 1985: Alex Pajitnov cria o Tetris, um jogo simples que se tornaria um enorme sucesso até hoje.



# Histórico

- 🐛 1986: Em mais um ano pra entrar pra História dos videogames, a Nintendo lança a versão americana do Famicom, o NES (Nintendo Entertainment System) em todos os EUA. O console vem com o jogo Super Mario Bros, que se torna um sucesso instantâneo e um clássico.
- 🐛 1986: Seguindo o sucesso do NES, a Atari lança o seu anunciado 7800, que se torna um grande fracasso.



# Histórico

## 🐉 NES x Atari 7800



# Histórico

👾 1986: Para não ficar atrás, a Sega também lança nos EUA o seu console, o Master System, para competir com a Nintendo.



# Histórico

- 🎮 1987: O NES é um grande sucesso e a Nintendo domina o mercado. Os consoles da Atari são praticamente ignorados e apenas a Sega consegue manter alguma competição.
- 🎮 1987: A NEC lança o PC-Engine no Japão.

# Histórico

🐛 1989: Num ano com muitas novidades, a Nintendo lança o GameBoy, a NEC lança o PC-Engine nos EUA com o nome de TurboGrafx-16, a Sega lança o Genesis e a Atari lança o Lynx.



# Histórico

- ❌ 1990: A Nintendo lança no Japão o Super Famicom, seu console de 16 bits, um sucesso imediato
- ❌ 1990: A Nintendo lança o jogo Super Mario Bros 3 para o NES, que se torna o jogo de console mais vendido de todos os tempos.
- ❌ 1990: A SNK lança o NeoGeo, de 24 bits, o console mais avançado da época. Seu alto preço, contudo, o torna um verdadeiro fracasso de vendas.

# Histórico

- 1991: A Nintendo lança o Super NES nos EUA.
- 1991: A Sega cria o personagem Sonic na esperança de competir com o Mario da Nintendo.
- 1991: A Capcom lança o revolucionário arcade Street Fighter II.
- 1993: A Panasonic lança o 3DO e a Atari o Jaguar (primeiro de 64 bits). Apesar de caros, ambos começam vendendo bem, mas depois fracassam com a concorrência da Sony, Nintendo e Sega.

# Histórico

- 1994: A Nintendo lança o Donkey Kong Country para o SNES, um grande sucesso, e anuncia planos para lançamento de um console de 64 bits.
- 1995: A Sega lança o Saturn e a Sony o PlayStation.





# Histórico

- ❌ 1996: O PlayStation é considerado o melhor console existente e tem um sucesso estrondoso.
- ❌ 1996: Amargando grandes prejuízos e cada vez mais perdendo mercado, a Nintendo finalmente lança seu novo console, o Nintendo 64.
- ❌ 1996: Acumulando fracasso após fracasso, morre a Atari, a maior precursora dos videogames.



# Histórico

- 1998: A Nintendo lança o GameBoy Color, versão colorida do velho GameBoy, que faz um enorme sucesso, apesar de simples e ultrapassado.
- 1998: A Sega lança o Dreamcast no Japão.
- 1999: O Sega Dreamcast é lançado com sucesso nos EUA, vendendo US\$ 100 milhões em 24 horas.
- 2000: O PlayStation 2 é lançado no Japão, estabelecendo um recorde ao vender um milhão de unidades em dois dias. A demanda é muito maior que a oferta e faltam consoles nas lojas.

# Histórico

- 2001: A Microsoft lança o XBox.
  - 16 jogadores com 4 consoles ligados
  - XBox Live!

- 2001: A Nintendo lança o Game Boy Advance e o GameCube, seu novo console de 128 bits.



# Histórico

🐜 2005: Xbox 360 da Microsoft

🐜 2006: Playstation 3 da Sony



# O Futuro?

- ❌ Microsoft e Sony anunciam que estão estudando para que seus próximos consoles sejam integrado a eletrodomésticos e outros aparelhos da casa.
- ❌ Jogos para dispositivos *wireless* e aparelhos portáteis ganham cada vez mais força.
- ❌ Mercado de consoles praticamente dominado por Sony, Nintendo e Microsoft.
- ❌ Jogos multiplayer pela Internet vêm sendo a grande prioridade dos desenvolvedores de jogos.

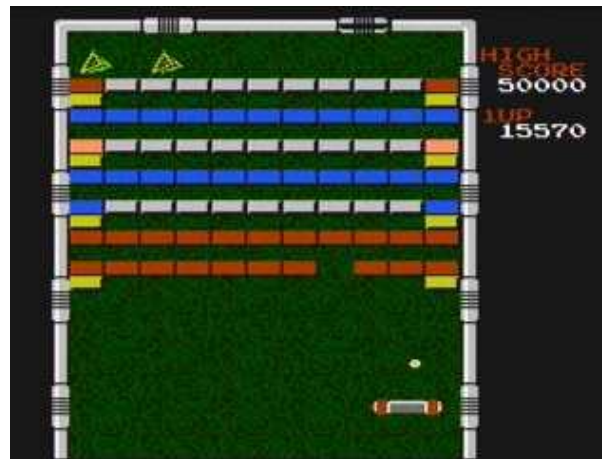
# Componentes de um jogo e seu relacionamento

Objetos   

Imagem = Sprites

Fases (ou “salas”)



















Imagem = Background



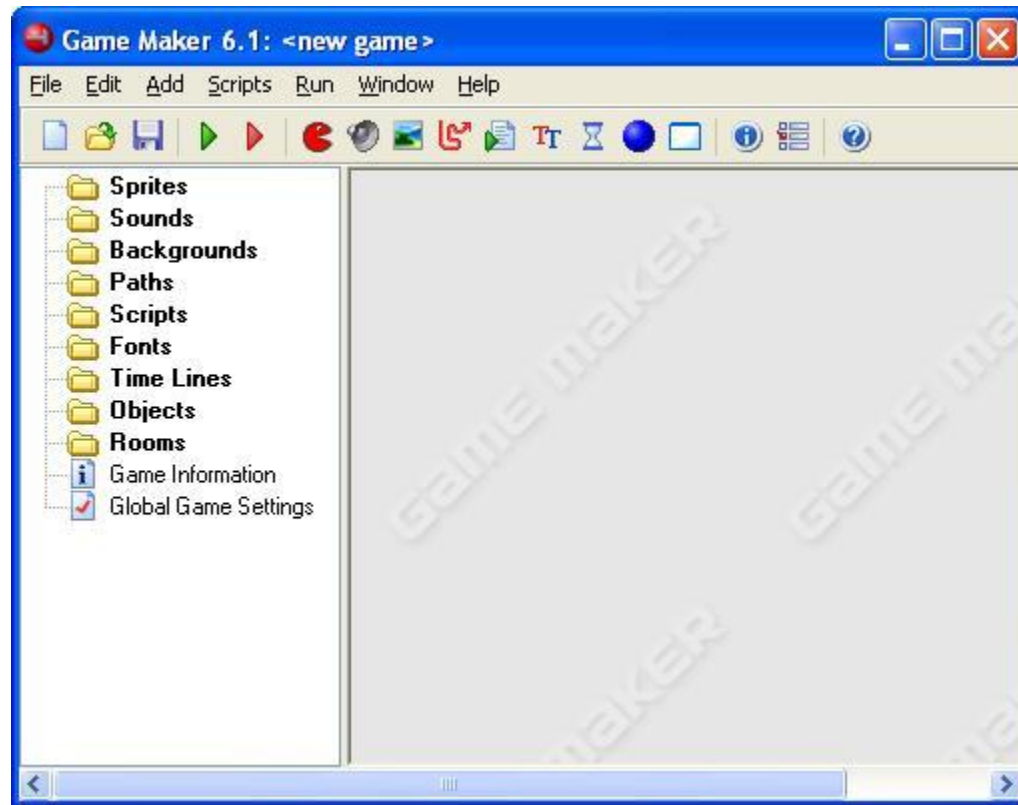
Sons

# Componentes de um jogo e seu relacionamento

## Eventos e Ações

Objeto	Evento	Objeto	Ação
	Colisão		 Quicar 
	Colisão		 Quicar   Destruir 
	Sair da <b>sala</b> por cima ou pelos lados	-	 Quicar 
	Sair da <b>sala</b> por baixo	-	 -1 vida  Se vida=0, game over  Senão, cria nova 

# Introdução ao GameMaker





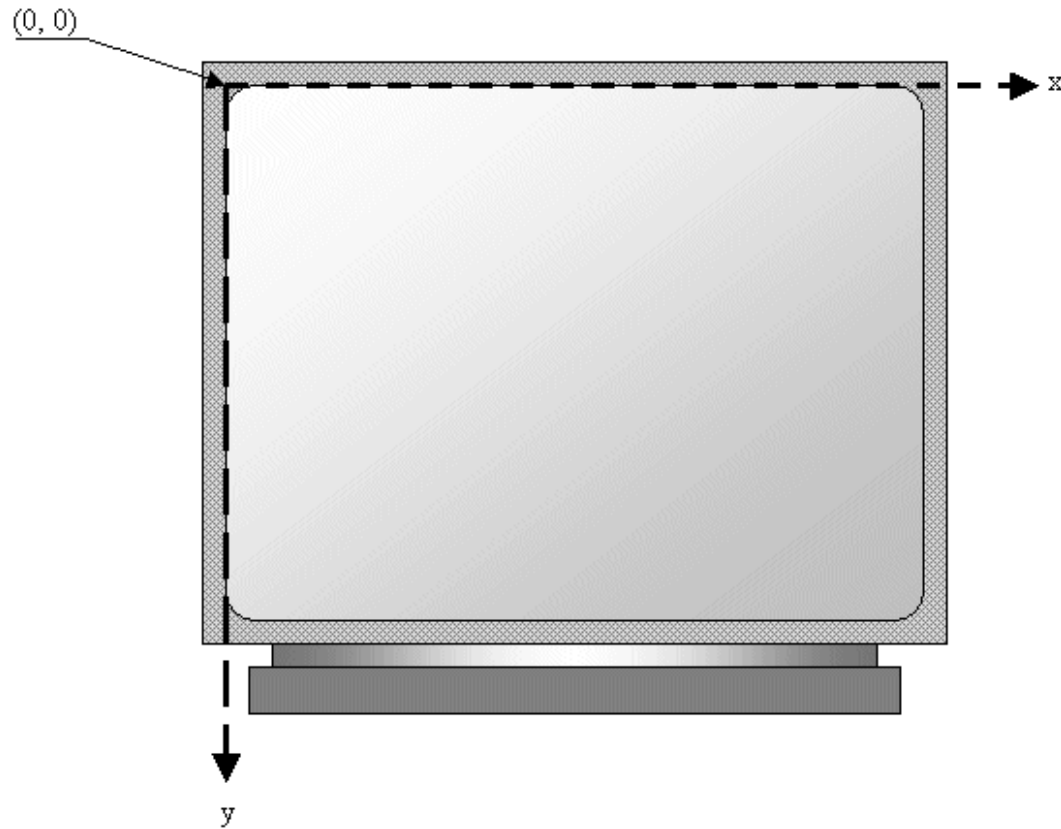
# Conceito

- 🐜 Ao dar **nome aos componentes** de um jogo no Game Maker, lembre-se que:
  - 🐜 Use apenas letras, números e underscore \_
  - 🐜 Evite usar acentos, espaços e pontuação
- 🐜 Não dê o mesmo nome objetos de tipos diferentes (por exemplo, "bola" para a sprite e para o objeto que representa uma bola)
  - 🐜 Sugestão: use prefixos: spr\_bola, obj\_bola
- 🐜 Nomes são case-sensitive. Assim, "bola" é diferente de "Bola".
  - 🐜 Cuidado com isso ao escrever código!



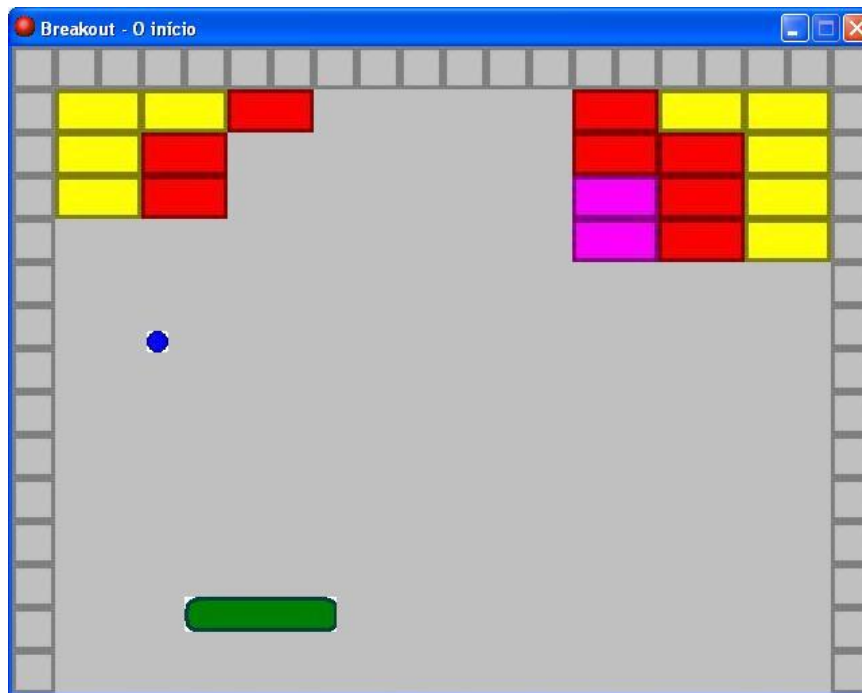
# Conceito

## 🐛 Sistema de Coordenadas



# Exercício: "Simple Breakout"

🐛 Conceitos: Objetos, sprites, eventos, uso do mouse



# Introdução à Criação de Jogos com GameMaker Dia 2

Alexandre S. Lobão  
[contato@alexandrelobao.com](mailto:contato@alexandrelobao.com)

# Agenda – Dia 2

- 🐜 **Como planejar um jogo**
- 🐜 **Eventos no GameMaker**
- 🐜 **Ações no GameMaker**
- 🐜 **Conceitos: Objetos, Instâncias e Variáveis**
- 🐜 **Exercício: River Raid**
  - 🐜 Conceitos: variáveis, uso de texto, background móvel (scrolling), uso do teclado, controle de energia e número de vidas, posição absoluta e relativa de objetos

# Eventos

👾 Cada objeto responde a um conjunto determinado de eventos



👾 Alguns eventos são detalhados em menus drop-down



# Eventos



- 🐜 Este evento acontece quando uma instância de um objeto é criada.
- 🐜 Geralmente é utilizado para colocar a instância em movimento desde o início e/ou configurar certas variáveis da instância.

# Eventos

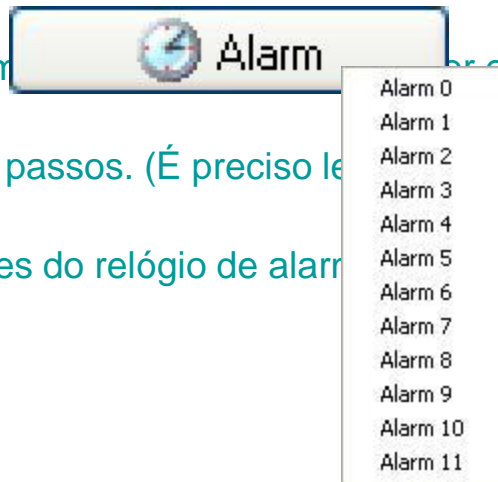


- 🐛 Este evento acontece antes da instância ser destruída, ou seja, a instância ainda existe quando o evento é executado!
- 🐛 Na maioria das vezes, este evento não é utilizado, mas você pode, por exemplo, utilizá-lo para modificar o placar do jogo ou criar um novo objeto.

# Eventos

## Alarm events

- Cada instância possui 8 relógios de alarme. É possível ajustá-los através de certas ações (consulte o
- próximo capítulo). O relógio do alarme então conta em ordem decrescente, até atingir 0, momento em
- que é gerado o **alarm event**. Para indicar ações a um determinado relógio de alarme, primeiramente é
- necessário selecioná-lo no menu. Os relógios de alarme são muito úteis. É possível utilizá-los para fazer
- com que certas coisas aconteçam. Por exemplo, um monstro pode modificar sua
- direção de movimento a cada 20 passos. (É preciso lembrar que, nestes casos, uma das ações no evento
- deve ajustar novamente os valores do relógio de alarme para que ele funcione só uma vez).





# Eventos

## Step events

- Eventos do tipo *step event* acontecem a cada passo do jogo. Utilize-os para colocar ações que precisam
- ser executadas continuamente. Por exemplo, se um objeto deveria seguir outro, é possível ajustar a
- direção do movimento de acordo com o objeto a ser seguido. Entretanto, é preciso se ter cuidado com
- este tipo de evento. Principalmente, não se deve colocar muitas ações complexas para este evento em
- objetos com muitas instâncias. Faz sentido para objetos com poucas instâncias. Para ser mais exato, há 3 tipos de
- step events*. Geralmente, apenas será preciso utilizar o que é considerado padrão, embora seja possível
- através do menu selecionar o **begin step event** e o **end step event**. O primeiro é executado no começo
- de cada passo, antes que qualquer outro evento aconteça. O **normal step event** (padrão) é executado
- antes que as instâncias sejam colocadas em suas novas posições. O *end step event* é executado no fim do
- passo, antes do redesenho da tela. Este último é tipicamente utilizado, por exemplo, para mudar a *sprite*
- dependendo de sua direção atual.



# Eventos

## Collision events

Sempre que duas instâncias colidem (isto é, as sprites se sobrepõem) é gerado um *collision event*. Na verdade, ocorrem 2, um para cada uma das instâncias. A instância pode reagir a este evento. Para este fim, selecione do menu o objeto para qual se quer definir o *collision event*. Após, defina as ações.

Há uma diferença do que acontece quando uma instância colide com um *objeto sólido* do que um *objeto não-sólido*. Antes de tudo, quando não há ações no *collision event*, nada acontece; a instância atual simplesmente se mantém em movimento, mesmo se o outro objeto for sólido. Já quando o evento contém alguma ação ocorre o seguinte:

Quando o outro objeto é **sólido**, a instância é colocada de volta ao seu lugar anterior (antes de ocorrer a colisão). Então o evento é executado. Por exemplo, se o evento reverte a direção, a instância muda para sua nova posição. Assim, continua em movimento, sem parar. Havendo um objeto não-sólido, a instância simplesmente, o

Quando o outro objeto é **não-sólido**, a instância simplesmente, o evento é executado com a instância em sua posição atual. Também, não há uma segunda verificação por colisão. Bem, a lógica nos diz que isto realmente é o que deveria acontecer. Por o objeto não ser sólido, é possível atravessá-lo e o evento notifica quando isto está acontecendo.

Há muitos usos para os *collision event*. Instâncias podem ricochetear em paredes; um objeto que seja atingido por um projétil pode ser destruído; etc.

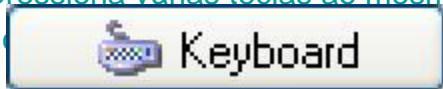


# Eventos

## Keyboard events

Quando o jogador pressiona uma tecla, acontece um *keyboard event* para todas as instâncias de todos os objetos. Há um evento diferente para cada tecla. Através do menu pode-se definir a tecla para o keyboard event e depois arrastar com o mouse as ações. Claramente, são poucos os objetos que precisam

associação com este evento, bem como são poucas as teclas utilizadas nesta associação. É possível obter um evento a cada passo, quando o jogador marca especiais. Um deles é chamado <No key>. Este é pressionado quando nenhuma tecla é pressionada. O outro chama-se <Any key> e acontece quando o jogador pressiona várias teclas ao mesmo tempo. Observe



lois keyboard events so, quando nenhuma tecla é pressionada. Quando o las as teclas pressionadas. ntes se o indicador

## Key press events

Este evento ocorre quando uma tecla é pressionada, mas não é um evento de teclado. Útil exatamente para quando se quer que uma ação aconteça apenas uma vez.



a tecla é pressionada, ao ação aconteça só 1 vez.

## Key release events

Este evento ocorre quando uma tecla é liberada, mas não é um evento de teclado. Útil exatamente para quando se quer que uma ação aconteça apenas uma vez.



cla é liberada, ao invés de

# Eventos

## Mouse events

Um *mouse event* acontece para uma instância sempre dentro da sprite que

a representa. Dependendo de qual botão do mouse es eventos: **no button**,

**left button**, **right button**, ou **middle button**. Os event passo,

contanto que o jogador mantenha o botão do mouse p gerados apenas

uma vez, quando o botão é press botão é

liberado. Observe que estes eventos só acontecem qu sobre a instância

(sprite). Desejando executar ações associadas simples liberar o algum

botão do mouse, use **global mouse events** que poder Há dois eventos

para mouse especiais: **mouse enter** e **mouse leave**. ( do mouse entra

na área da instância. O segundo acontece quando o c eventos são

usados tipicamente para mudar a imagem (sprite) ou t eventos

relacionados ao joystick. Pode-se indicar ações para as 4 direções principais do joystick (na direção

diagonal ocorrem ambos eventos). Da mesma forma, pode-se definir ações para os 8

Left button	
Right button	
Middle button	
No button	
Left pressed	
Right pressed	
Middle pressed	
Left released	
Right released	
Middle released	
Mouse enter	
Mouse leave	
Mouse wheel up	
Mouse wheel down	
Global mouse	▶
Joystick 1	▶
Joystick 2	▶

Left
Right
Up
Down
Button 1
Button 2
Button 3
Button 4
Button 5
Button 6
Button 7
Button 8

stiver

se os

dos a cada

nts são

s quando o

stiver

sionar ou

submenu.








do o cursor

ncia. Estes

te, há os

# Eventos

## Drawing event

-  As instâncias, quando visíveis, têm sua sprite desenhada na tela a cada passo. Ao se especificar ações
-  para o *drawing event*, a sprite não é desenhada, mas as ações é que são executadas. Pode ser usado para
-  desenhar algo diferente da sprite, ou fazer modificações aos parâmetros da sprite. Há diversas ações que
-  podem ser utilizadas em drawing events. Observe que o drawing event só é executado se o objeto for
-  visível. Também observe que, incluindo o  desenhado, os collision events são baseados na
-  sprite que é associada com a instância.

# Eventos

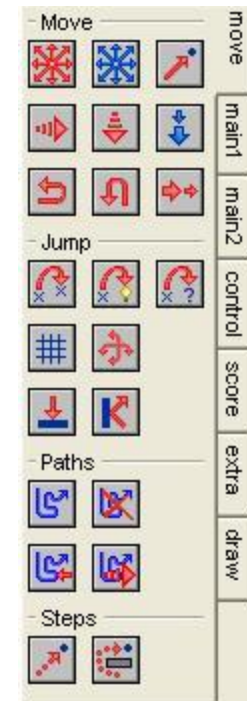
- ✚ Em algumas situações, é importante compreender a ordem em que o *Game Maker* processa os eventos.
- ✚ Esta ordem está descrita abaixo:
- ✚ Begin step events
- ✚ Alarm events
- ✚ Keyboard, Key press, e Key release events
- ✚ Mouse events
- ✚ Normal step events  
(agora todas as instâncias são ajustadas para suas novas posições)
- ✚ Collision events
- ✚ End step events
- ✚ Drawing events
- ✚ A criação, destruição, e outros eventos são realizados quando os fatos correspondentes acontecem.

# Ações

As ações descrevem o que será realizado em resposta aos eventos.

Há diversas abas de ações:

- Mover (Move)
- Principal 1 e 2 (Main)
- Controle (Control)
- Pontuação (Score)
- Ações adicionais (Extra)
- Desenho (Draw)



# Ações de Movimento

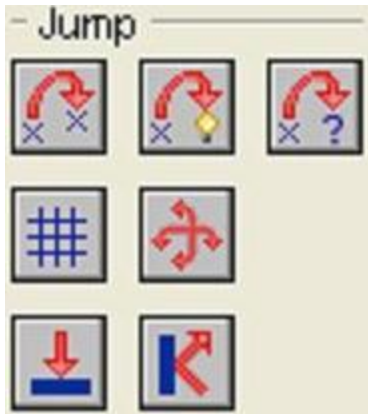


## MOVER:

- 🐜 **Movimentar em uma direção** (com setas)
  - 🐜 **Movimentar em uma direção** (com ângulo)
  - 🐜 **Mover-se em direção a um ponto**
  
  - 🐜 **Ajustar a velocidade horizontal** (valores positivos para a direita, negativos para a esquerda).
  - 🐜 **Ajustar a velocidade vertical**
  - 🐜 **Ajustar a gravidade** (ângulo e velocidade)
  
  - 🐜 **Reverter a direção horizontal**
  - 🐜 **Reverter a direção vertical**
  - 🐜 **Ajustar o atrito**
- 🐜 Dica: Use relative para aumentar ou diminuir os valores com base nos atuais



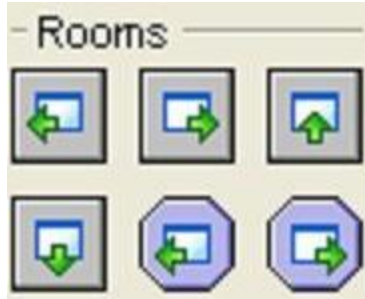
# Ações de Movimento



## SALTAR:

- 🐸 Saltar a uma determinada posição
- 🐸 Saltar à posição inicial
- 🐸 Saltar a uma posição aleatória
- 🐸 Alinhar à a uma grade
- 🐸 “Wrap” (retornar pelo outro lado) ao sair da tela
- 🐸 Mover a uma posição de contato
- 🐸 Ricochetear ao bater em objetos

# Ações Principais (1)



## **SALAS (ROOMS):**

**Ir ao room anterior**

**Ir ao próximo room**

**Reiniciar o room atual**

**Ir a um room diferente**

**Testar se há o room anterior**

**Testar se há o próximo room**



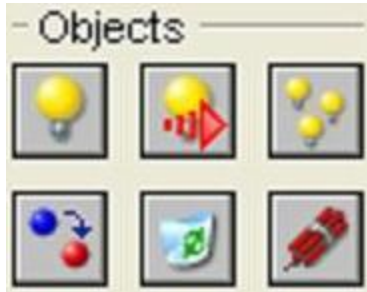
## **SOMS:**

**Tocar um Som**

**Interromper um Som**

**Testar se um som está sendo tocado**

# Ações Principais (1)



## OBJETOS:

- ❖ Criar uma instância de um objeto
- ❖ Criar uma instância de um objeto com velocidade e direção
- ❖ Criar uma instância de um objeto aleatório
- ❖ Modificar a instância
- ❖ Destruir a instância
- ❖ Destruir instâncias em uma posição



## SPRITES:

**Modificar a sprite**

**Transformar a sprite** (*versão registrada*)

**Definir ajuste de cor para a sprite** (*versão registrada*)

# Ações Principais (2)



## TEMPO:

- ⌘ Configurar um alarme
- ⌘ Interromper durante um tempo



## INFORMAÇÃO:

- ⌘ Exibir uma mensagem
- ⌘ Exibir informações sobre o jogo



## JOGO:

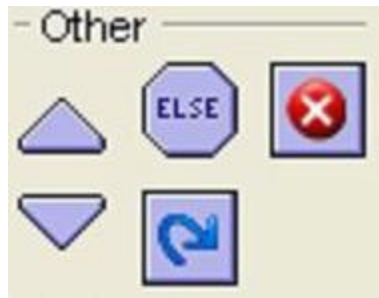
- ⌘ Reiniciar o jogo
- ⌘ Finalizar o jogo
- ⌘ Salvar o jogo
- ⌘ Carregar um jogo

# Ações de Controle



## QUESTÕES:

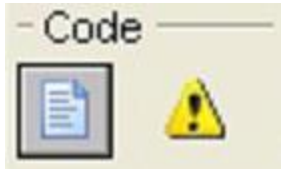
- Se uma posição está livre de colisão
- Se há colisão em uma posição
- Se há objeto em uma posição
- Se o número de instâncias é um certo valor
- Sorteado um número, realize uma ação
- Se o usuário responde “sim” a uma pergunta
- Se uma expressão é verdadeira
- Se está pressionado um botão do mouse
- Se instância está alinhada com a grade



## OUTROS:

- Início de bloco
- Senão
- Sair do evento atual
- Fim de bloco
- Repetir uma ação

# Ações de Controle



## CÓDIGO:

- 🐞 Executar um trecho de código
- 🐞 Comentários



## VARIÁVEIS:

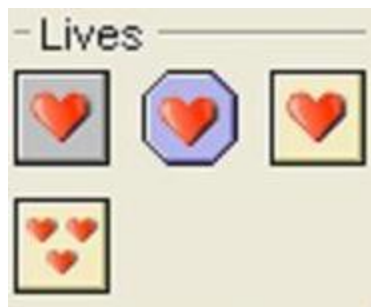
- 🐞 Ajustar o valor de uma variável
- 🐞 Se o valor da variável é igual a
- 🐞 Desenhar o valor da variável

# Ações de Pontuação



## PONTUAÇÃO:

- 🐾 Ajustar o escore
- 🐾 Se o escore é igual a
- 🐾 Desenhar o valor do escore
- 🐾 Exibir os melhores escores
- 🐾 Limpar os melhores escores



## VIDAS:

- 🐾 Ajustar o número de vidas
- 🐾 Se número de vidas é igual a
- 🐾 Desenhar o número de vidas
- 🐾 Desenhar o número de vidas com imagem

# Ações de Pontuação



## ENERGIA:

- 🐜 Ajustar a energia
- 🐜 Se energia é igual a
- 🐜 Desenhar a barra de energia
- 🐜 Atualizar no título da janela



# Ações de Desenho



## DESENHO:

- Desenhar uma imagem sprite
- Desenhar uma imagem background
- Desenhar um texto
- Desenhar um texto modificado (*versão registrada*)
- Desenhar um retângulo
- Desenhar retângulo-gradiente horizontal (*versão registrada*)
- Desenhar retângulo-gradiente vertical (*versão registrada*)
- Desenhar uma elipse
- Desenhar elipse-gradiente (*versão registrada*)
- Desenhar uma linha
- Desenhar uma seta

# Ações de Desenho



## CONFIGURAÇÕES:

- 🐜 Ajustar as cores
- 🐜 Alternar entre modos de exibição



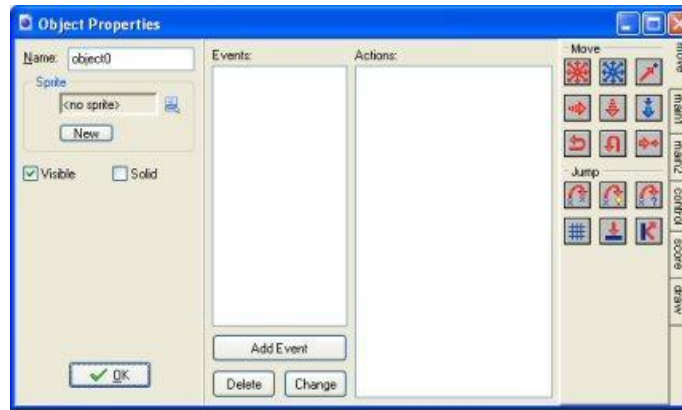
## OUTRAS:

- 🐜 Gravar um instantâneo do jogo (*versão registrada*)

# Conceitos

## Objetos

- É a *definição* ou *modelo* do elemento que será criado durante o jogo (sprite a ser usada, eventos, ações, etc)







## Instâncias

- É cada uso desta definição para criar um elemento *durante a execução do jogo*
- Podemos criar diversas instâncias de um mesmo objeto
  - Por exemplo, diversos “tijolos amarelos” baseados no objeto “TijoloAmarelo”, no jogo Breakout

# Conceitos






## Variáveis

-  São “locais de memória que armazenam informação”. Podem ser:
  -  Pré-definidas pelo GameMaker
    -  Gerais (visíveis em todo jogo):
      - **score** - o valor atual do escore
      - **lives** - o número de vidas atual
      - **health** - a energia atual (0-100)
      - **mouse\_x**, **mouse\_y** - a posição x do mouse
      - **room\_height**, **room\_width** – o tamanho da sala atual
      - Etc
    -  Locais (visíveis por um objeto)
      - **x** - a coordenada x da instância
      - **y** - a coordenada y da instância
      - **hspeed** - a velocidade horizontal (em pixels por passo)
      - **vspeed** - a velocidade vertical (em pixels por passo)
      - **Direction** - a direção atual do movimento, em graus (0-360)
      - **speed** - a velocidade atual na direção (direction) corrente
      - **image\_speed** - a velocidade com que as sub-imagens são exibidas

# Conceitos

## Variáveis

### Definidas pelo usuário

-  Qualquer nome que você definir
-  Lembre-se: “Energia” é diferente de “energia”
-  São definidas com `var <nome da variável>`, ou no primeiro uso
-  Globais (visíveis em todo jogo):
  - Tem o prefixo **global**.
  - Exemplo: `global.Ouro`, `global.Chaves`
-  Locais (visíveis por um objeto)
  - Sem prefixo, qualquer nome não reservado pelo Game Maker
  - Exemplo: “score” é variáveis pré-definida pelo Game Maker, “pontos” é variável do jogador

# Exercício: “River Raid”

Conceitos: variáveis, uso de texto, background móvel (scrolling), uso do teclado, controle de energia e número de vidas, posição absoluta e relativa de objetos



# Introdução à Criação de Jogos com GameMaker Dia 3

Alexandre S. Lobão  
[contato@alexandrelobao.com](mailto:contato@alexandrelobao.com)

# Agenda – Dia 3

- 🐜 **Conceitos Avançados no GameMaker**
- 🐜 **Gêneros de Jogos**
- 🐜 **Como planejar um jogo**
- 🐜 **Palavras dos Gurus:** “Como ser um desenvolvedor de jogos”, tradução da palestra de David Weller, gerente de programas das comunidades de desenvolvimento de jogos da Microsoft Game Technology Group
- 🐜 **Definição do projeto final** (individual por aluno)
- 🐜 **Exercício: Dungeon Dude**
  - 🐜 Conceitos: Uso avançado de sprites e variáveis, informação sobre o jogo (game info), criação de inimigos “inteligentes” (escolha de caminho), controle de objetos carregados, mudança de fases



# Modo Avançado

## Ações de Movimento



### **PATHS:**

- 🐜 Definir o Path (caminho) para uma instância
- 🐜 Remover o Path de uma instância
- 🐜 Mover a instância a um ponto do Path
- 🐜 Configurar a velocidade do path



### **STEPS:**

**Realizar um passo em direção a um ponto**

**Realizar um passo em direção a um ponto, desviando dos obstáculos**

# Modo Avançado

## Ações Principais (1)



### TEMPO:

- Definir uma linha de tempo
- Definir a posição de uma linha de tempo



### INFO:

- Mostrar um vídeo



### RECURSOS:

- Ler uma sprite de um arquivo (*versão registrada*)
- Ler um som de um arquivo (*versão registrada*)
- Ler um background de um arquivo (*versão registrada*)

# Modo Avançado

## Ações de Controle



### CÓDIGO:

 Executar um script



### OUTRAS:

 Chamar o evento herdado (do objeto-pai)

# Modo Avançado

## Ações Adicionais



- ❧ Criar um sistema de partículas
- ❧ Destruir um sistema de partículas
- ❧ Remover todas as partículas do sistema
  
- ❧ Criar um tipo de partícula
- ❧ Definir a cor de uma partícula
- ❧ Definir o tempo de vida de uma partícula
  
- ❧ Definir o movimento de uma partícula
- ❧ Definir a gravidade de uma partícula
- ❧ Criar partículas secundárias
  
- ❧ Criar um emissor de partículas
- ❧ Destruir um emissor de partículas
- ❧ Gerar conjunto de partículas a partir de um emissor
  
- ❧ Gerar partículas seqüencialmente a partir de um emissor

# Modo Avançado

## Ações Adicionais



CD (*versão registrada*):

- 🐜 Tocar trilha do CD
- 🐜 Terminar de tocar o CD
- 🐜 Pausa no CD
- 🐜 Reiniciar do ponto anterior (após pausa)
- 🐜 Testar se há um CD no drive
- 🐜 Testar se há um CD tocando



OUTRAS:

- 🐜 Configurar o ponteiro do mouse
- 🐜 Abrir uma página no *browser* padrão da máquina

# Ações de Desenho



## CONFIGURAÇÕES:

- 🐜 Configurar a fonte para desenhar texto

# Gêneros

- 🐉 Estratégia: planejamento, comando.
- 🐉 Simuladores: realismo, ambiente virtual.
- 🐉 Adventure: raciocínio, solução de enigmas.
- 🐉 Esporte: modalidades esportivas.
- 🐉 Infantil: apelo visual, em geral educativos.
- 🐉 Educativos: em geral infantis.

# Gêneros

- 🐜 Tabuleiro: raciocínio, pouca ação.
- 🐜 Cartas: baralho, pouco apelo visual.
- 🐜 “Shoot'em up”: matar e correr.
- 🐜 Arcade/Ação: mais perícia e habilidade.
- 🐜 RPG: personagens e mundos complexos.
- 🐜 Puzzle: quebra-cabeça, desafio.



# Como planejar um jogo

























Nome:	Tema:
-------	-------

## Plano de Jogo

### Resumo

1. FASE 1 - Análise e Projeto .....	2
1.1. Definir o objetivo do jogo .....	2
1.2. Definir as metas do jogo .....	2
1.3. Descrever os objetos do jogo .....	3
1.4. Descrever os eventos do jogo .....	4
1.5. Descrever as fases do jogo .....	5
2. FASE 2 - Implementação .....	6
2.1. Importar imagens do jogo ("Sprites") .....	6
2.2. Importar os sons do jogo .....	6
2.3. Importar imagens de fundo do jogo ("Backgrounds") .....	7
2.4. Criar objetos e atribuir eventos .....	8
2.5. Criar salas e colocar objetos nas salas .....	10
3. Fase 3 - Testes .....	11
4. Fase 4 - Evolução ! .....	12

.....

Desenvolvido no ambiente online por Amanda                        

Página 1

# Como se Tornar um Desenvolvedor de Jogos

David Weller  
Technical Evangelist  
.NET Developer Division  
Microsoft

# Para um Jogo de Sucesso...

## ❧ Desafie o jogador com:

- ❧ Objetivos claros e possíveis de atingir, oferecendo feedback sobre o progresso do jogador
- ❧ Objetivos de curto e longo prazo
- ❧ Níveis de dificuldade e objetivos secundários, para atender a jogadores com diferentes habilidades

## ❧ Recompense o Jogador

- ❧ Por atingir objetivos principais e secundários
- ❧ Randomicamente

# Para um Jogo de Sucesso...

- ❧ Faça o jogador se sentir no controle
  - ❧ Dando a ele escolhas que pareçam fazer diferença real para o correr do jogo
  - ❧ Não o confundindo com controles demais
  - ❧ Não o punindo por coisas além do seu controle
  - ❧ Dando ao jogador feedback visual e auditivo sobre suas interações com o jogo
- ❧ E Lembre-se:
  - ❧
  - ❧ O jogo precisa ser divertido, não frustrante!

# Exercício: “Dungeon Dude”

Conceitos: Uso avançado de sprites e variáveis, mudança de fases, informação sobre o jogo (game info), criação de inimigos “inteligentes” (escolha de caminho)



# Introdução à Criação de Jogos com GameMaker Dia 4

Alexandre S. Lobão  
[contato@alexandrelobao.com](mailto:contato@alexandrelobao.com)

# Agenda - Dia 4

- 🐜 **Visão Geral da versão registrada**
  - 🐜 Programa RegDemo
- 🐜 **Ferramentas para desenvolvimento de jogos**
- 🐜 **Palavras dos Gurus:** “Um guia para criar jogos de sucesso”, tradução da palestra de Bruce Shelley, do Ensemble Studios (criadores de sucessos como a série “Age of Empires”)
- 🐜 **Projeto Final**
- 🐜 **Próximos passos**

# Próximos Passos – Cursos acadêmicos

- ❖ Curso de Programação de Jogos com XNA na Unieuro, em Brasília – **Em breve!**
- ❖ “Projeto e Implementação de Jogos”, curso da UFPE
  - ❖ <http://www.cin.ufpe.br/~game>
- ❖ “ICAD-Igames – Laboratório de CAD e Jogos Inteligentes”, da PUC-Rio
  - ❖ <http://www.icad.puc-rio.br>
- ❖ “Introdução à Programação de Jogos”, na UFRJ
  - ❖ <http://www.labic.nce.ufrj.br/jogos>
- ❖ “Desenvolvimento de software para entretenimento digital”, da Universidade Estácio de Sá, RJ
  - ❖ [http://www.estacio.br/politecnico/cursos/des\\_soft\\_entr\\_digital.asp](http://www.estacio.br/politecnico/cursos/des_soft_entr_digital.asp)
- ❖ “Arte e design para entretenimento digital”, da Universidade Estácio de Sá, RJ
  - ❖ [http://www.estacio.br/politecnico/cursos/arte\\_design\\_entr\\_digital.asp](http://www.estacio.br/politecnico/cursos/arte_design_entr_digital.asp)
- ❖ “Curso Superior em Tecnologia de Games”, da PUC-SP
  - ❖ [http://www.pucsp.br/paginas/ensino\\_pesquisa/cursos/design\\_de\\_games/index.htm](http://www.pucsp.br/paginas/ensino_pesquisa/cursos/design_de_games/index.htm)
- ❖ “Curso de Desenvolvimento de Jogos Eletrônicos”, da Faculdade Infórium, MG
  - ❖ [www.inforium.com.br](http://www.inforium.com.br)
- ❖ “Curso Superior de Tecnologia em Programação de Jogos Digitais”, da PUC Minas (Unidade São Gabriel)
  - ❖ [www.inf.pucminas.br/index.php?secao=noticias&id=1099](http://www.inf.pucminas.br/index.php?secao=noticias&id=1099)



# Referências

- ✚ Made in Brasil Games – Portal sobre jogos nacionais
  - ✚ <http://www.madeinbrasilgames.com.br/php/>
- ✚ Gametrack - portal de jogos banda larga
  - ✚ <http://www.gametrack.com.br>
- ✚ Programadores e Desenvolvedores de Jogos
  - ✚ <http://www.pdj.com.br>
- ✚ Gaming Brasil – Dicas sobre criação de jogos em diversas plataformas, downloads diversos
  - ✚ <http://gamingbrasil.mundoperdido.com.br>
- ✚ Comunidade Brasileira de Desenvolvimento e Programação de jogos
  - ✚ <http://www.gamedev.com.br>
- ✚ Max Dimension
  - ✚ <http://maxdimension.pdj.com.br>
- ✚ União de Gamers do Brasil
  - ✚ <http://www.ungbrasil.org>
- ✚ Cursos online sobre programação de jogos 2D e 3D gratuitos (em inglês)
  - ✚ <http://www.microsoft.com/events/series/msdnvideodev.mspix>
- ✚ Cursos e Tutoriais sobre OpenGL e outras ferramentas de desenvolvimento de jogos (em inglês)
  - ✚ <http://nehe.gamedev.net>
- ✚ Tutoriais diversos (em inglês)
  - ✚ <http://www.gametutorials.com>
- ✚ Jogos Daqui
  - ✚ <http://www.jogosdaqui.com.br>
  - ✚ <http://www.unidev.com.br/>

# Empresas de Jogos Brasileiras

🐉 Associação Brasileira das Desenvolvedoras de Jogos Eletrônicos

🐉 <http://www.abragames.org>

🐉 Locz games

🐉 <http://www.locz.com.br/loczgames>

🐉 Jynx Playware

🐉 <http://www.jynx.com.br>

🐉 Hoplon Infotainment

🐉 <http://www.hoplon.com.br>

🐉 Délirus Entertainment

🐉 <http://www.delirus.com.br>

🐉 Devworks Game Technology

🐉 <http://www.devworks.com.br>

🐉 Canoasoft.com

🐉 <http://www.canoasoft.com>

🐉 Três D1

🐉 <http://www.tresd1.com.br>

🐉 Ignis Games – Criadora do Jogo Erinia

🐉 <http://www.ignisgames.com.br/erinia>

🐉 Diversos outros em:

🐉 <http://www.abragames.org/membros.htm>


# Eventos

- 🐜 Concurso anual de Jogos Eletrônicos do MEC
  - 🐜 <http://www.jogosbr.org.br>
- 🐜 Concurso Anual de jogos da PDJ
  - 🐜 <http://www.pdj.com.br>
- 🐜 Imagine Cup – Concurso anual, inclui categoria de jogos
  - 🐜 <http://www.imaginecup.com.br/pt-br.aspx/home.aspx>
- 🐜 Conferência de Desenvolvedores de Games
  - 🐜 <http://www.cdgr.io.com.br>

# Ferramentas para desenvolvimento de jogos


## Klik & Play , The Games Factory, Multimedia Fusion, Jamagic

-  <http://www.clickteam.com>

-  No fácil e amigável sistema do tipo clique e arraste, a Clickteam apresenta uma linha de **produtos** para criação de games, dos mais amadores aos com qualidade comercial, sem que a pessoa precise escrever uma só linha de programação. Todos vem com cenários, personagens e efeitos sonoros pré-fabricados para você criar suas aventuras. Interfaces simples, diversos tutoriais e fóruns para trocar idéias com outros criadores. Em Inglês.

## Game Maker


-  <http://www.gamemaker.com.br>,  
<http://www.gamemaker.nl>

-  Também no estilo clique e arraste, o Game Maker possui extensa **comunidade de criadores** pelo mundo todo, oferece suporte on-line na criação do seu game e o melhor, o download do programa completo é grátis. Em Inglês.

# Ferramentas para desenvolvimento de jogos


## 3dgamestudio

-  <http://www.3dgamestudio.com>

-  Faça **games em 3D** com o Gamestudio. Cenários ao ar livre, salas futuristas ou medievais. Homens, mulheres e robôs com poder de andar, nadar e atirar. Você mesmo aprende a manusear tudo isso sem esforço com os objetos e scripts pré-fabricados. Se quiser ir mais longe, daí sim, tutoriais sobre a linguagem de programação C-Script estarão à sua espera. Em Inglês.

## Dark Basic

-  <http://www.darkbasicpro.com.br>

-  Se você sonha em criar games em 3D e 2D com **qualidade comercial** e com suporte em português, visite o site do Dark Basic. Para dominar o programa, você terá que aprender uma linguagem de programação específica baseada em Basic.

## Blitz Basic


-  <http://www.blitzbasic.com>

-  Muito semelhante ao Dark Basic, com **linguagem derivada do Basic**. Permite exportar e importar cenários, personagens e criar um game original e autoral em 3D ou em 2D. Em Inglês.

# Ferramentas para desenvolvimento de jogos


## Milk Shape

 <http://www.swissquake.ch/chumbalum-soft>

 Software de fácil aprendizado, ideal para criação de **personagens em 3D**. Você começa do nada e vai acrescentando a pele, o rosto, as vestimentas. O editor de animação embutido permite que você crie os movimentos do seu herói ou heroína e depois exporte para o ambiente desejado. Em Inglês.

## Anim8or

 <http://www.anim8or.com>

 Dê seus primeiros passos na **criação e animação** de personagens e objetos em 3D com o Anim8tor, programa gratuito mas poderoso e fácil de aprender. Download de manual disponível em português. Em Inglês.

# Ferramentas para desenvolvimento de jogos

## **Truespace**

- ✿ <http://www.caligari.com>

- ✿ Programa de **criação e animação** em 3D muito popular no mundo dos games. A versão mais recente, a 6.6, disponível para experimentação, vem com vários esqueletos e rostos pré-fabricados para montagem de personagens. Versão 3.2 gratuita até para fins comerciais. Em Inglês.

## **Terragem**

- ✿ <http://www.planetside.co.uk>

- ✿ Programa especializado na criação de **cenários** - florestas, montanhas, construções. Os detalhes impressionam. Dá para criar reflexos na água, irregularidades no terreno, sombras nas árvores e até estabelecer a posição do sol. Download grátis para fins não comerciais. Em Inglês.

## **Mataleone Counter Strike**

- ✿ <http://www.mataleone.com>

- ✿ Site de Roger Mataleone, brasileiro cujo trabalho na criação de mods - níveis próprios - para o **Half-Life** é reconhecido no exterior. Tutoriais detalhados para quem deseja criar suas próprias fases do jogo Half-Life. A seção downloads disponibiliza o Valve-Hammer 3.4, editor de mapas oficial da Valve, produtora do game.

# Ferramentas para desenvolvimento de jogos

## Softimage

- ✿ <http://www.softimage.com>
- ✿ Star Wars - The Clone Wars, Soldier of Fortune 1 e 2, Half-Life 2, Resident Evil e Battlezone 1 e 2 são alguns top games que utilizaram a tecnologia do Softimage, sofisticado programa da Avid para **animação em 3D**. Quer saber como eles foram feitos? Baixe a versão gratuita para fins não-comerciais e estude o programa Softimage XSI EXP v.3.0. Importante. Seu computador deve ter no mínimo Pentium 4 e 256 MB de memória para rodar o poderoso software. Em Inglês.

## Adobe


- ✿ <http://www.adobe.com.br/print/main.html>
- ✿ Cansado de objetos, cenários e heróis pré-fabricados? Desenhe você mesmo no computador seus castelos, guerreiros, florestas e naves espaciais com os programas de **design e layout** da Adobe como o Illustrator, o Photoshop e o Dimensions. Depois exporte para o programa usado para criar o game.



# Ferramentas para desenvolvimento de jogos


## Macromedia

 <http://www.macromedia.com/br>

 Com recursos de desenho, som e animação, o **Flash MX** também pode ser um valioso aliado na criação de elementos originais para games on-line já que,ção invés de bitmaps, utiliza arte do tipo vetor, mais fácil de baixar e transmitir na rede.

## Fastgraph

 <http://www.fastgraph.com>

 Programa recomendado para quem deseja criar objetos, cenários e personagens para games em 2 e 3D com alta qualidade e **rápida renderização**. Em Inglês.

 Etc, etc, Etc...

# Um Guia para Desenvolver Jogos de Sucesso

Bruce Shelley  
Ensemble Studios

# Exercício: “Projeto Final”

Conceitos: Agora é com vocês!

- ✿ Mudança de fases
- ✿ Vidas, Energia, Bônus
- ✿ Uso avançado de variáveis
- ✿ Uso de decisões randômicas
- ✿ Criação de objetos dinamicamente
- ✿ Uso de código para funções avançadas
  - ✿ Use a imaginação!

# Próximos Passos

♣ Decida o que você quer ser!

♣ Amador

♣ Profissional

- Artista (imagem, som)
- Roteirista
- Programador
- Etc

♣ Pesquise e se aprofunde

♣ Estude. Muito.

♣ Dúvidas? Use os grupos da internet ou nosso grupo:

♣ Página: <http://br.groups.yahoo.com/group/GameMakerBrasilia>

♣ mails: [GameMakerBrasilia@yahoogrupos.com.br](mailto:GameMakerBrasilia@yahoogrupos.com.br)