

Serious Games para Saúde e Treinamento Imersivo

Liliane S. Machado

*Laboratório de Tecnologias para o Ensino Virtual e Estatística
Universidade Federal da Paraíba*

Ronei M. Moraes

*Laboratório de Tecnologias para o Ensino Virtual e Estatística
Universidade Federal da Paraíba*

Fátima L.S. Nunes

*Laboratório de Aplicações de Informática em Saúde
Universidade de São Paulo*

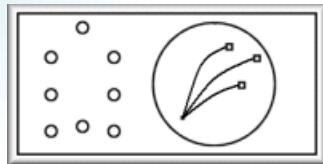


Jogos Eletrônicos

- Entretenimento
- Atividade espontânea
- Regras
 - Pontos
 - Vitória/Derrota
 - Tempo
- Tomada de decisão

Como isso começou?

- Primeiro jogo eletrônico: Míssil
- Desenhado para jogo em CRTs
- Patenteado em 1947 - EUA
- Simula o disparo de um míssil sobre um alvo



O Pinball

- A indústria dos videogames teve suas origens no pinball
- Originalmente, os pinballs eram, de fato, pouco mais que pinos e bolas.
- Baffle Ball, de 1931, foi o primeiro pinball a fazer grande sucesso.



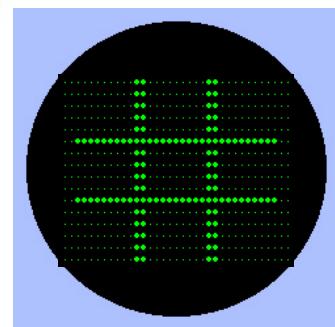
Ralph Baer

- Visionário dos videogames, concebeu a televisão interativa em 1951
- Criador de vários jogos eletrônicos, entre eles o Genius, em 1978



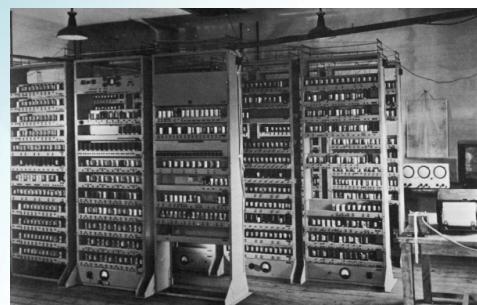
OXO

- Em 1952, A. S. Douglas (Univ. de Cambridge - UK) cria o OXO, como parte de sua tese em Cambridge
- OXO é o primeiro jogo a ter gráficos, em uma matriz de LEDs.



OXO

- OXO rodava no EDSAC, o segundo computador com arquitetura Von Neumann



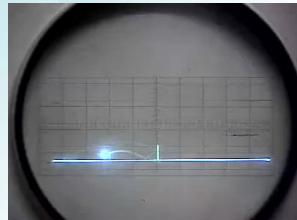
Tennis for Two

- Em 1958, W. Higinbotham criou Tennis for Two, para uma feira de ciências.
- Tennis for Two foi o primeiro jogo a possuir simulação física realista.



Spacewar

- Em 1961, Spacewar foi o primeiro jogo a sair de um laboratório.
- Ele usava tão plenamente os recursos do PDP-1, que era usado como self-test.



Space Travel

- Nessa época, Ken Thompson criou Space Travel.
- Após Space Travel, ele criou a linguagem C e o UNIX.



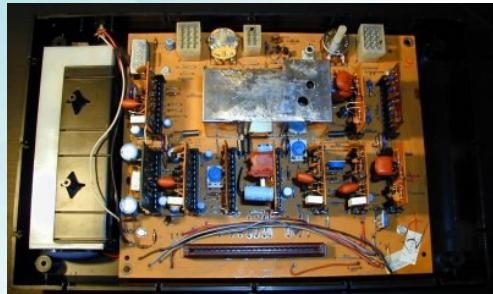
Odyssey

- Entre 1966 e 1969 inventa o video-game e o jogo Pong



Odyssey

- Em 1972, Ralph Baer faz um contrato com a Magnavox que lança o Odyssey.
- O Odyssey foi o primeiro a ter cartuchos.



Odyssey

- O cenário de fundo dos jogos era feito com telas acrílicas sobre a TV.



Nolan Bushnell

- Em 1972, Bushnell criou um arcade baseado em Spacewar.
- Logo após, criou sua própria empresa.
- Um amante de Go, nomeou-a com o termo para xequem Go que, em japonês, é ...



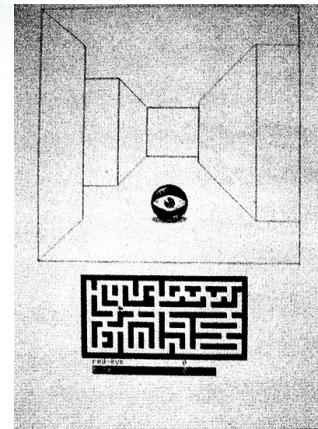
Nolan Bushnell

- Em 1972, Bushnell criou um arcade baseado em Spacewar.
- Logo após, criou sua própria empresa.
- Um amante de Go, nomeou-a com o termo para xequem em Go que, em japonês, é ...



Maze War

- Em 1974, rodando no Xerox Alto, Maze War foi o primeiro multiplayer sobre ethernet, e também o primeiro first person shooter.



Breakout

- Breakout, em 1976, foi criado por dois empregados da Atari: Steve Jobs e Steve Wozniak.



Atari VCS

- Em 1977, chega ao mercado o Atari VCS, o maior sucesso da geração.



Space Invaders

- Em 1978, Space Invaders vira mania nos arcades, causando uma escassez de moedas no Japão.



Intellivision

- O “Intelligent Television”, de 1980, era único por ter um processador de 10 bits.



Pac Man

- A Nakamura Manufacturing Company criou Puck-Man em 1980, mas teve que mudar o nome do jogo fora do Japão.



Game & Watch

- Em 1980, Gunpei Yokoi, da Nintendo, cria o Game & Watch, um dos primeiros videogames portáteis.



Radar Scope

- Radar Scope, de 1980, foi o primeiro arcade da Nintendo nos EUA, e foi um fracasso.
- Para contornar o prejuízo, mandaram do Japão um estagiário que prometeu criar um novo jogo para a placa...



Donkey Kong

- ... Shigeru Miyamoto, que criou Donkey Kong e seu herói, Mario.



Dragon's Lair

- Em 1983, Don Bluth, animador da Disney, cria Dragon's Lair, o primeiro jogo com vídeo em tela cheia.
- Todas as animações eram tocadas a partir de um laserdisc.



I, Robot

- Ainda em 83, a Atari lança o arcade I, Robot; o primeiro com polígonos em 3D e controle de câmera.



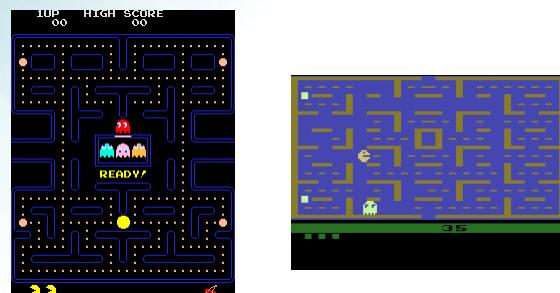
O Crash de 83

- Com o mercado saturado, a proliferação de jogos ruins, a gerência incompetente e o crescimento dos computadores pessoais, o mercado de videogames implodiu em 1983.



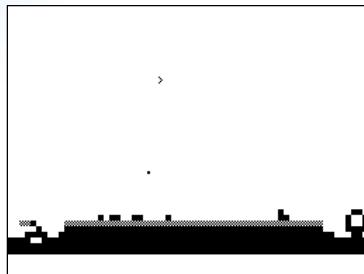
O Crash de 83

- Pac-Man do Atari, além de muito inferior ao arcade, teve mais cartuchos produzidos que a quantidade total de consoles.



Aeroporto 83

- No Brasil, em 1983, Renato Degiovani produz Aeroporto 83.
- Ele foi o primeiro jogo nacional escrito em assembly, em open source para a revista Micro Sistemas.



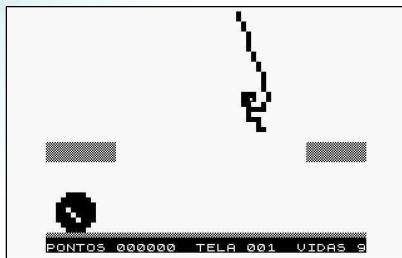
Divino Leitão

- Nas prateleiras, os primeiros jogos nacionais foram criados por Divino Leitão, como Cavernas de Marte e Q-Bert...



Em Busca dos Tesouros

- ...e um destaque espacial vai para Em Busca dos Tesouros, de Tadeu Curinga da Silva, possivelmente o melhor jogo escrito para a plataforma ZX81.



Sega SG-1000

- No Japão, a Service Games (SEGA) lançou o SG-1000, com hardware similar ao Colecovision.



Girl's Garden

- O jogo Girl's Garden foi o primeiro jogo direcionado para garotas.
- Foi criado por Naka Yuuji, para o SG-1000, quando ainda era um estagiário.



Commodore 64

- Sem os videogames como concorrência, essa foi a época de ouro para os jogos de computador.
- Nos EUA, o campeão de vendas era o Commodore 64...



ZX Spectrum

- Na Europa, o ZX Spectrum...



MSX

- E no Japão, o MSX.



Ultimate

- Nessa época, várias softwares se estabeleceram, como a Ultimate...



Konami

- E a Konami, que lançava no MSX várias franquias de sucesso, como Metal Gear



Karateka

- No Apple II, Jordan Mechner trazia a rotoscopia para os videogames, criando Karateka; e, mais tarde, Prince of Persia.



O Retorno dos Videogames

- Em 1983, no Japão, a Nintendo lançava seu videogame, o Family Computer (Famicom). No ano seguinte, já era um sucesso de vendas.



O Retorno dos Videogames

- Nos EUA, o Famicom foi chamado de Nintendo Entertainment System (NES), perdeu a cara de brinquedo, e só conseguiu entrar de fato no mercado em 1985



O Selo de Qualidade

- Para evitar um novo crash, todos os jogos dos NES deviam passar por uma avaliação. Sem o selo de qualidade, jogos não eram lançados.



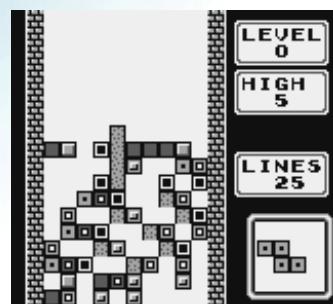
O Game Boy

- Em 1989, Gunpei Yokoi da Nintendo cria o Game Boy.
- O portátil era bem inferior à capacidade do NES, possuindo tela monocromática.
- Para vender o console, ele precisava de um killer app.



Tetris

- A solução foi encontrada além da Cortina de Ferro: Tetris, um jogo russo.



Game Gear

- O Tetris vendeu o Game Boy, assim como Mario havia vendido o NES.
- Concorrentes, como o Game Gear da Sega, não tiveram espaço, mesmo sendo coloridos.



Mega Drive

- A vingança da Sega viria no mesmo ano de 89. Eles lançaram o Mega Drive, de 16 bits, dois anos antes da Nintendo lançar seu sistema de 16 bits.



Jogos de Celebridades

- Inicialmente o Mega Drive não fez sucesso
- Sem a presença de franquias populares, como Mario e Zelda, o Mega Drive buscava reconhecimento fácil licenciando imagens de celebridades.
- Nessa época surgem jogos como Buster Douglas Boxing e Michael Jackson Moonwalker.



Sonic

- Em 1991, Naka Yuuji, criou o Sonic para o Mega Drive.
- Naquela época, pesquisas indicavam que 70% das crianças preferiam o Sonic ao Mario.



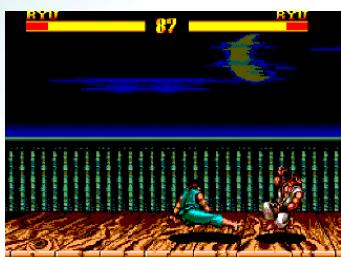
Street Fighter 2

- Enquanto isso, em 1991, os arcades davam sua contribuição final aos jogos: Street Fighter 2, que iniciava uma mania de jogos de luta.
- Curiosamente, o jogador brasileiro era um homem-macaco elétrico.



Street Fighter 2

- Street Fighter 2 foi convertido para todos os consoles da época, mas a melhor versão era a do Super Nintendo.
- A conversão para Master System foi feita no Brasil, pela Tec Toy.



Virtual Boy

- Em 1995, a Nintendo lançou o Virtual Boy, console portátil com estereoscopia real.
- Mas o console foi um fracasso e selou a carreira de seu criador, Gunpei Yokoi.



Real Time Strategy

- Nessa época, o PC já era um computador popular. A interface com mouse permitiu o aparecimento de novos tipos de jogos, como o RTS.
- O jogo que popularizou o estilo foi Dune 2



O Shareware

- O grande número de BBS permitiu a criação de um novo modelo de venda, o shareware.
- Empresas como Apogee e Epic Megagames foram pioneiras no formato.



Wolfenstein 3D

- Nessa época, o shareware de maior influência foi Wolfenstein 3D, feito pela ID Software, o primeiro FPS a usar técnicas de raycasting para simular um ambiente 3D.



Doom

- A engine de Wolfenstein 3D era limitada na direção vertical. Doom, o jogo seguinte da ID, aperfeiçoou a engine, permitindo paredes de altura variável e técnicas avançadas de iluminação.



Doom

- Mais que um jogo, Doom iniciou uma nova cultura.
- Ele popularizou o multiplayer em rede e a customização dos jogos (os próprios jogadores expandiam o jogo, refazendo mapas e gráficos).



Duke Nukem 3D

- Porém, Doom ainda tinha limitações em sua engine. Duke Nukem 3D permitia a criação de cenários móveis e paredes inclinadas.



Descent

- Mas o primeiro jogo a usar realmente uma engine totalmente em 3D, com seis graus de liberdade, foi Descent.



Quake

- Logo em seguida, a ID Software criou sua própria engine 6DoF, e lançou mais uma franquia de sucesso: Quake.



Mega CD

- Enquanto isso, nos consoles, o tamanho limitado das EPROMs já começava a ser um problema.
- A SEGA lançou o Mega CD, permitindo que o Mega Drive tivesse acesso a CD-ROMs.



O CD do Super Famicom

- A Nintendo, não pretendendo ficar pra trás, procurava como parceiros a Sony e a Philips, com experiência na mídia.
- No final, acabou escolhendo a Philips, que já tinha lançado o console CD-i.



O Playstation

- A Sony resolveu dar continuidade ao projeto abandonado pela Nintendo.
- O adaptador de CD para SNES foi então aprimorado e tornou-se o Playstation.



O Playstation

- O Playstation popularizou os aceleradores 3D implementados em hardware.
- 2MB de RAM e 1MB de vídeo
- Mas inúmeros outros fatores contribuíram para o seu sucesso...



O Playstation

- Um desses fatores foi o suporte ao third party, como, por exemplo, com o kit Yaroze de programação.
- Em algumas promoções, você podia pegar o kit de graça, desde que se comprometesse a lançar pelo menos um jogo no mercado.



O Playstation

- Outro fator foi a exclusividade no lançamento de franquias populares, como Final Fantasy.
- Jogo mais vendido: Gran Turismo



O Playstation

- Não menos importante, a facilidade de compra de CDs piratas tornou o sistema extremamente popular, principalmente em países como o Brasil.
- ... E a Sony desbancou a Nintendo



O Nintendo 64

- A Nintendo respondeu em 1996 com o Nintendo 64.
- Usava cartuchos
- Tinha apenas 3 títulos: Super Mario 64 , Wave Race 64 e PilotWings 64.



O Nintendo 64

- Tinha processador gráfico da Silicon Graphics (cores de até 32 bits).
- Avançado processador de áudio => alta qualidade da trilha e efeitos sonoros.



O Sega DreamCast

- 1998 – Acesso à Internet via Modem
- Usava o Windows CE como SO
- Jogo mais vendido: Sonic Adventure
- Último console Sega
- Descontinuado em Março de 2001, devido ao Playstation 2



O Playstation 2

- Lançado em 1999
- Processador Sony/Toshiba Emotion Engine (processa grupos de 32 bits => 128 bits)
- joystick com force-feedback (vibração)



O Playstation 2

- Privilegia a jogabilidade e gráficos
- Jogo mais vendido: Gran Turismo 3



Nintendo Game Cube

- Lançamento em 2001
- Processador PowerPC 750CX e Vídeo ArtX (vendida para ATI-AMD)
- Usava miniDVD
- Ficou em 3º lugar de vendas atrás do PS2 e do Xbox



Nintendo Game Cube

- Adaptadores para modem e banda larga
- Jogo mais vendido: Super Smash Bros. Melee



XBOX

- Primeiro console da Microsoft - 2001
- Processador Intel Pentium 3 733MHz
- 64MB compartilhado
- Foi o primeiro a incorporar HD
- Ethernet 100Mbit e Wireless



XBOX

- Segundo lugar em vendas atrás do Playstation 2
- Descontinuado em 2006
- Jogo mais vendido: Halo 2



Nintendo Game Boy Advance

- Lançamento em 2001
- Uso de cartucho
- Processador ARM7TDMI
- 32 Kb + 96 Kb VRAM (interno na CPU) + 256 Kb WRAM (externo)
- Jogo mais vendido: Pokémon Ruby e Sapphire



Nintendo Game Boy Advance SP

- Lançamento em 2003
- Redesenho do GBA
- Processador ARM7TDMI (32 bits) e co-processor Z80 (8bits)
- Mesma memória do Game Boy Advance
- Jogo mais vendido: Pokémon Ruby e Sapphire



Nintendo DS

- Lançamento em 2004
- Duas telas (inferior é sensível ao toque)
- Conexão Wireless
- Processador ARM946E-S e co-processadores ARM7TDMI
- 4MB de memória RAM



Nintendo DS

- Usa cartões de jogo
- Jogo mais vendido: Nintendogs



O Playstation 2 Slim

- Lançado em 2004
- Menor, mais fino e mais silencioso que o PS2
- Inclui uma porta ethernet
- Novo joystick com force-feedback



Nintendo Game Boy Micro

- Lançamento em 2005
- 2º redesenho do GBA
- Processador ARM7TDMI (32 bits)
- Jogo mais vendido: Pokémon Ruby e Sapphire



Playstation Portable

- Lançamento em 2005
- Gráficos compatíveis com o PS2
- Processador MIPS R4000
- 32MB RAM
- Tela Widescreen com alta definição e luminosidade



Playstation Portable

- Conexão Wireless
- USB 2.0
- Disco UMD (1,8 GB) e Memory Stick Duo
- Jogo mais vendido: Grand Theft Auto: Liberty City Stories



Xbox 360

- Lançamento em 2005
- Processador 3.2 GHz IBM PPC Tri-Core Xenon
- Vídeo ATI Xenos
- HD 20 a 120GB
- 512 MB RAM



Xbox 360

- Jogo mais vendido: Halo 3



Playstation 3

- Lançamento em 2006
- Uma porta HDMI, uma Gigabit ethernet e 2 USBs
- Privilegia desempenho: Processador IBM com 7 núcleos (existe um 8º para redundância)
- Somente 6 são acessíveis (um para SO)



Playstation 3

- Várias capacidades de disco rígido: 20 a 160GB)
- 256 MB RAM e 256 MB GDDR3 de vídeo
- Jogo mais vendido: MotorStorm
- Joystick wireless



Nintendo Wii

- Lançamento em 2006
- Processador IBM PPC
- Vídeo ATI
- Em termos de hardware é o pior console do mercado, mas ...



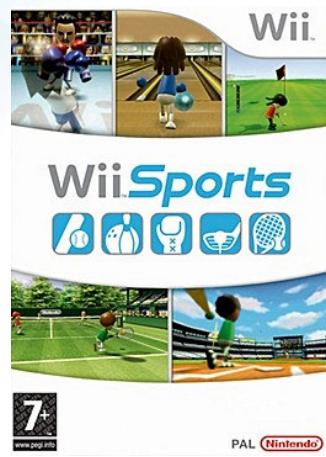
Nintendo Wii

- ... tem a interatividade !



Nintendo Wii

- Controle Wii – primeiro controle remoto para consoles
- Possui acelerômetros e detecção por infravermelho => posição no espaço 3D
- Jogo mais vendido: Wii Sports Europe



Nintendo DS-Lite

- Lançamento em 2006
- Redesenho melhorado do DS
- Processador ARM946E-S e co-processadores ARM7TDMI
- Mesmos processadores, mas usa menos energia
- É menor e mais leve que o Nintendo DS



Playstation Portable Slim and Lite (PSP-2000)

- Lançamento em 2007
- Redesenho do PSP: menor e mais leve
- Melhorias: 64MB memória RAM e navegador
- Jogo mais vendido: Monster Hunter Portable 2nd G



PSP-3000

- Lançamento em outubro/2008
- Atributos: Tela antireflexo, aumento do contraste, microfone integrado
- 4GB Memory Stick Pro Duo



Nintendo DSi

- Lançamento em novembro/2008 (Japão)
- Incluirá duas câmeras e microfone
- Menor, mas com telas maiores
- Processador ARM946E-S e co-processadores ARM7TDMI



Tipos de Jogos

- Objetivo
 - Simuladores
 - Estratégia
 - Infantis
 - RPG
 - Aventura
 - Passatempo
 - Esportivos

Tipos de Jogos Simuladores

- Simulação de situações
 - combate / deslocamento
 - comportamento



Tipos de Jogos Estratégia

- Tomada de Decisão



Tipos de Jogos Infantis

- Desenvolver coordenação motora
- Diferenciação de formas e cores



Tipos de Jogos RPG (role playing game)

- Ação em primeira pessoa
- Defesa/Ataque
- Busca de Elementos



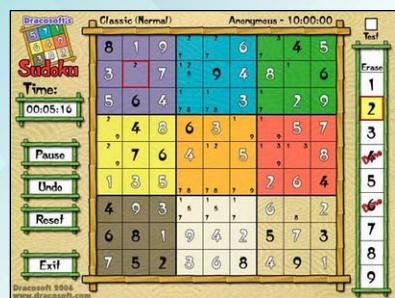
Tipos de Jogos Aventura

- Enredo atrativo
- Missão



Tipos de Jogos Passatempo

- Raciocínio



Tipos de Jogos Esportivos

- Modalidade esportiva



Jogos Eletrônicos

- Entretenimento
- Atividade espontânea
- Regras
 - Pontos
 - Vitória/Derrota
 - Tempo
- Tomada de decisão

“Só” entretenimento ??

Novas abordagens

- Agregar conteúdo específico
 - treinamento
 - educação
 - planejamento
 - conscientização
 - venda e produtos
- Jogar = atividade importante e agradável

Serious Games

Serious Games

- Abordagem de situações práticas
- Propósito específico
- Potencial de aplicação!
- Mercado em expansão = \$\$\$\$\$\$

Advergames

- Venda ou divulgação de produtos
 - chama atenção por mais tempo
 - maior impacto
 - fixa marca

video



1983 – Pepsi Invaders

adverGAME.com™

HOME WHY GAMES? ABOUT SKYWORKS GAMES FOR LICENSING CUSTOM SOLUTIONS CONTACT US

WHY GAMES?

Let's face it. The 30-second television commercial is tired. The banner ad just ain't clicking. And pop-ups are public enemy #1. But Advergames are the branded interactive entertainment experiences that generate buzz and keep consumers coming back for more.

>LEARN MORE

WHY SKYWORKS?

Skyworks pioneered the concept of Advergaming by creating and launching Lifesavers CandyStand.com in 1996. Today our Advergames are played over 40 million times each month.

>LEARN MORE

SELECT CLIENTS

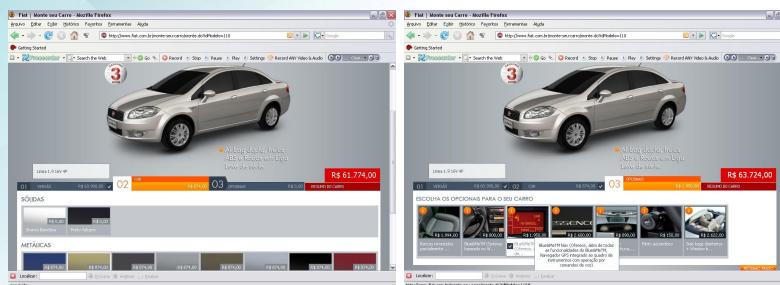
ADVERGAMES FROM SKYWORKS

ESPN arcade **CROSSOVER** **Custom Solutions** **Choose a game from our 100+ game library.** **Games for Licensing**

©2008 SKYWORKS INTERACTIVE • 505 MAIN STREET HACKENSACK NJ 07601 • 201.457.1000 • F:201.457.1099

Personalização

- Montadoras de veículos
 - escolha de opções
 - agilizar vendas
 - simular possibilidades (e preços!)



Conscientização

- Virtual University (www.virtual-u.org)
 - entender o funcionamento de uma universidade
 - tomada de decisões administrativas e acadêmicas
 - ações de melhoria dos indicadores universitários



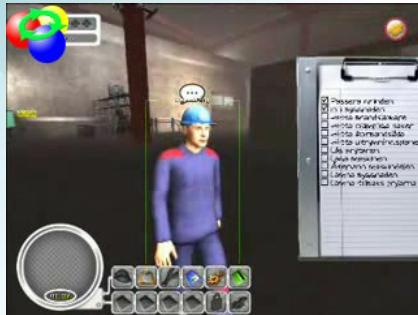
Educação

- Ace's Adventures
 - bons hábitos no trânsito para crianças de 6-9 anos



Educação

- S.A.F.E.
 - segurança no trabalho



video

Treinamento

- Virtual Surgeon Open Heart
- (www.abc.net.au/science/lcs/heart.htm)
 - ensinar tarefas para realizar cirurgia cardíaca



Treinamento

- 3D Driving Academy (www.3d-fahrschule.de)
 - simulação das leis de trânsito
 - IA para controle de tráfego
 - engine de física para veículos



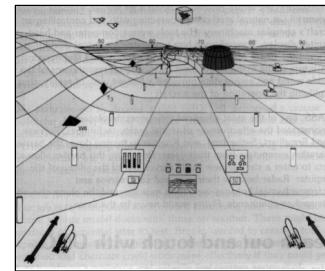
Simulações de Risco

- Triple Zero Hero
 - Departamento de Serviços Emergenciais – Austrália
 - salvamento de pessoas
 - ações em caso de incêndio
 - uso de gamepads



Serious Games e RV

- Anos 80 – simuladores de vôo
 - deslocamento em campo de batalha
 - manejo de aviões
 - desvio de obstáculos
 - exploração
- jogos de guerra ou simulações
 - Pontos
 - Regras
 - Tomada de Decisão
 - Absorção de Estratégias



Treinamento

- Militares (America's Army)
 - deslocamento em campo de batalha
 - manejo de tanques e aviões
 - técnicas de batalha
 - tomada de decisão
 - relações culturais



[video](#)

Educação

- Serious Gordon
- (Dublin Institute of Technology - Irlanda)
 - RPG
 - boas práticas no manejo de alimentos

[video](#)



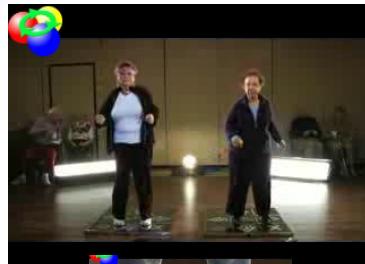
Educação

- GeoespaçoPEC
 - geometria espacial
 - RPG com estereoscopia



Reabilitação

- Uso de dispositivos especiais
 - atividade física (exergames)
 - reabilitação de movimentos



[video](#)



Tratamento de Fobias

- Uso da realidade aumentada
 - altura
 - insetos
 - público



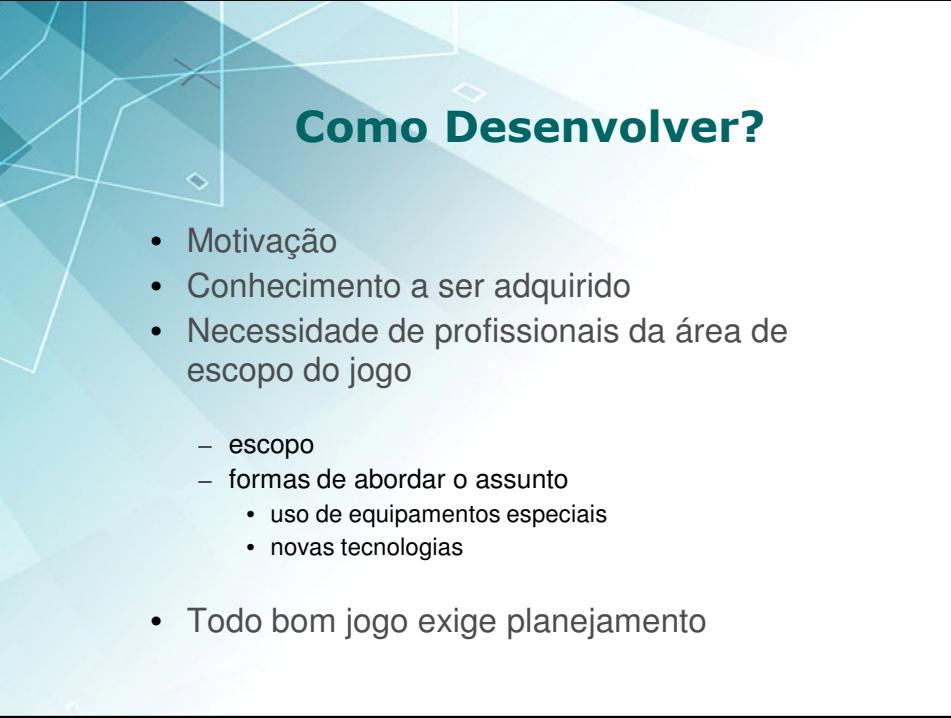
Recursos necessários para desenvolvimento

- Computação
- Física
- Matemática
- Artes (design)
- Estatística
- Pedagogia
- Conhecimentos específicos do tema



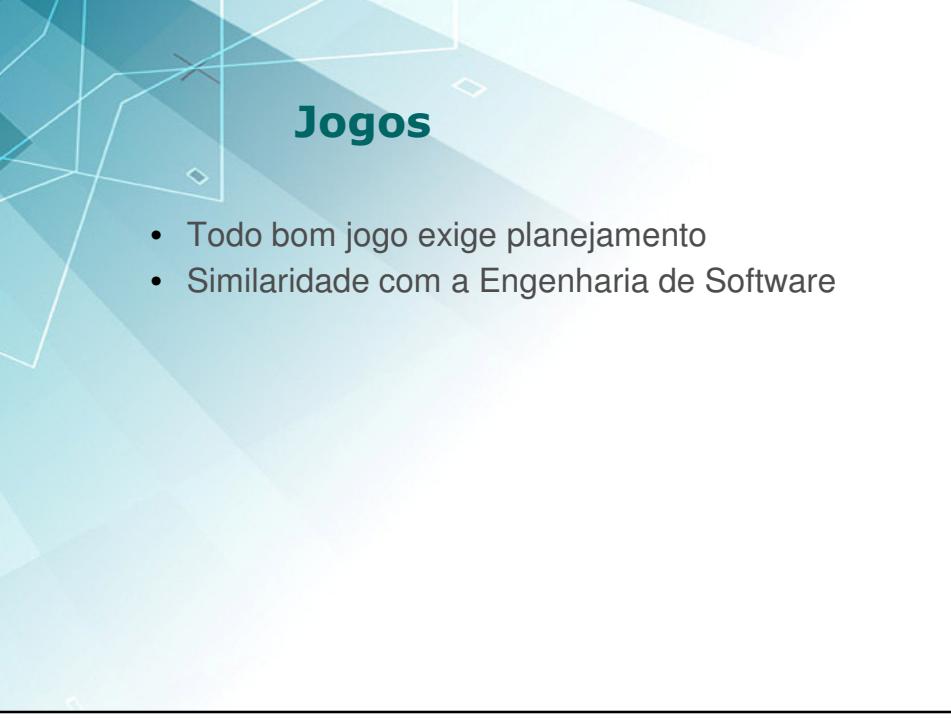
Recursos necessários Computação

- Inteligência Artificial
- Computação Gráfica
- Engenharia de Software
- Estruturas de Dados
- Programação
- Conceitos de Realidade Virtual



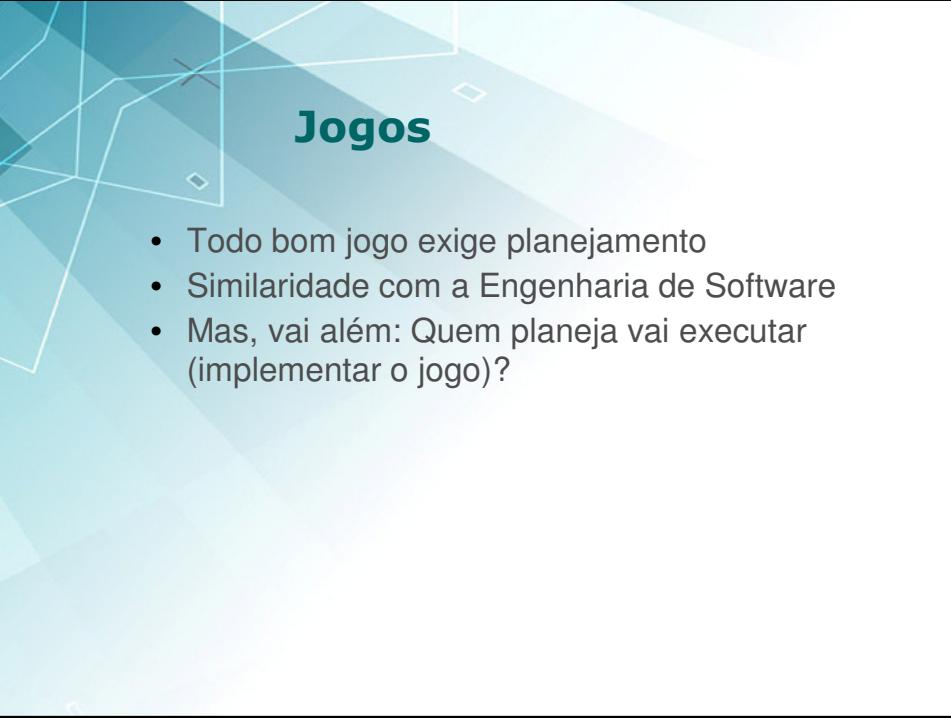
Como Desenvolver?

- Motivação
- Conhecimento a ser adquirido
- Necessidade de profissionais da área de escopo do jogo
 - escopo
 - formas de abordar o assunto
 - uso de equipamentos especiais
 - novas tecnologias
- Todo bom jogo exige planejamento



Jogos

- Todo bom jogo exige planejamento
- Similaridade com a Engenharia de Software



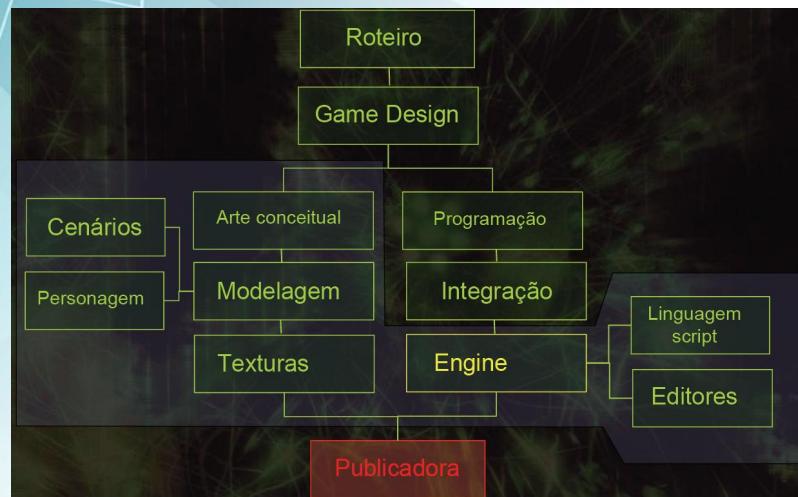
Jogos

- Todo bom jogo exige planejamento
- Similaridade com a Engenharia de Software
- Mas, vai além: Quem planeja vai executar (implementar o jogo)?

Jogos

- Todo bom jogo exige planejamento
- Similaridade com a Engenharia de Software
- Mas, vai além: Quem planeja vai executar (implementar o jogo)?
- Como isso é feito?
 - Equipe: pode conter dezenas de pessoas:
 - desenhistas, artistas, roteiristas, programadores, etc

Jogos



Jogos

- Documentação: todas as especificações do jogo
- => Design Bible
 - manual de instruções para os futuros desenvolvedores do jogo
- Processo de desenvolvimento não deve iniciar sem esta especificação pronta

Design Bible

- Roteiro
 - temática
 - objetivo
 - desafio / dificuldades
 - regras
- Conceituação Artística
 - fases
 - mapas
 - personagens e cenários
- Interface
 - modo de interação
 - abertura
- Programação
- Inteligência

Design Bible

Jogos

- Design Bible deve conter os seguintes elementos:
 - Roteiro
 - Game Design
 - Game Play
 - Interface

Jogos

- Roteiro
 - Assemelham-se a roteiros de filmes.
 - Fundamental para o processo de criação => deve mostrar seu diferencial em relação a outros jogos.
 - Entretenimento, desafio, gráficos, interação, imersão, etc
 - Roteiros de jogos são roteiros interativos
 - diferentemente que os roteiros de filmes, devem ter espaço para interferência do usuário no desenrolar da história.
 - Deve-se ter em conta qual o estilo do jogo que se está desenvolvendo.

Jogos

- Roteiro de Jogos Educacionais
 - Conceitos que serão passados ou cobrados do jogador-aluno
 - Aspecto lúdico => ensinar brincando
 - Jogo comercial => entretenimento
 - Jogo educacional => entretenimento com o significado educacional.

Jogos

- Game Design:
 - Conceituação artística do jogo.
 - Em geral, é escrita por um artista, devido a complexidade das histórias e dos cenários.
 - Características dos cenários e esboços de personagens (story-board),
 - descrição das texturas fundamentais,
 - mapas e descrições das fases.

Jogos

- Game Play:
 - Descreve a jogabilidade (regras do jogo) e o balanceamento das regras (game balancing).
 - É o guia os programadores em grande parte da programação.
 - Pode ser usada para nortear a inteligência do jogo.

Jogos

- Interface:
 - Divide-se em ingame e outgame.
 - Ingame: instrumentação disponível durante o jogo e é responsável pela entrada de dados do jogador para a aplicação.
 - Outgame é a forma de apresentar a introdução do jogo, sua configuração, instruções, etc.
 - A melhor interface é aquela que passa desapercebida para o jogador, permitindo que o mesmo possa focar-se no desenrolar da história e das ações.

Jogos

- Terminada a etapa de conceituação, o desenvolvimento de um game divide-se em dois caminhos distintos:
 - Criação artística: elaboração dos elementos que serão usados para sua montagem: modelos 3D, texturas, terrenos, sons, trilha sonora, entre outros;
 - Programação: implementação do motor do jogo (ou reutilização de um motor já existente) e implementação de protótipo(s).

Jogos

- Desenvolvimento
 - Linhas gerais do jogo. Story-boards. Conceitos
 - Implementação do Motor do Jogo (Engine) ou reutilização.
 - Realização das fases: artistas, projetistas de jogos (lógica das fases, comportamentos de personagens e do jogo - jogabilidade) e engenheiros de som.

Jogos

- Desenvolvimento
 - Depuração: Testes do jogo e documentação.
 - Distribuição do jogo.

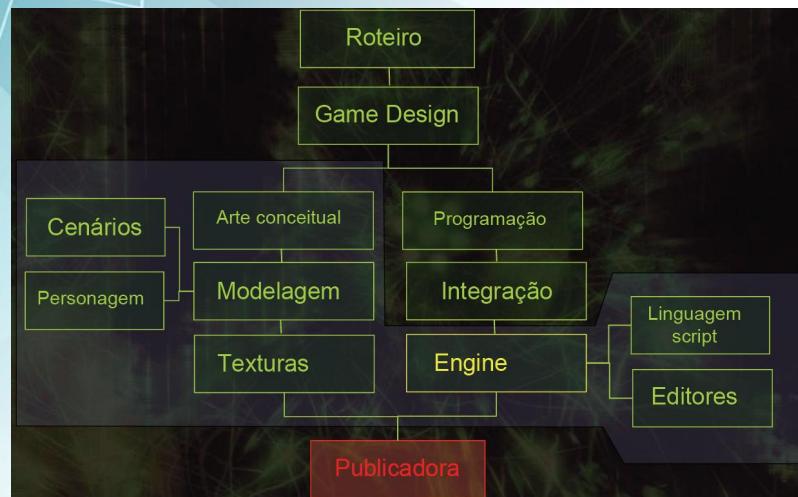
Jogos

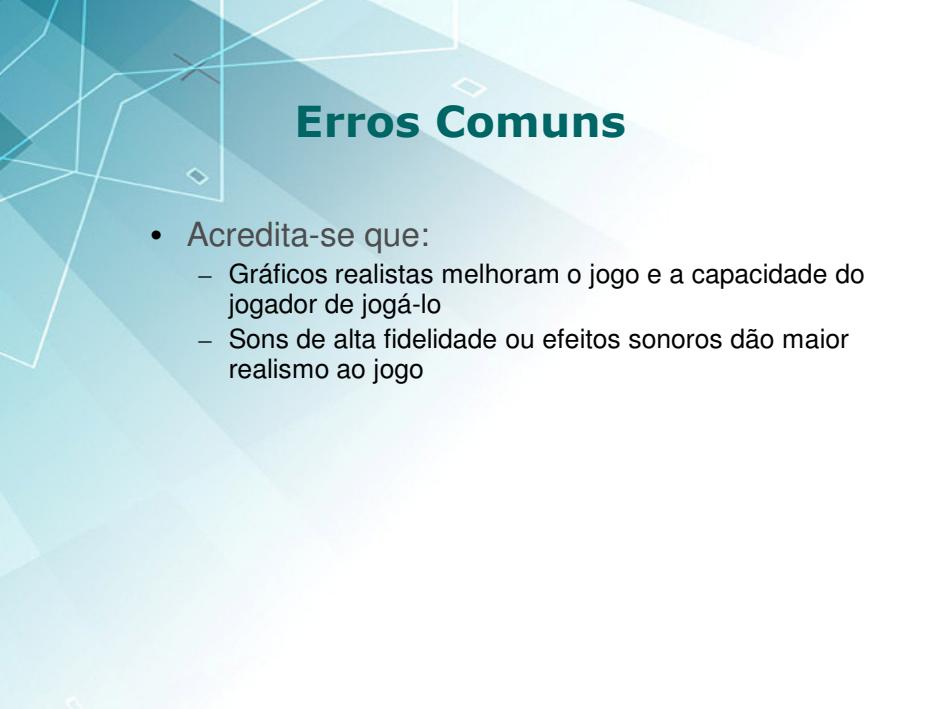
- Componentes da arquitetura de um jogo
 - Motor (engine):
 - renderização gráfica do jogo e coordenando de outros componentes
 - Rede:
 - comunicação com outros jogadores e/ou servidores
 - Som:
 - gerenciador de sons e trilha sonora do jogo

Jogos

- Componentes da arquitetura de um jogo
 - Inteligência Artificial:
 - implementa o controle e comportamento dos oponentes e aliados automatizados.
 - em jogos educacionais: também atua sobre os aspectos educacionais do jogo

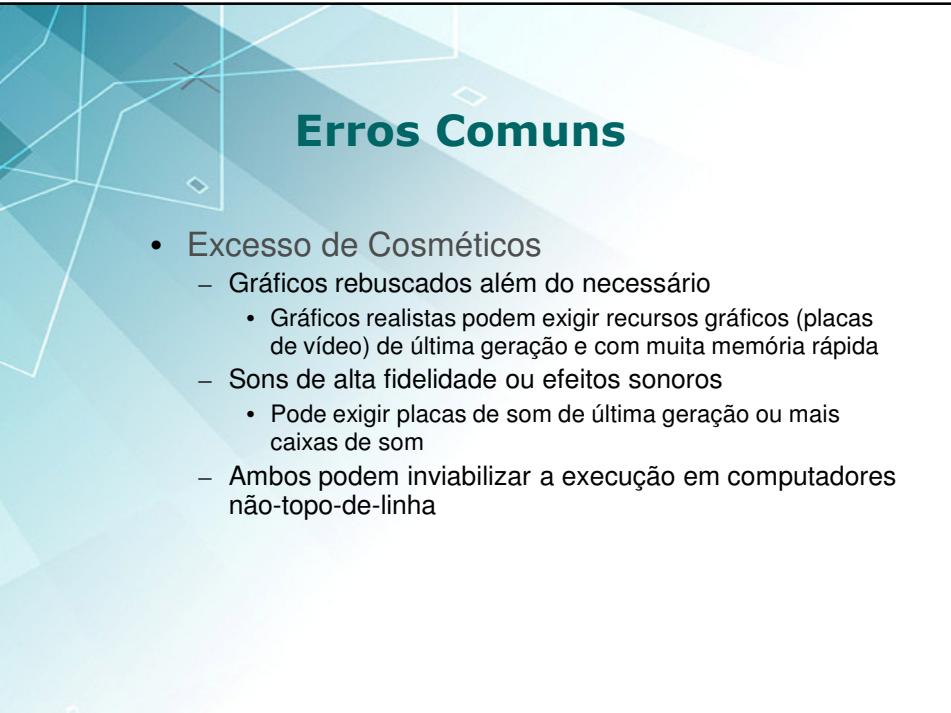
Jogos - Produção





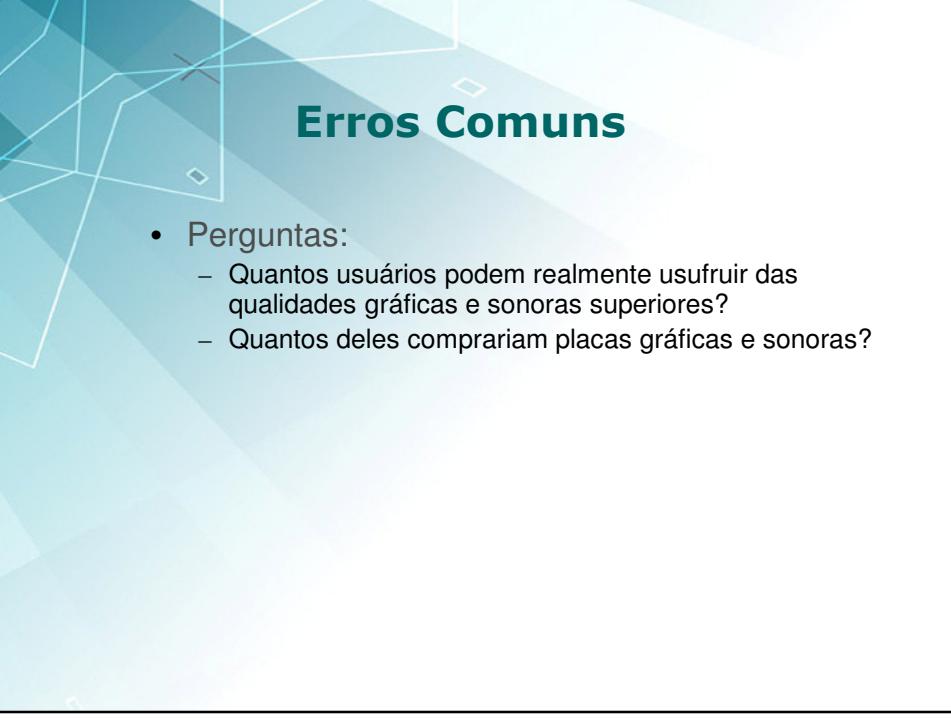
Erros Comuns

- Acredita-se que:
 - Gráficos realistas melhoram o jogo e a capacidade do jogador de jogá-lo
 - Sons de alta fidelidade ou efeitos sonoros dão maior realismo ao jogo



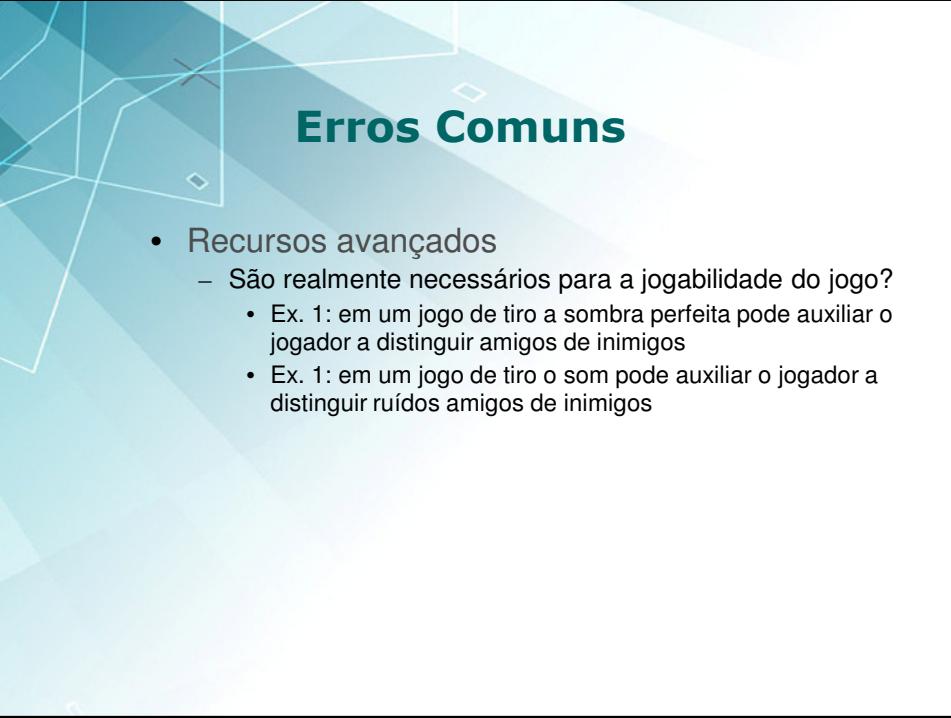
Erros Comuns

- Excesso de Cosméticos
 - Gráficos rebuscados além do necessário
 - Gráficos realistas podem exigir recursos gráficos (placas de vídeo) de última geração e com muita memória rápida
 - Sons de alta fidelidade ou efeitos sonoros
 - Pode exigir placas de som de última geração ou mais caixas de som
 - Ambos podem inviabilizar a execução em computadores não-topo-de-linha



Erros Comuns

- Perguntas:
 - Quantos usuários podem realmente usufruir das qualidades gráficas e sonoras superiores?
 - Quantos deles comprariam placas gráficas e sonoras?



Erros Comuns

- Recursos avançados
 - São realmente necessários para a jogabilidade do jogo?
 - Ex. 1: em um jogo de tiro a sombra perfeita pode auxiliar o jogador a distinguir amigos de inimigos
 - Ex. 1: em um jogo de tiro o som pode auxiliar o jogador a distinguir ruídos amigos de inimigos

Erros Comuns

- Lições do Cinema:
 - Shrek X Final Fantasy (2001)
- Estratégias diferentes:
 - criatividade no enredo e qualidade gráfica mediana
 - enredo pouco criativo, mas qualidade gráfica apurada



Erros Comuns

- Resultado final:
 - Shrek 10 X Final Fantasy 1



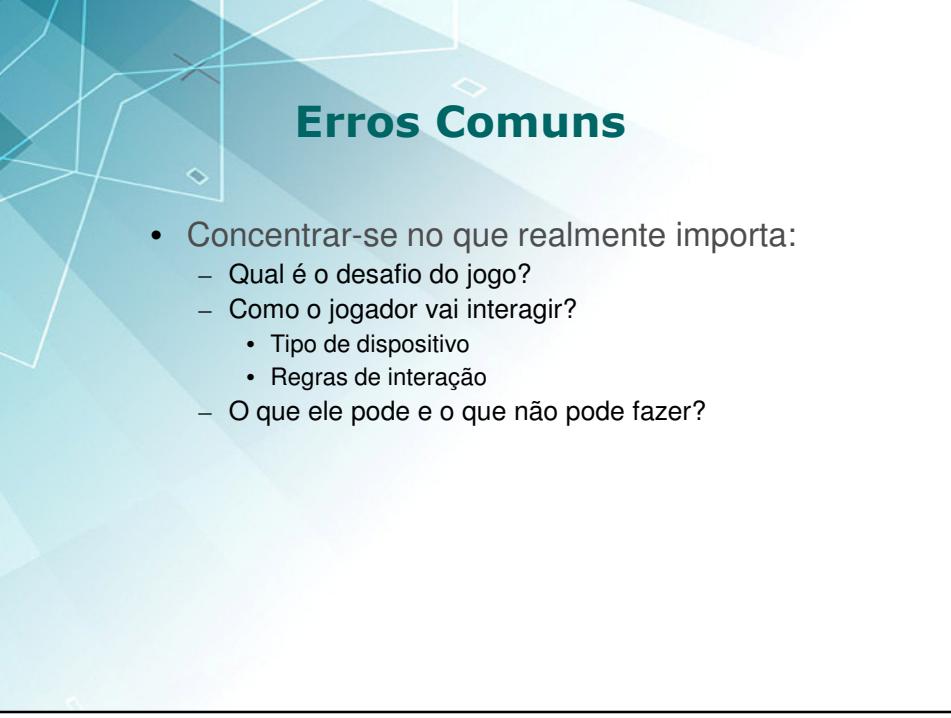
Erros Comuns

- Realismo gráfico (DOOM 1)
- Primeiro plano: melhor qualidade
- Segundo plano: ambiente escurecido com iluminação pobre



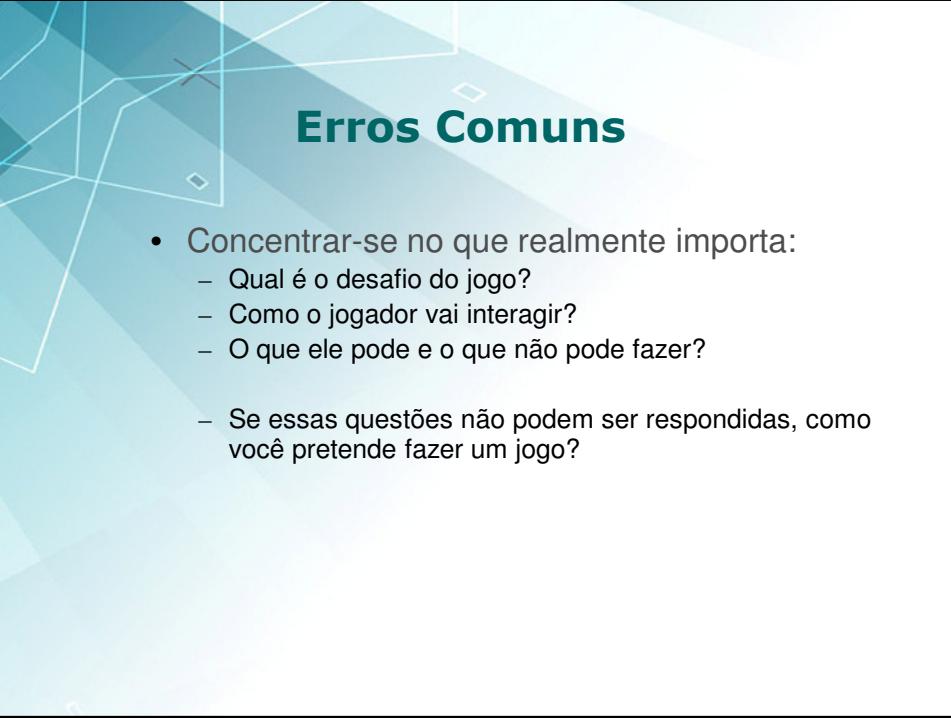
Erros Comuns

- Concentrar-se no que realmente importa:
 - Jogo deve ser interativo
 - Reflexos rápidos ou estratégia?
 - Lógica ou intuição?
 - Tentativa e erro?



Erros Comuns

- Concentrar-se no que realmente importa:
 - Qual é o desafio do jogo?
 - Como o jogador vai interagir?
 - Tipo de dispositivo
 - Regras de interação
 - O que ele pode e o que não pode fazer?



Erros Comuns

- Concentrar-se no que realmente importa:
 - Qual é o desafio do jogo?
 - Como o jogador vai interagir?
 - O que ele pode e o que não pode fazer?
 - Se essas questões não podem ser respondidas, como você pretende fazer um jogo?

Storyboard

- Organização do projeto:
 - Comunicação seqüencial: tela após tela na ordem correta
 - Limita a liberdade do jogador durante o jogo => foco na estória
 - É a antítese da interatividade
 - Coleção deles para os pontos-chave do jogo

Ferramentas

- Desenvolvimento de *serious games*:
 - grupo multidisciplinar
 - conceitos particulares do tema abordados
 - muitos detalhes

Ferramentas

- Engines (gráficas, sons, IA), modeladores, etc
- Facilitam o desenvolvimento
- Aceleram o cumprimento das tarefas,
- mas...

Ferramentas

- Toma tempo construir ferramentas e esse custo não é produtivo
- Ferramentas são desenvolvidas para tarefas específicas
- Requer tempo para aprender a usá-la correta e eficientemente
- Podem custar entre algumas centenas de dólares a milhões

Ferramentas

- Como fazer para desenvolvimento mais produtivo?
 - engines de jogos
 - bibliotecas
 - toolkits
 - frameworks
- Não há padronização nesses recursos

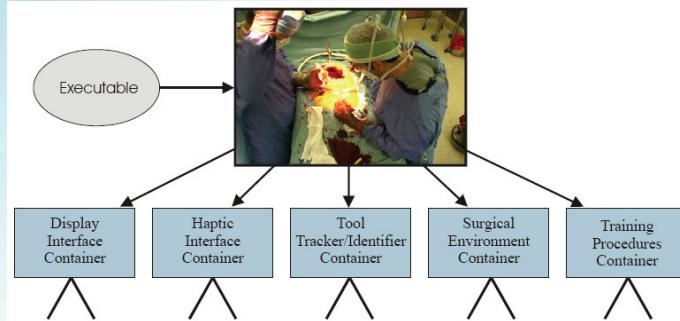
Bibliotecas

- Conjunto de classes
 - Suporte a tarefas de baixo-nível
 - Controle de dispositivos
- MVL – Universidade de Tsukuba
 - Manipulação múltipla
 - Deformação interativa
 - Suporte a dispositivos hapticos
 - Imersividade

Bibliotecas

- SSTML

- Simulação de cirurgias abertas
- XML para combinar diferentes linguagens



Toolkits

- ARToolkit

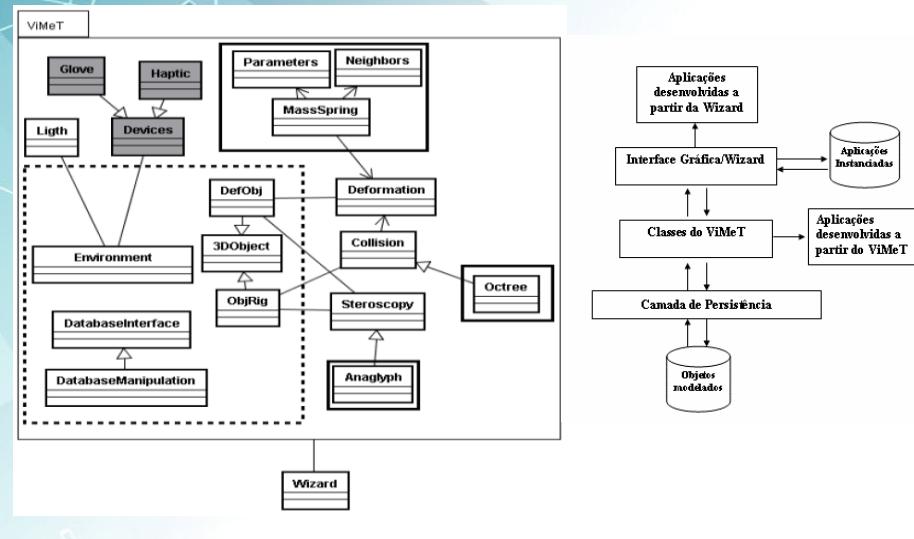
- Suporte a captura de imagens
- Reconhecimento de Padrões
- Rastreamento
- Utilizado para realidade aumentada

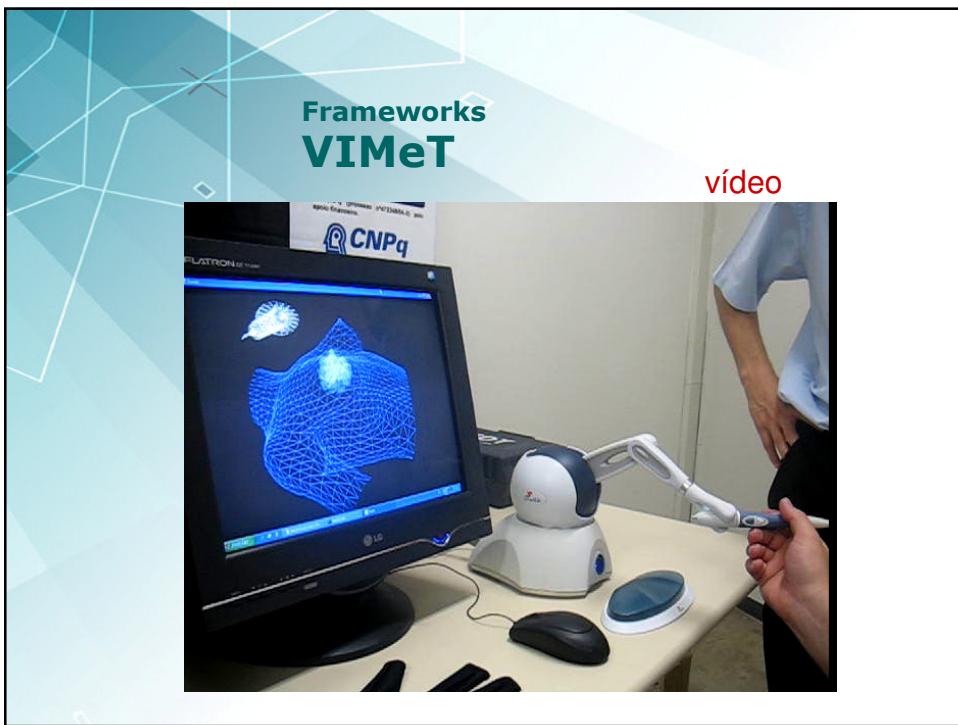
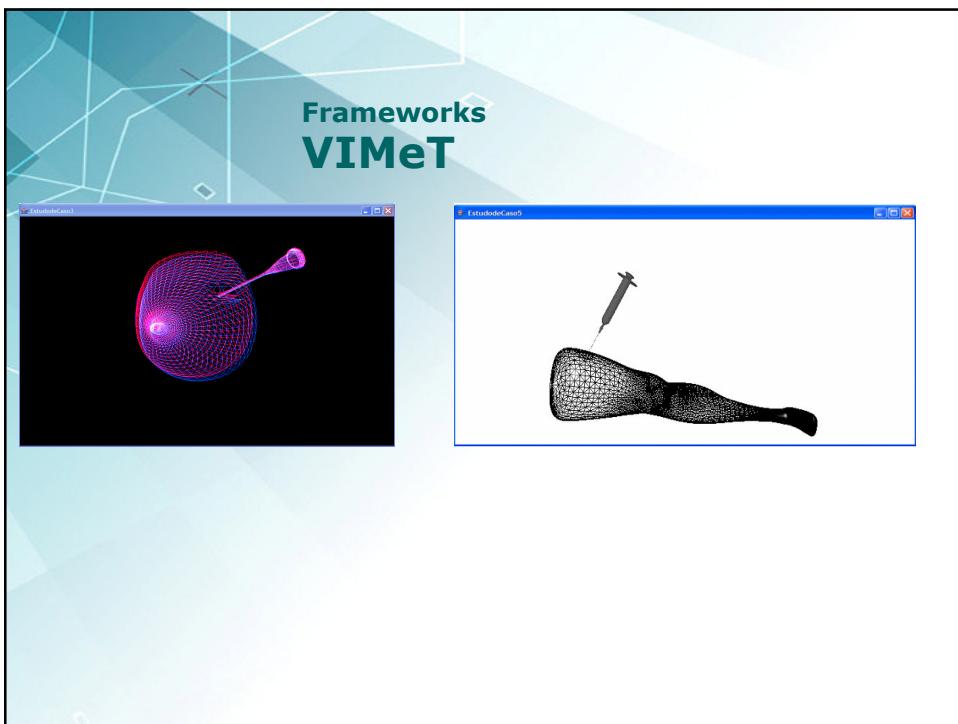


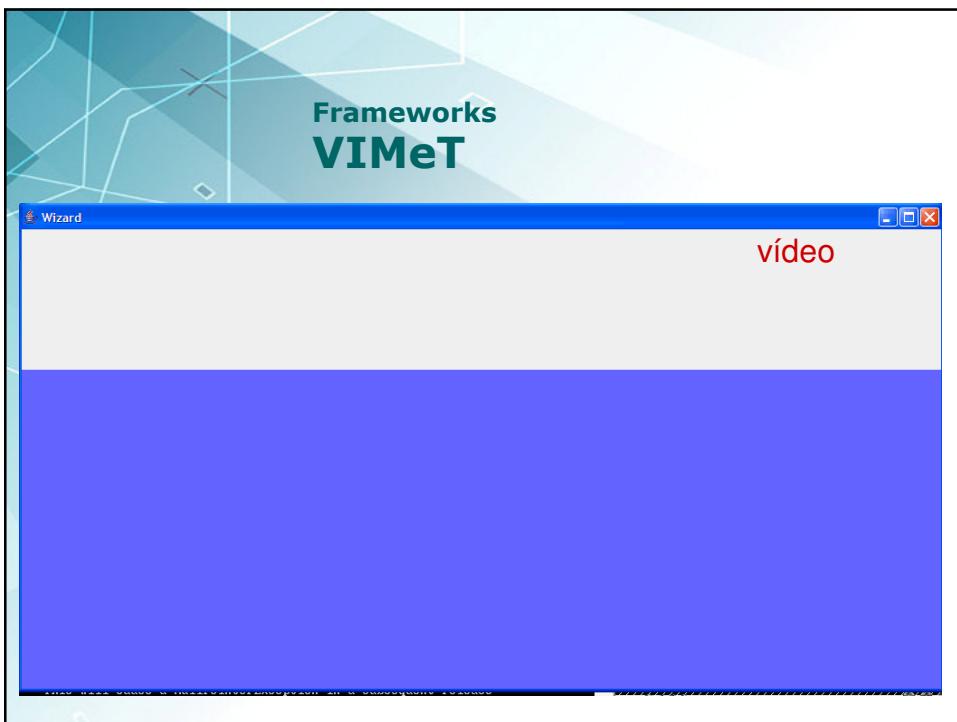
Frameworks

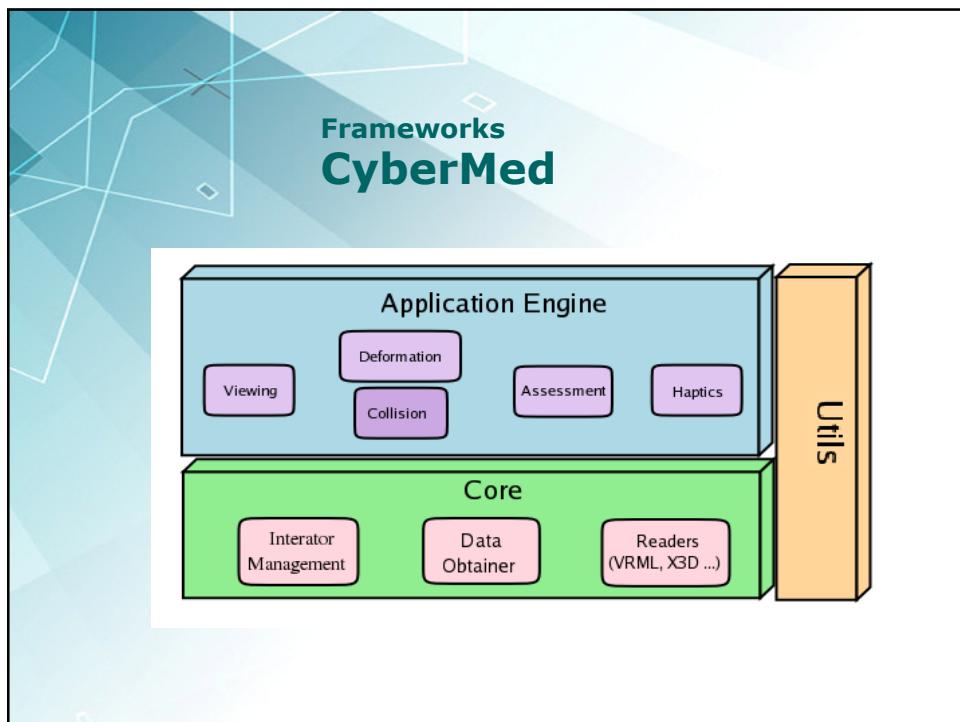
- ViMeT
 - voltado a exames de biópsia
 - Suporte a
 - Estereoscopia
 - Detecção de colisão
 - Deformação
 - Dispositivos não-convencionais
 - Modelagem 3D
 - Uso de linguagem Java
 - ViMeTWizard
 - Instanciação automática de classes

Frameworks ViMeT









Frameworks - outros

- **Avango**—aplicações distribuídas de ambientes virtuais (AVs) interativos.
- **basho**—renderizador de grafo de cena, baseado em *OpenSceneGraph*.
- **IVORY**—visualização de informação baseada em física (força, massa, aceleração, potência, energia, entre outros), com classes implementadas em Java e objetos em VRML 2.

Frameworks - outros

- **SOFA**—aplicações de simulação de procedimentos médicos
- **ViRAL (Virtual Reality Abstraction Layer)**—aplicações de RV que sejam operadas por interfaces WIMP (*Windows, Icons, Menus and Pointing Device*).
- **VPat (Virtual Patients)**—visualização, simulação de movimento, aplicações de RV.

Engines

- Gerenciamento do fluxo do código
- Simulação de física e/ou suporte a diferentes plataformas.
- Suporte a sistemas imersivos disponível em algumas delas
- O que falta:
 - mecanismos de inteligência mais apurados
 - suporte a dispositivos hápticos

Engines

- OGRE
 - diferentes plataformas
 - diversas configurações.
- Panda3D
 - linguagem de *script* Python.
 - rotinas de renderização gráfica, suporte a estereoscopia com anaglifo e multiplexação

Engines

- Panda3D



Geografia



Geometria espacial



Personagens

- PC – Player Character
 - Controlador
 - Não autonomo
 - Responde a ações definidas pelo usuário
Teclado, Joystick, Mouse
- NPC –NonPlayer Character
 - Controlador
 - Autonomo
 - Responde automaticamente à situações do cenário –
Máquinas de Estado

Personagens

- Controladores
 - Responsáveis por coordenar as ações tomadas pelos personagens
 - Pode-se ter controladores específicos para diferentes propósitos
 - Um para locomoção
 - Outro para conversação
 - Integram o engine de simulação

Controladores de Personagens

- Diferentes arquiteturas
 - Controladores independentes
 - Cada ação é controlada de forma independente das demais ações
 - Controladores com dependência hierárquica
 - Uma ação depende da existência de um contexto na hierarquia dos controladores



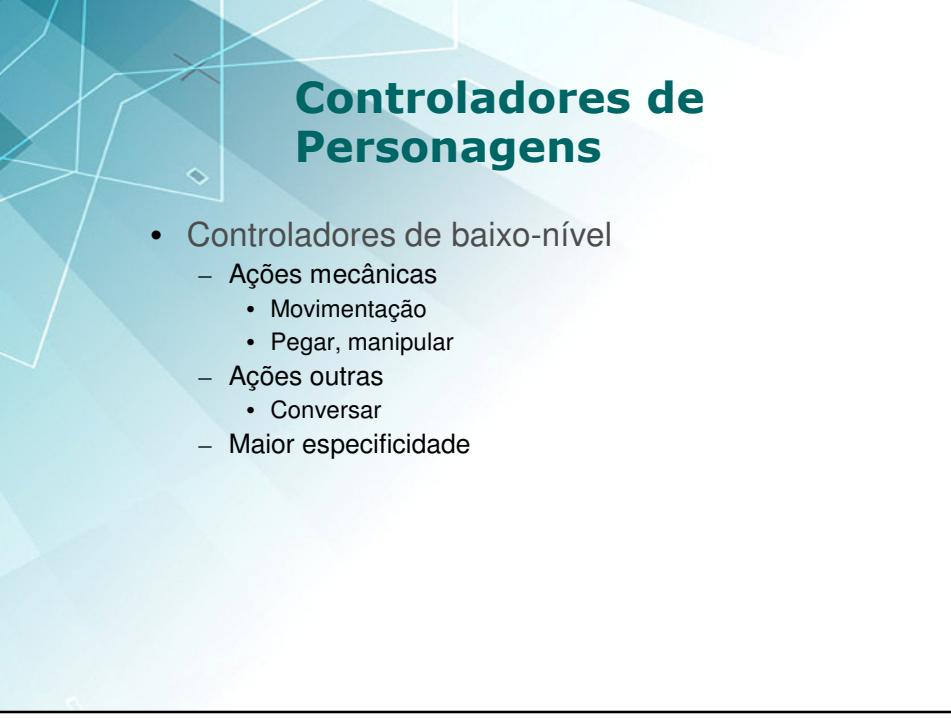
Controladores de Personagens

- Diferentes arquiteturas
 - Configuração em pilha
 - Ações são tomadas apenas se uma seqüência de eventos forem verificados



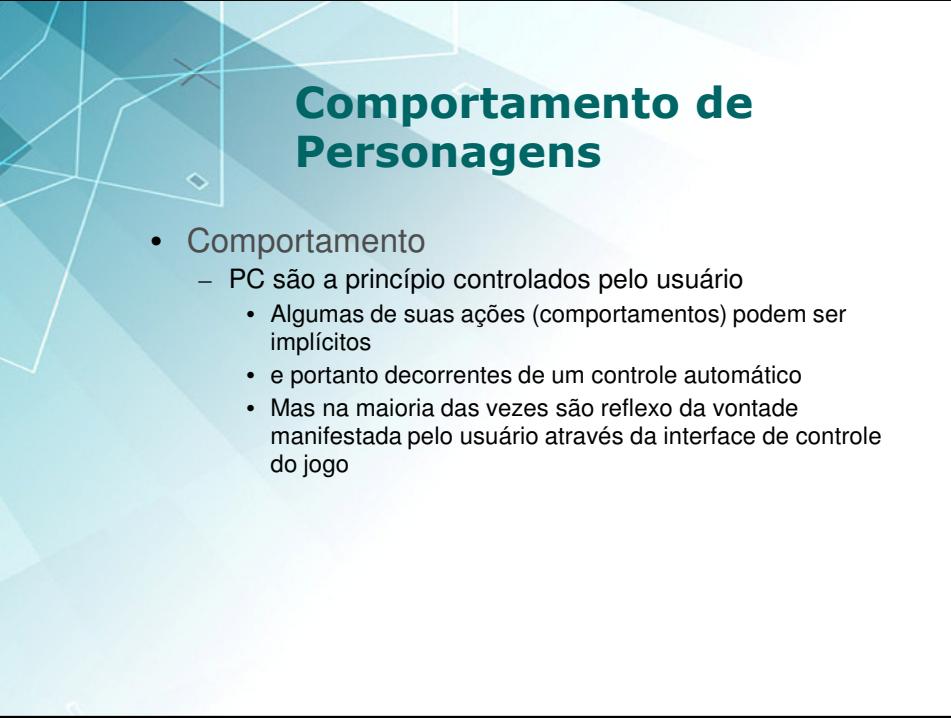
Controladores de Personagens

- Controladores de alto-nível
 - Tomada de decisão
 - Coordena outros controladores
 - Planejamento estratégico
 - Maior generalidade



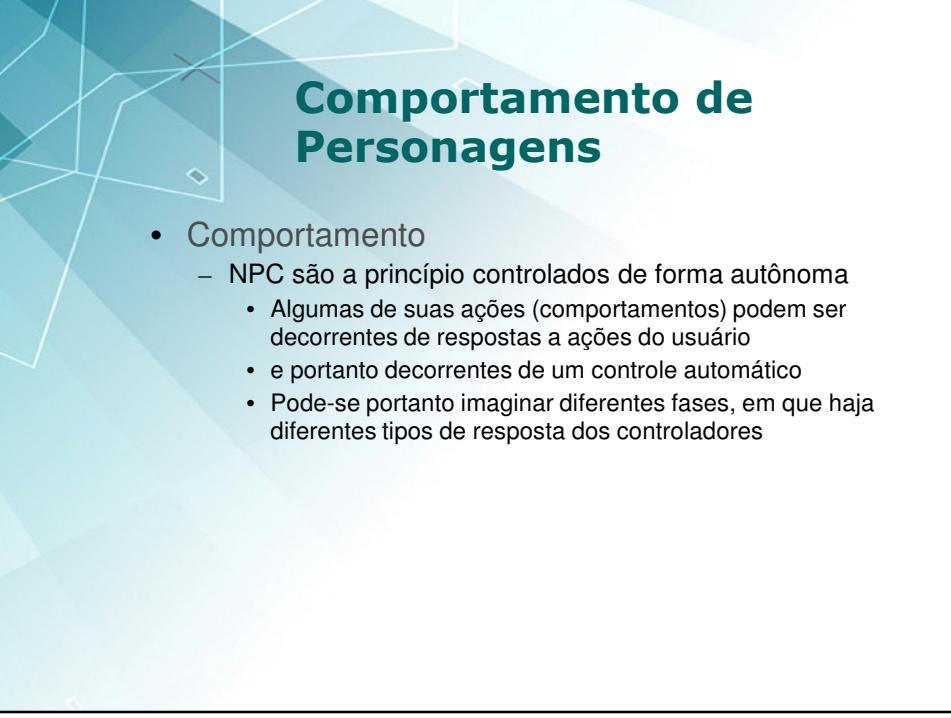
Controladores de Personagens

- Controladores de baixo-nível
 - Ações mecânicas
 - Movimentação
 - Pegar, manipular
 - Ações outras
 - Conversar
 - Maior especificidade



Comportamento de Personagens

- Comportamento
 - PC são a princípio controlados pelo usuário
 - Algumas de suas ações (comportamentos) podem ser implícitos
 - e portanto decorrentes de um controle automático
 - Mas na maioria das vezes são reflexo da vontade manifestada pelo usuário através da interface de controle do jogo



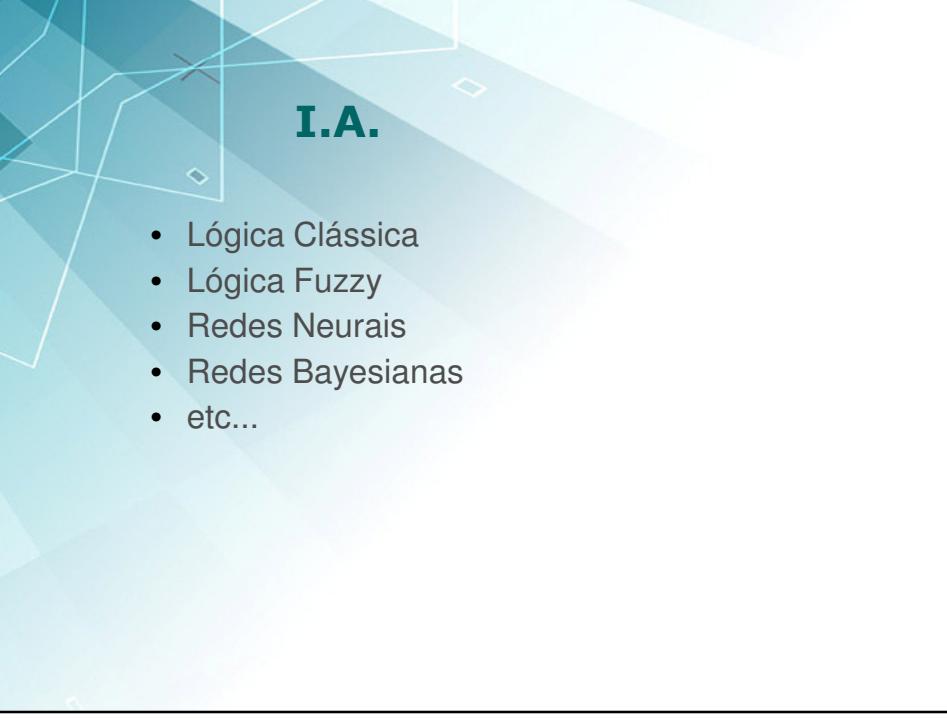
Comportamento de Personagens

- Comportamento
 - NPC são a princípio controlados de forma autônoma
 - Algumas de suas ações (comportamentos) podem ser decorrentes de respostas a ações do usuário
 - e portanto decorrentes de um controle automático
 - Pode-se portanto imaginar diferentes fases, em que haja diferentes tipos de resposta dos controladores



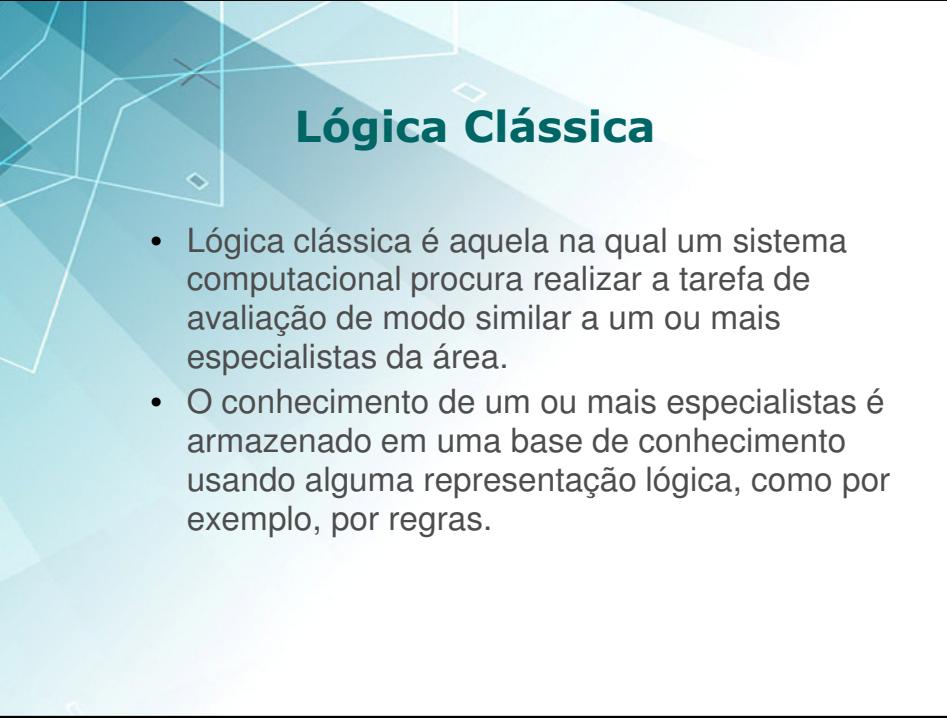
Controladores

- Inteligência Artificial
 - Controle de nível superior
 - Tomada de decisões



I.A.

- Lógica Clássica
- Lógica Fuzzy
- Redes Neurais
- Redes Bayesianas
- etc...



Lógica Clássica

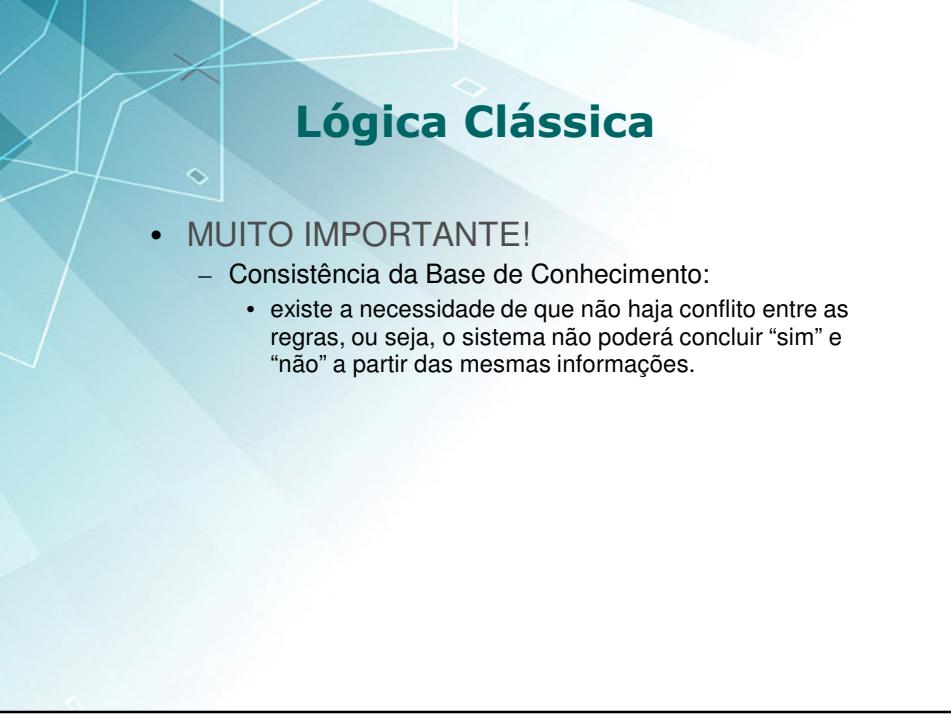
- Lógica clássica é aquela na qual um sistema computacional procura realizar a tarefa de avaliação de modo similar a um ou mais especialistas da área.
- O conhecimento de um ou mais especialistas é armazenado em uma base de conhecimento usando alguma representação lógica, como por exemplo, por regras.

Lógica Clássica

- Regras são formulações lógicas do tipo:
 - SE <condição> ENTÃO <conclusão>.
- Uma regra pode ser composta por conjunções e disjunções e pode levar a mais de uma conclusão simultaneamente. Por exemplo:
 - SE (<condição1> OU <condição2>) E (<condição3>) ENTÃO (<conclusão1> E <conclusão2>).

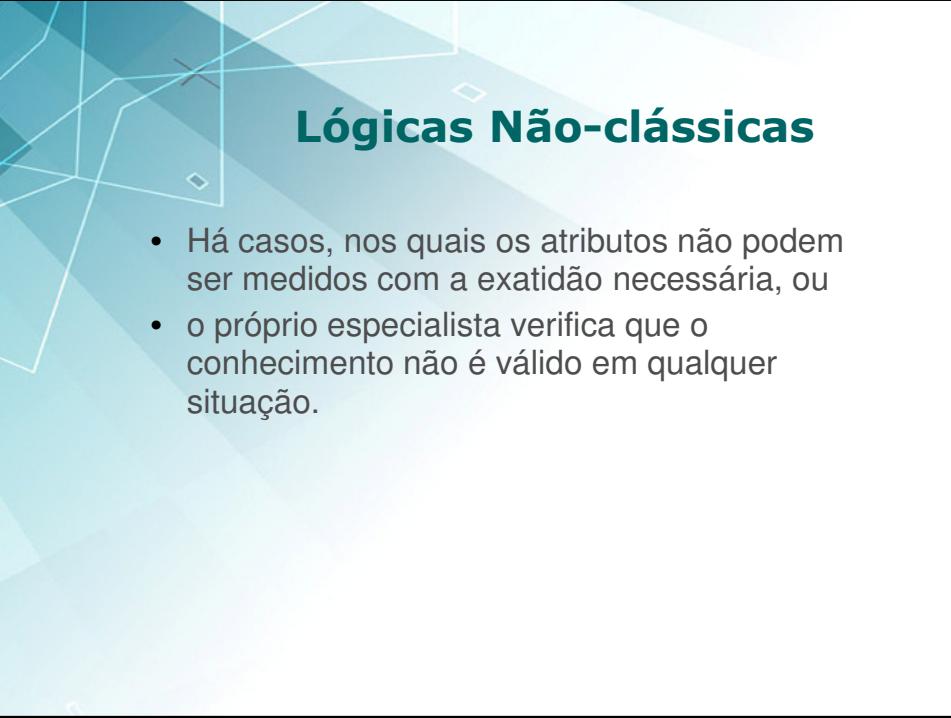
Lógica Clássica

- A regra acima com mais de uma conclusão é equivalente a um conjunto de regras com apenas uma conclusão:
 - SE (<condição1> OU <condição2>) E (<condição3>) ENTÃO <conclusão1>
 - SE (<condição1> OU <condição2>) E (<condição3>) ENTÃO <conclusão2>.



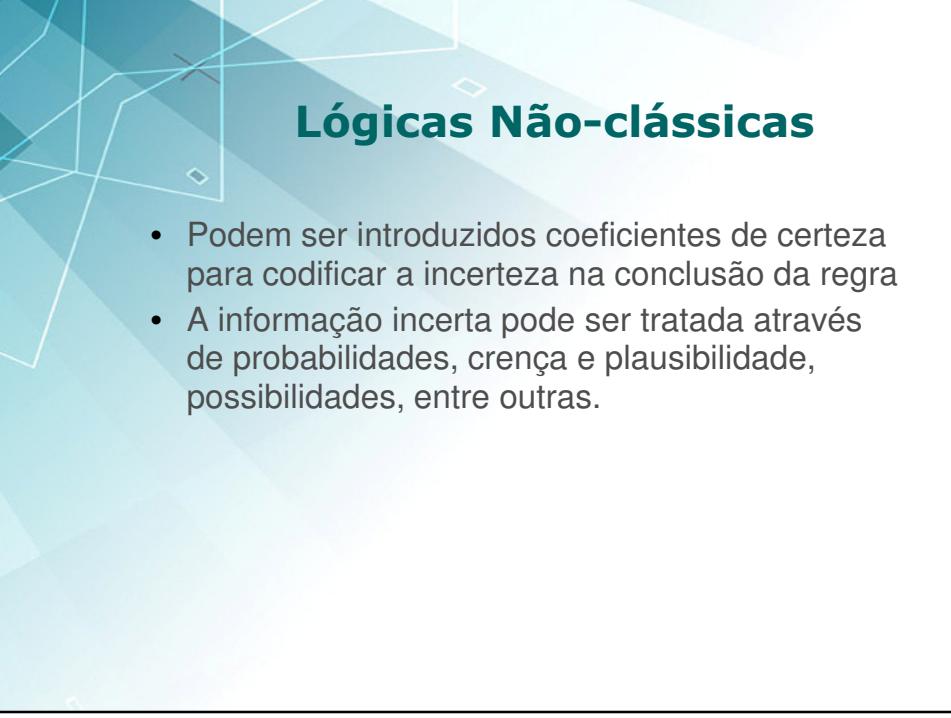
Lógica Clássica

- MUITO IMPORTANTE!
 - Consistência da Base de Conhecimento:
 - existe a necessidade de que não haja conflito entre as regras, ou seja, o sistema não poderá concluir “sim” e “não” a partir das mesmas informações.



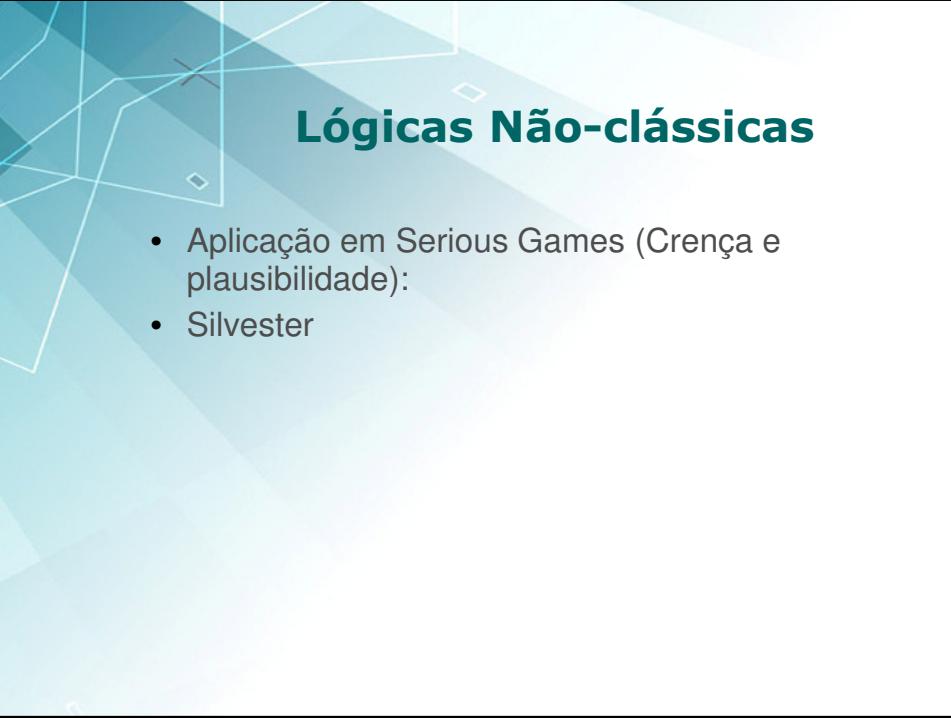
Lógicas Não-clássicas

- Há casos, nos quais os atributos não podem ser medidos com a exatidão necessária, ou
- o próprio especialista verifica que o conhecimento não é válido em qualquer situação.



Lógicas Não-clássicas

- Podem ser introduzidos coeficientes de certeza para codificar a incerteza na conclusão da regra
- A informação incerta pode ser tratada através de probabilidades, crença e plausibilidade, possibilidades, entre outras.



Lógicas Não-clássicas

- Aplicação em Serious Games (Crença e plausibilidade):
- Silvester

Lógica Fuzzy

- Definição de conjunto na teoria dos conjuntos clássica:
 - um conjunto A de um universo X pode ser definido por uma função de pertinência $\square A(x)$, com $\square A: X \rightarrow \{0,1\}$, onde 1 significa que um dado $x \in X$ está incluído em A e 0 significa que x não pertence a A.

Lógica Fuzzy

- Porém, para um conjunto “fuzzy” não se pode afirmar sempre que um dado elemento $x \in X$ pertence ou não ao conjunto A.
- Nova definição:
 - $\square A(x)$ onde $\square A: X \rightarrow [0,1]$, ou seja,
 - podem existir graus de pertinência parciais para elementos de X em A

Lógica Fuzzy

- Um conjunto “fuzzy” A de X é dito ser “preciso” quando $c^* \in X$ tal que $\mu_A(c^*)=1$ e $c \neq c^*$, $\mu_A(c)=0$.
- Um conjunto “fuzzy” A é dito ser exato (ou “crisp”), quando $c \in X$, $\mu_A(c) \in \{0,1\}$.

Lógica Fuzzy

- Um exemplo de regra fuzzy:
 - Se Inimigo é Fraco E Castelo é destruído OU Castelo é não_construído E Força_Aliados é forte E Número_de_Tropas_Disponíveis é Grande Então Condição_Ação é Atacar_Inimigo.

Lógica Fuzzy

- Segundo Mandani (1975), a avaliação das proposições com conectivo E e OU são realizadas de acordo com as operações sobre conjuntos fuzzy de intersecção e união.
- A intersecção e a união de dois conjuntos “fuzzy” são executadas através do uso dos operadores de t-normas e t-conormas, respectivamente

Lógica Fuzzy

- Tabela 1 – T-normas e t-conormas duais

T-normas	T-conormas
$\min(a, b)$	$\max(a, b)$
$a \cdot b$	$a + b - ab$
$\max(a + b - 1, 0)$	$\min(a + b, 1)$
$\begin{cases} a, & \text{se } b = 1 \\ b, & \text{se } a = 1 \\ 0, & \text{senão} \end{cases}$	$\begin{cases} a, & \text{se } b = 0 \\ b, & \text{se } a = 0 \\ 1, & \text{senão} \end{cases}$

Lógica Fuzzy

- Uma t-norma \square (respec. t-conorm \square) tem 1 (respec. 0) como elemento neutro (por exemplo: $\square=\min$, $\square=\max$) [Dubois e Prade, 1988].

Lógica Fuzzy

- A intersecção de dois conjuntos “fuzzy” A e B com funções de pertinência $\square A(x)$ e $\square B(x)$ é um conjunto “fuzzy” C com função de pertinência dada por:
 - $C=A \cap B \quad \square C(x)=\min\{\square A(x), \square B(x)\}, \quad x \in X.$

Lógica Fuzzy

- A união de dois conjuntos “fuzzy” A e B com funções de pertinência $\square A(x)$ e $\square B(x)$ é um conjunto “fuzzy” C com função de pertinência dada por:
 - $C = A \sqcup B \sqcup C(x) = \{\square A(x), \square B(x)\}, x \in X.$

Lógica Fuzzy

- O complemento de um conjunto “fuzzy” A em X, denotado por $\sim A$ é definido como:
 - $\square \sim A(x) = n(\square A(x)), x \in X.$
- onde: $n: [0,1] \rightarrow [0,1]$ é um operador de negação que satisfaz as seguintes propriedades:
 - $n(0)=1$ e $n(1)=0$
 - $n(a) \leq n(b)$ se $a > b$
 - $n(n(a))=a, x \in [0,1]$

Lógica Fuzzy

- Um operador de negação é chamada “estrito” se é continua e também satisfaz:
 - $n(a) < n(b)$ se $a > b$.

Lógica Fuzzy

- Um operador de negação é chamada “estrito” se é continua e também satisfaz:
 - $n(a) < n(b)$ se $a > b$.
- O principal operador de negação que satisfaz as quatro condições acima é $n(a) = 1-a$.

Lógica Fuzzy

- A função de implicação entre dois conjuntos “fuzzy” A e B com funções de pertinência $\square A(x)$ e $\square B(x)$ é um conjunto “fuzzy” C com função de pertinência dada por:
 - $C = A \square B \quad \square C(x,y) = \square\{\square A(x), \square B(y)\}, \quad x \in X, \quad y \in Y$
- onde $\square : [0,1]^2 \rightarrow [0,1]$ é um operador de implicação que obedece as seguintes propriedades:

Lógica Fuzzy

- $a, a', b, b' \in [0,1]$:
- Se $b \leq b'$ então $\square(a,b) \leq \square(a,b')$;
- $\square(0,b)=0$;
- $\square(1,b)=b$.
- As implicações puras obedecem também a:
- Se $a \leq a'$ então $\square(a,b) \leq \square(a',b)$;
- $\square(a, \square(b,c)) = \square(b, \square(a,c))$.

Lógica Fuzzy

- Quando se usa uma t-norma \square para representar o conectivo E e uma t-conorma \Box para representar o conectivo OU, a implicação \rightarrow é dada por uma t-norma \Box , como por exemplo min (Mandani, 1975)
- Neste caso:
 - $C = A \rightarrow B \rightarrow C(x,y) = \min \{\Box A(x), \Box B(y)\}, \quad x \in X, \quad y \in Y$

Lógica Fuzzy

- De acordo com a escolha da t-norma e t-conorma dual para representar o conectivo E e OU, usa-se uma t-norma para a implicação
- Para essa escolha é importante ver a Tabela 1

Lógica Fuzzy

- Tabela 2 – Operadores de Implicação

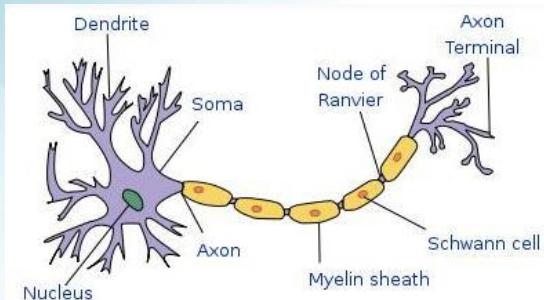
Operador de Implicação	Nome
$\max(1 - a, b)$	Kleene-Dienes
$\min(1 - a + b, 1)$	Lukasiewicz
$1 - a + ab$	Reichenback
$\min(a, b)$	Mandani

Redes Neurais

- Existem várias formas e tipos de Redes Neurais para diversas aplicações
- Rede Neural “Multi-Layer Perceptron” (MLP) é indicada para avaliação de contextos e inteligência artificial

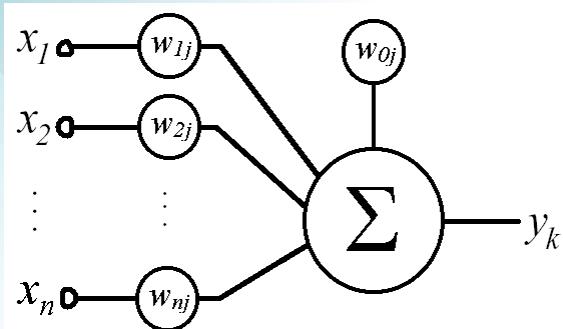
Redes Neurais

- Inspiração Biológica



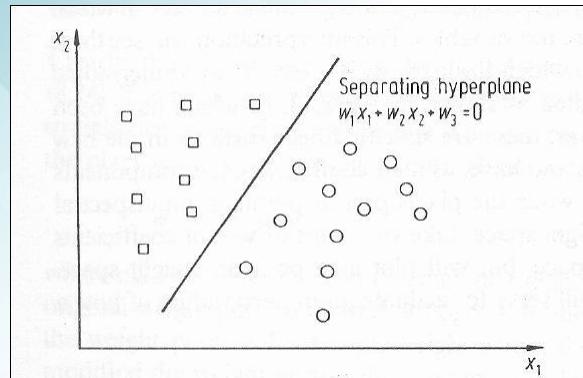
Redes Neurais

- Inspiração Biológica (Perceptron)



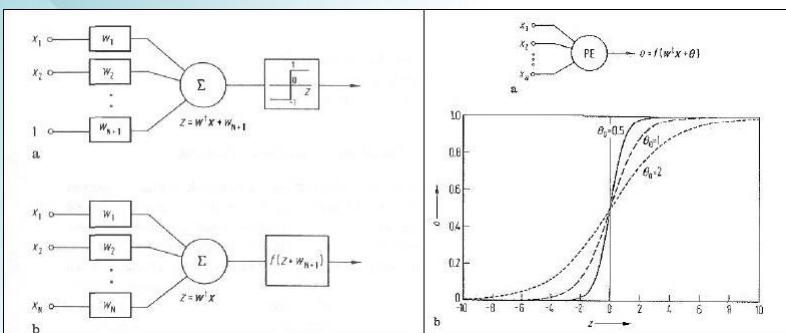
Redes Neurais

- O que o Perceptron é



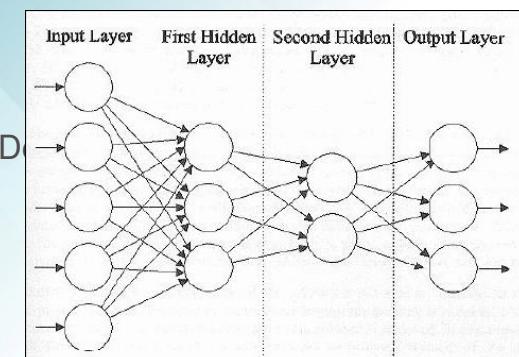
Redes Neurais

- Modelos:



Redes Neurais

- Rede Neural MLP



Redes Neurais

- Existem basicamente três tipos de camadas:
 - a camada de entrada X, com vetores de entrada x_1, x_2, \dots, x_n ;
 - a camada intermediária ou escondida, que pode conter um ou mais de um vetor de neurônios e
 - a camada de saída Y, y_1, y_2, \dots, y_k saídas.

Redes Neurais

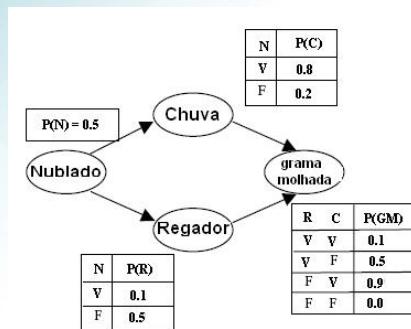
- Treinamento (aprendizado) da Rede: vários métodos existentes – mais popular:
 - Retropropagação do Erro (Backpropagation)
 - método supervisionado de treinamento
 - A partir de uma entrada, estima a saída e calcula o erro. Esse erro é retropropagado para a rede que recalcula os pesos até o acerto ou erro mínimo geral da rede.

Redes Bayesianas

- Como nas Redes Neurais, existem várias formas e tipos de Redes Bayesianas
- A maioria delas é indicada para avaliação de contextos e inteligência artificial

Redes Bayesianas

- O Método Naive Bayes (NB), também chamado Discreto ou Multinomial NB, é um método robusto.



Redes Bayesianas

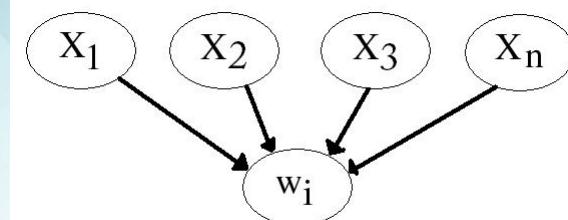
- Formalmente, sejam as classes de desempenho w_i , no espaço de decisões $\square=\{1, \dots, M\}$, onde M é o número total de decisões.
- O NB computa as probabilidades condicionais das decisões e então prediz a mais provável para o vetor de dados de entrada X , onde X é um vetor com n atributos, ou seja, $X=\{X_1, X_2, \dots, X_n\}$.

Redes Bayesianas

- O Naive Bayes recebe este nome devido à ingênuia hipótese de que cada vetor de atributos X_k é condicionalmente independente de qualquer outro vetor de atributos X_l , para todo $k \neq l \in n$.
- Isto significa que o conhecimento da classe é suficiente para determinar a probabilidade de um valor X_k .
- Deve-se lembrar que isto nem sempre é verdadeiro

Redes Bayesianas

- Grafo de uma Naive Bayes



Redes Bayesianas

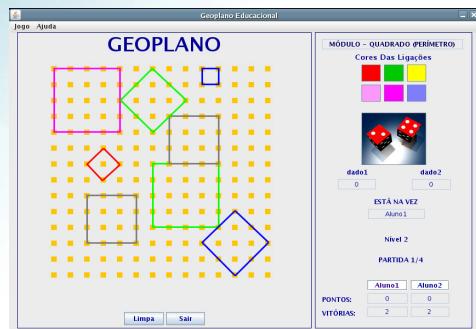
- $P(w_i | X_1, X_2, \dots, X_n) = (1/S) P(w_i) \prod_{k=1}^n P(X_k | w_i)$
- Regra de Decisão:
- $X \in w_i$ se $P(w_i | X_1, X_2, \dots, X_n) > P(w_j | X_1, X_2, \dots, X_n)$ para todo $i \neq j$ e $i, j \in \Omega$

Redes Bayesianas

- Para estimar os parâmetros para $P(X_k | w_i)$ para cada classe w_i , pode-se usar um estimador de máxima verossimilhança, chamado Pe:
 - $Pe(X_k | w_i) = \#(X_k, w_i) / \#(w_i)$
 - onde $\#(X_k, w_i)$ é o número de casos amostrais pertencentes à decisão w_i tendo o valor X_k , $\#(w_i)$ é o número de casos amostrais que pertencem à decisão w_i .

Redes Bayesianas

- Aplicação em Serious Games:
- Jogo Educacional GeoPlanoPEC



Redes Bayesianas

- Aplicação: Age of Empires



Controladores

- Controle de nível inferior
 - Tomada de alguns tipos de decisões (agentes)
 - Máquinas de Estados Finitos

Máquinas de Estados Finitos (MEF)

- Não se enquadram propriamente na categoria de sistemas inteligentes, mas são técnicas muito utilizadas em jogos de computador.
- Em geral, associa-se um conjunto de ações a serem tomadas a cada estado da MEF.
- As ações podem ser implementadas diretamente na linguagem do motor do jogo, ou por "script".
- O uso de "script" objetiva trazer maior comodidade ao processo de implementação da IA de um jogo.

Máquinas de Estados Finitos (MEF)

- Formalmente, uma MEF é definida por uma tupla $(E; X; f; x_0; F)$, onde:
 - E é um conjunto finito de eventos
 - X é um conjunto de estados finito
 - f é uma função de transição de estado, dado por $f : X \times E \rightarrow X$
 - x_0 é o estado inicial, com $x_0 \in X$
 - F é um conjunto de estados finais, com $F \subseteq E$

Máquinas de Estados Finitos (MEF)

- Pode-se representar uma MEF de forma gráfica, através de um diagrama de transição de estados.
- Este diagrama é um grafo direcionado, no qual os nós representam os estados e os arcos representam os eventos.
- Um arco rotulado como e que conecta dois nós saindo de x e indo para x' representa uma transição do estado x para o estado x' quando da ocorrência do evento e .

Máquinas de Estados Finitos (MEF)

- Exemplo simples:



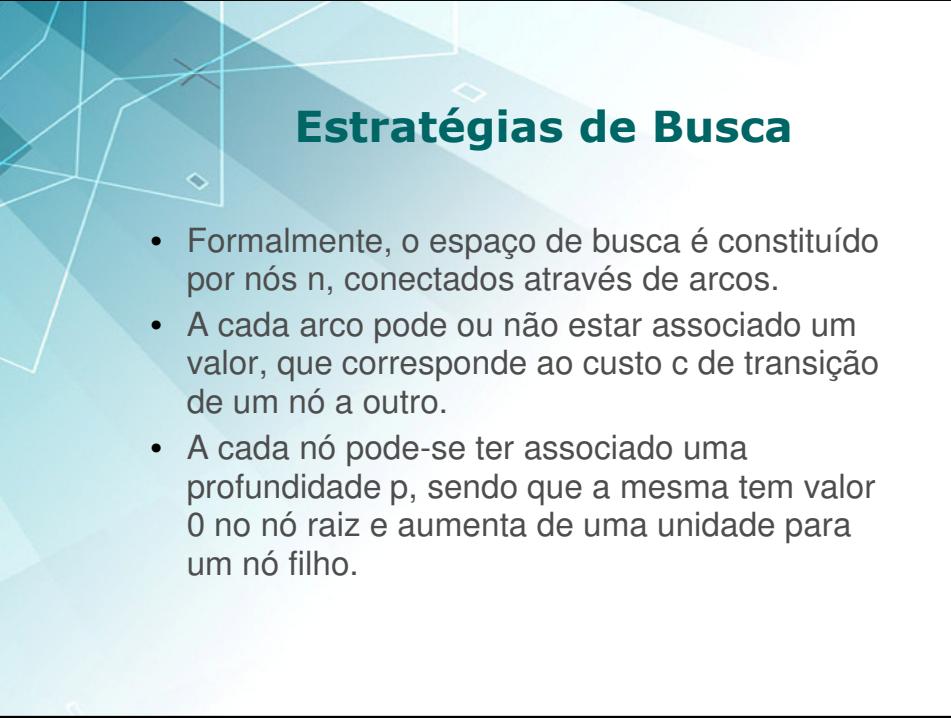
Máquinas de Estados Finitos (MEF)

- Aplicação em Serious Games:
 - GeoespaçoPEC



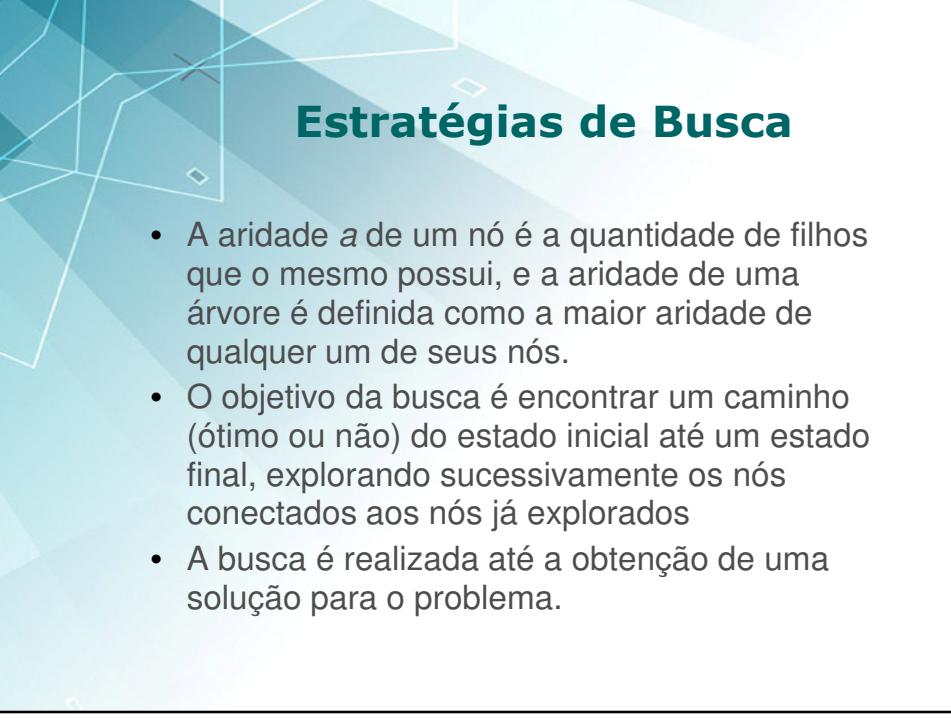
Máquinas de Estados Finitos (MEF)

- Aplicação em no jogo (*serious game*)
SilvesterMob



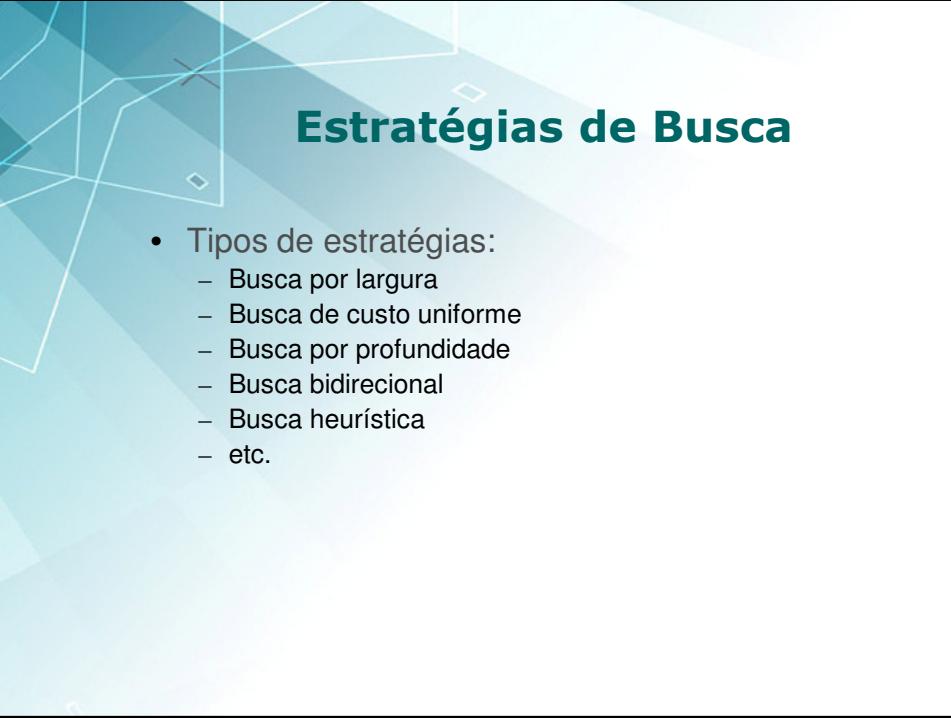
Estratégias de Busca

- Formalmente, o espaço de busca é constituído por nós n, conectados através de arcos.
- A cada arco pode ou não estar associado um valor, que corresponde ao custo c de transição de um nó a outro.
- A cada nó pode-se ter associado uma profundidade p, sendo que a mesma tem valor 0 no nó raiz e aumenta de uma unidade para um nó filho.



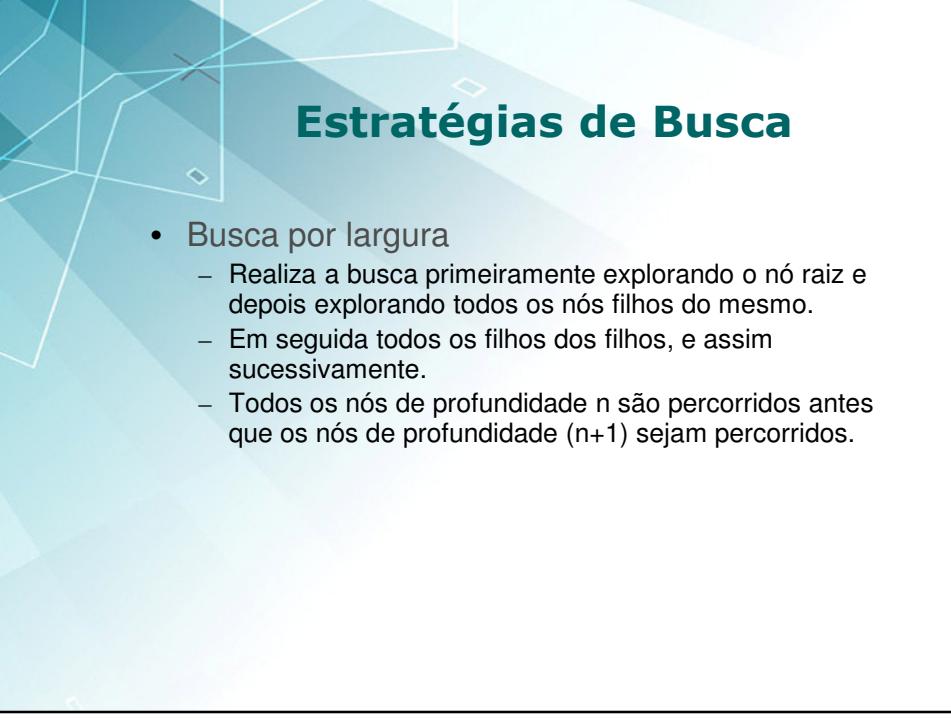
Estratégias de Busca

- A aridade a de um nó é a quantidade de filhos que o mesmo possui, e a aridade de uma árvore é definida como a maior aridade de qualquer um de seus nós.
- O objetivo da busca é encontrar um caminho (ótimo ou não) do estado inicial até um estado final, explorando sucessivamente os nós conectados aos nós já explorados
- A busca é realizada até a obtenção de uma solução para o problema.



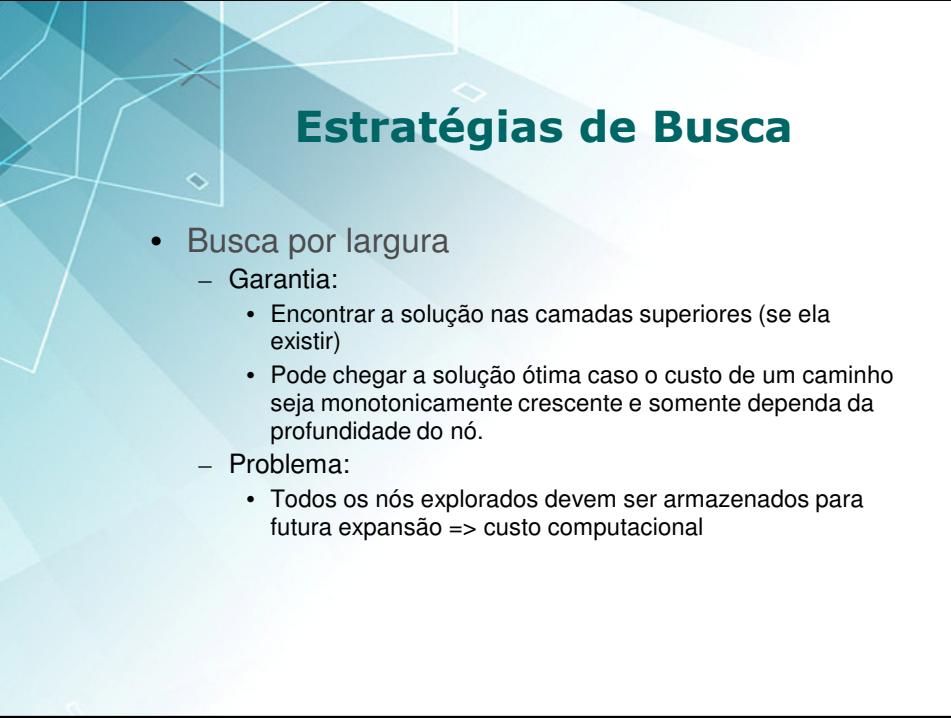
Estratégias de Busca

- Tipos de estratégias:
 - Busca por largura
 - Busca de custo uniforme
 - Busca por profundidade
 - Busca bidirecional
 - Busca heurística
 - etc.



Estratégias de Busca

- Busca por largura
 - Realiza a busca primeiramente explorando o nó raiz e depois explorando todos os nós filhos do mesmo.
 - Em seguida todos os filhos dos filhos, e assim sucessivamente.
 - Todos os nós de profundidade n são percorridos antes que os nós de profundidade $(n+1)$ sejam percorridos.



Estratégias de Busca

- Busca por largura
 - Garantia:
 - Encontrar a solução nas camadas superiores (se ela existir)
 - Pode chegar a solução ótima caso o custo de um caminho seja monotonicamente crescente e somente dependa da profundidade do nó.
 - Problema:
 - Todos os nós explorados devem ser armazenados para futura expansão => custo computacional

Estratégias de Busca

- Busca de custo uniforme
 - Atribui-se um custo a cada arco através de uma função $g(n)$
 - Explora-se sempre o nó de menor custo total acumulado até o momento.
 - Esta busca é similar à busca em largura, com profundidade igual ao último custo total calculado.

Estratégias de Busca

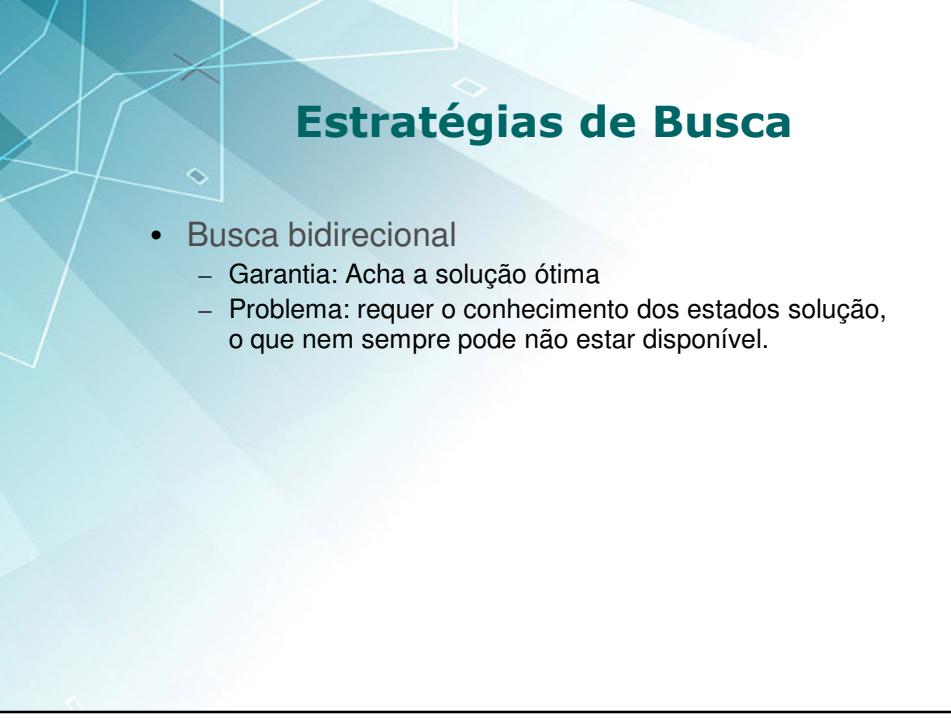
- Busca de custo uniforme
 - Garantia:
 - Sempre encontra uma solução.
 - Ela pode ser ótima se o custo de um caminho for monotonicamente crescente
 - Problema:
 - Os mesmos da busca em largura

Estratégias de Busca

- Busca por profundidade
 - Realiza a busca primeiramente explorando o nó de maior profundidade
 - consome menos memória que as anteriores, pois só o caminho corrente precisa ser armazenado => tende a ser mais rápida
 - Garantia:
 - Sempre encontra uma solução.
 - Não necessariamente encontra a solução ótima
 - Problema:
 - Pode ficar presa em ramos que não contenham a solução

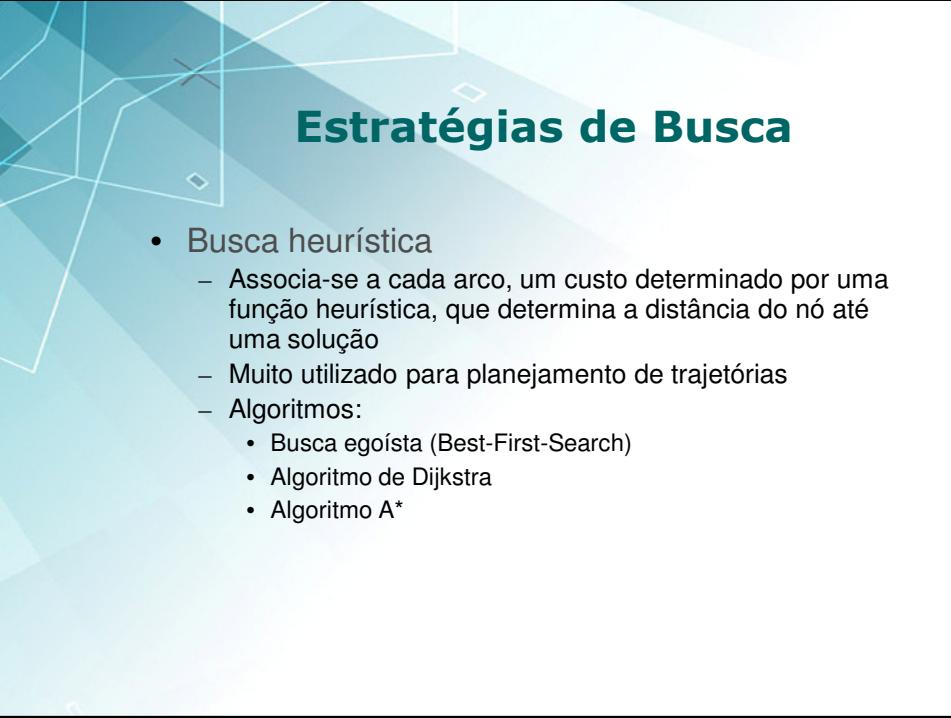
Estratégias de Busca

- Busca bidirecional
 - Realiza a busca de forma concorrente a partir do estado inicial e do estado solução.
 - Critério de parada: quando as duas buscas se encontram em um ponto intermediário.
 - Para cada uma das buscas, pode usar uma estratégia diferente.
 - Pode realizar tanto o caminho de ida (à partir do estado inicial) quanto de volta (partindo do(s) estado(s) final(is)).



Estratégias de Busca

- Busca bidirecional
 - Garantia: Acha a solução ótima
 - Problema: requer o conhecimento dos estados solução, o que nem sempre pode não estar disponível.



Estratégias de Busca

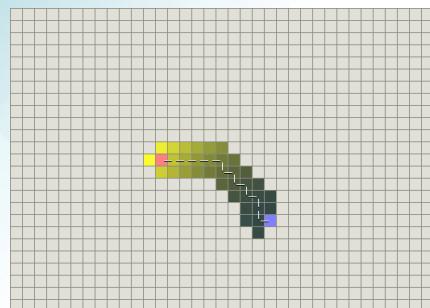
- Busca heurística
 - Associa-se a cada arco, um custo determinado por uma função heurística, que determina a distância do nó até uma solução
 - Muito utilizado para planejamento de trajetórias
 - Algoritmos:
 - Busca egoísta (Best-First-Search)
 - Algoritmo de Dijkstra
 - Algoritmo A*

Estratégias de Busca

- Busca heurística – Busca egoísta
 - Estratégia mais simples: realizar uma busca egoísta -> usar uma função heurística $h(n)$ e seguir o nó mais próximo do objetivo.
 - Problema: não se garante uma solução (quanto mais a ótima!)

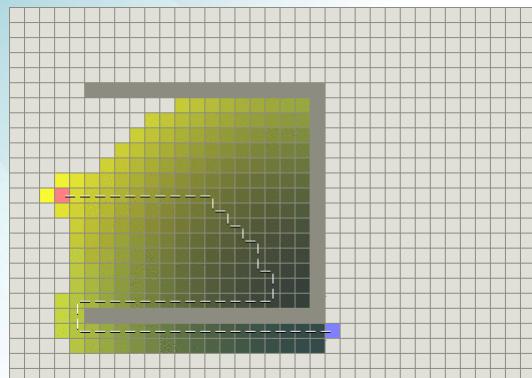
Estratégias de Busca

- Busca heurística – Busca egoísta
- Área que o algoritmo pesquisou (sem barreira):



Estratégias de Busca

- Busca heurística – Busca egoísta

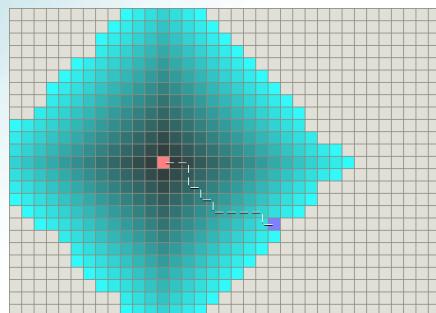


Estratégias de Busca

- Busca heurística - Algoritmo de Dijkstra
 - Acha o menor caminho entre dois nós de um grafo direcionado e com pesos
 - Encontra a solução ótima, desde que as extremidades não possuam custo negativo
 - Faz uma estimativa do custo mínimo, a partir da fonte e vai ajustando essa estimativa ao longo da execução
 - Um vértice só é considerado permanente quando o caminho de custo mínimo até ele foi encontrado
 - Problema: alta complexidade computacional

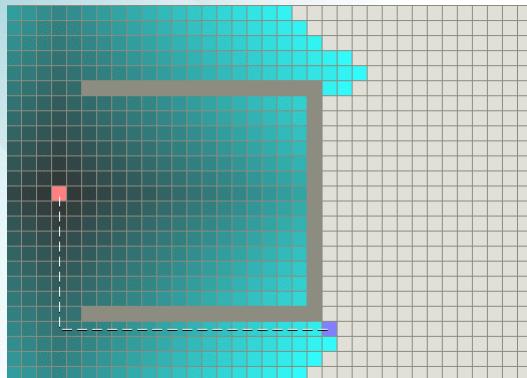
Estratégias de Busca

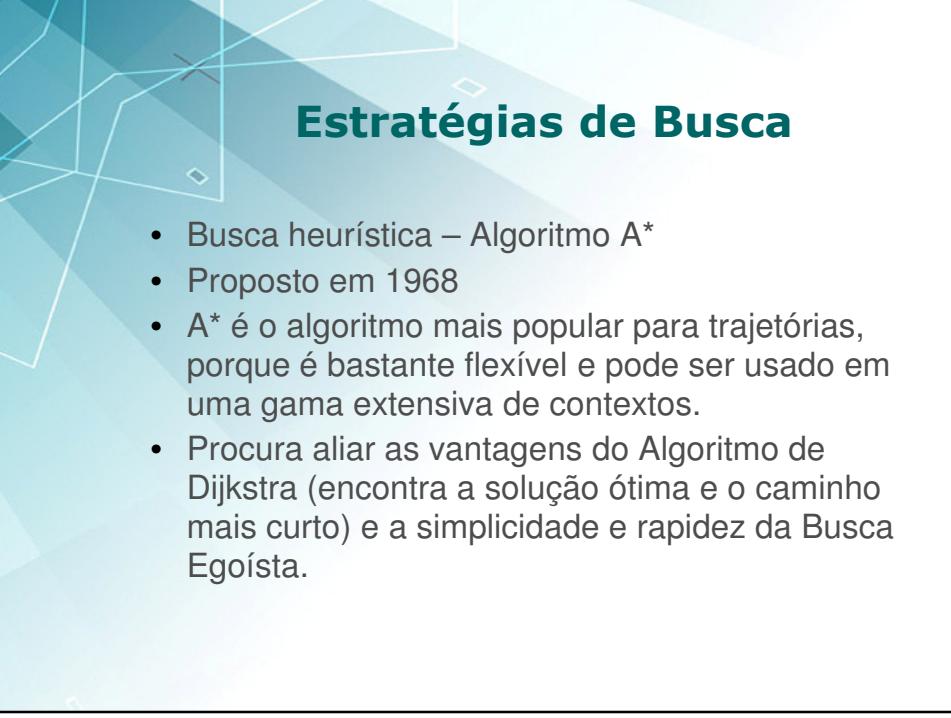
- Busca heurística - Algoritmo de Dijkstra
- Área que o algoritmo pesquisou (sem barreira):



Estratégias de Busca

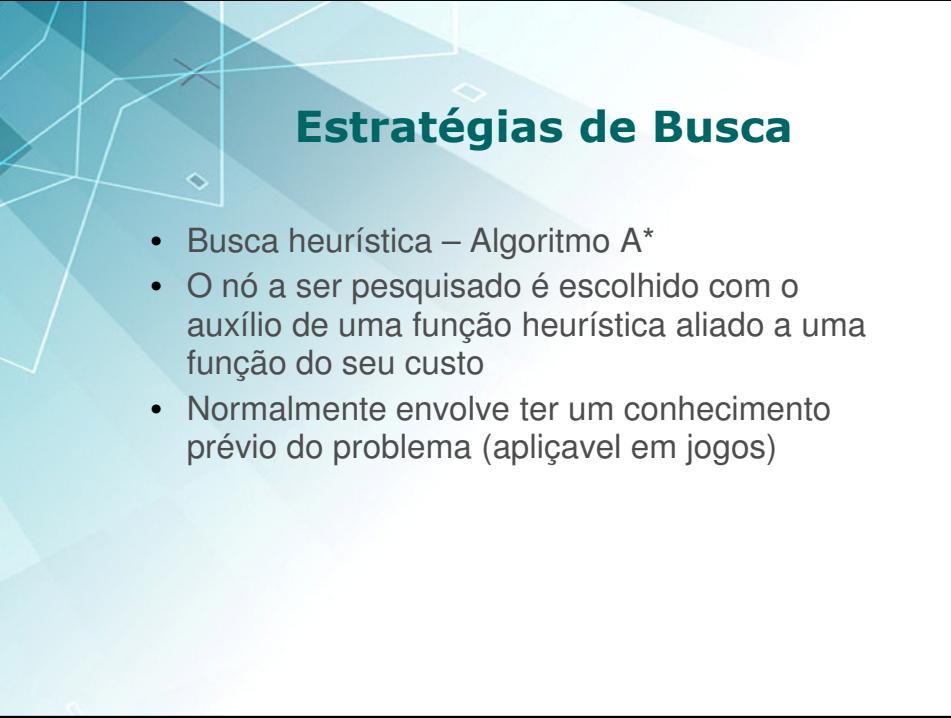
- Busca heurística - Algoritmo de Dijkstra





Estratégias de Busca

- Busca heurística – Algoritmo A*
- Proposto em 1968
- A* é o algoritmo mais popular para trajetórias, porque é bastante flexível e pode ser usado em uma gama extensiva de contextos.
- Procura aliar as vantagens do Algoritmo de Dijkstra (encontra a solução ótima e o caminho mais curto) e a simplicidade e rapidez da Busca Egoísta.



Estratégias de Busca

- Busca heurística – Algoritmo A*
- O nó a ser pesquisado é escolhido com o auxílio de uma função heurística aliado a uma função do seu custo
- Normalmente envolve ter um conhecimento prévio do problema (aplicável em jogos)

Estratégias de Busca

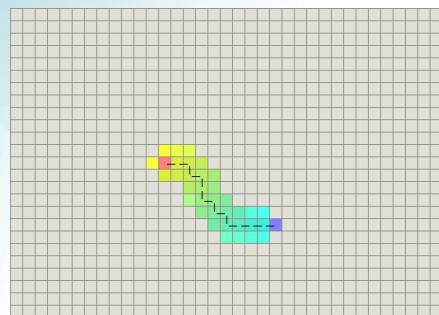
- Busca heurística – Algoritmo A*
- Segredo do sucesso:
 - combinar informações usadas pelo algoritmo de Dijkstra (favorecendo os vértices que estão perto do ponto de partida) e informações da Busca Egoísta (favorecendo vértices que estão perto da meta).
- Garante encontrar a solução ótima
- Problemas: complexidade é exponencial e deve armazenar todos os nós explorados

Estratégias de Busca

- Busca heurística – Algoritmo A*
 - Existem um grande número de variações para contornar esses problemas:
 - IDA* (1985) – computa o A* até um valor fixo da função custo + heurística. Este valor aumenta iterativamente.
 - SMA* (1992) - realiza a busca de forma limitada à memória disponível. Armazena os nós explorados até o limite da memória e os elimina quando precisa de espaço para a exploração de novos nós.

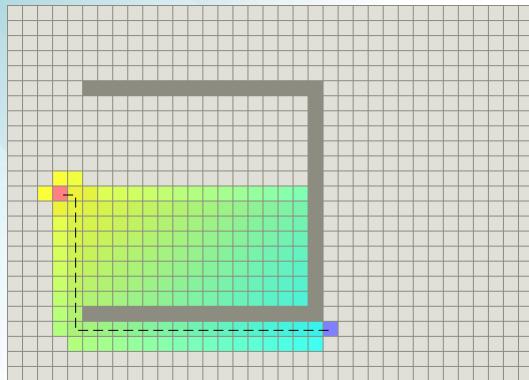
Estratégias de Busca

- Busca heurística – Algoritmo A*
- Área que o algoritmo pesquisou (sem barreira):



Estratégias de Busca

- Busca heurística – Algoritmo A*



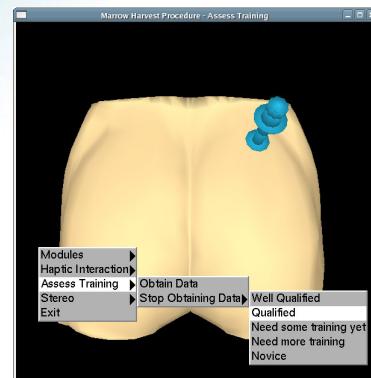
Estratégias de Busca

- Aplicações em jogos:
 - Controle de ações dos personagens:
 - Ataque/defesa
 - Simular ações com alguma inteligência
 - Encontrar uma trajetória entre localidades

Aplicações no Brasil

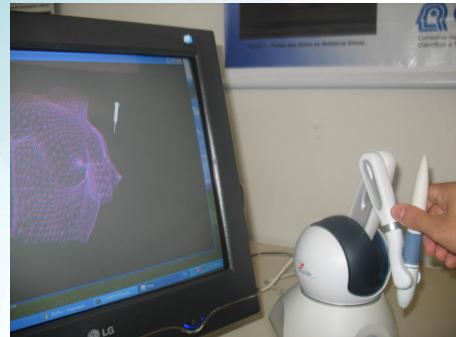
- Coleta de Medula Óssea

video CMO



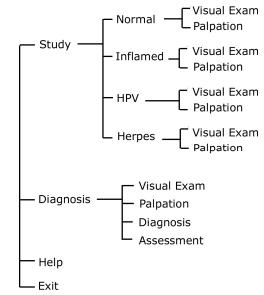
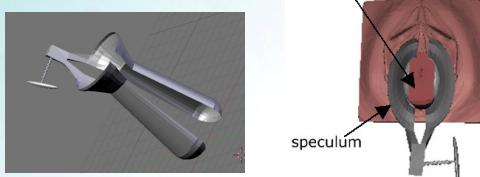
Aplicações no Brasil

- Punção de Mama



Aplicações no Brasil

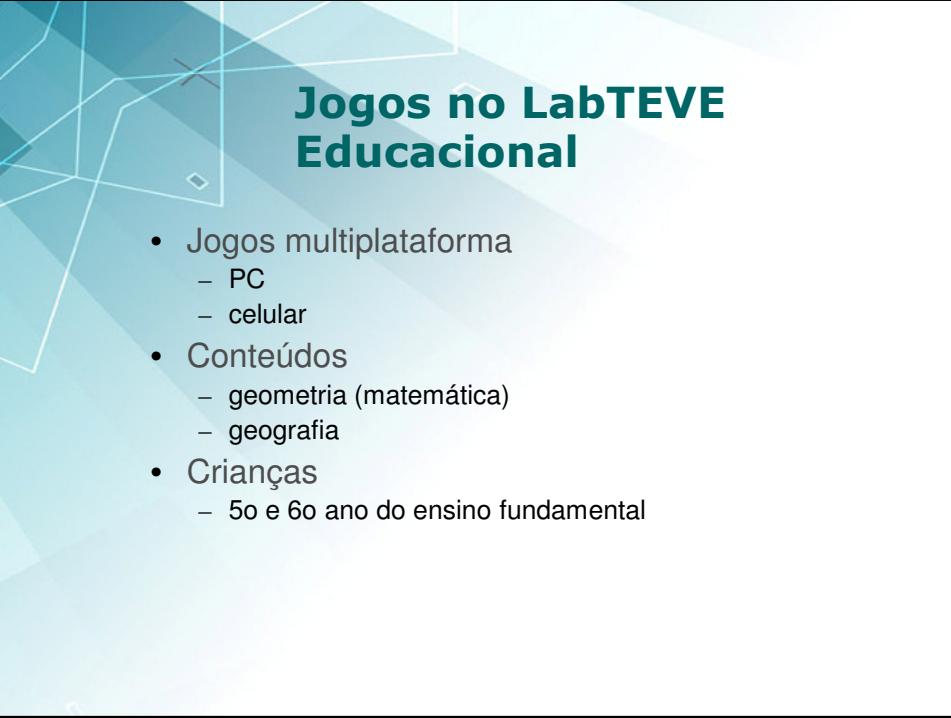
- Ensino de Cuidado Bucal
 - crianças
 - adultos
- Treinamento em Exame Ginecológico





Jogos no LabTEVE - UFPB

- Educacional
 - crianças – ensino fundamental
 - adultos – áreas de saúde
 - crianças – áreas de saúde
- Novas formas de aprendizado
- Novas formas de avaliação
- Novas formas de interação
- Desenvolvimento de ferramentas
- Multi e Interdisciplinaridade

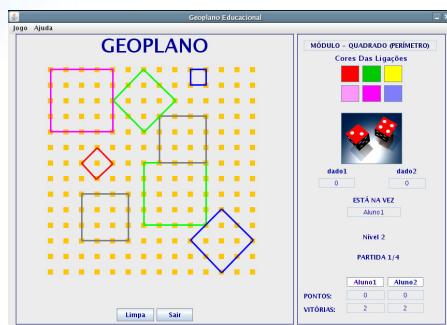
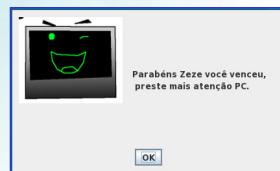


Jogos no LabTEVE Educacional

- Jogos multiplataforma
 - PC
 - celular
- Conteúdos
 - geometria (matemática)
 - geografia
- Crianças
 - 5º e 6º ano do ensino fundamental

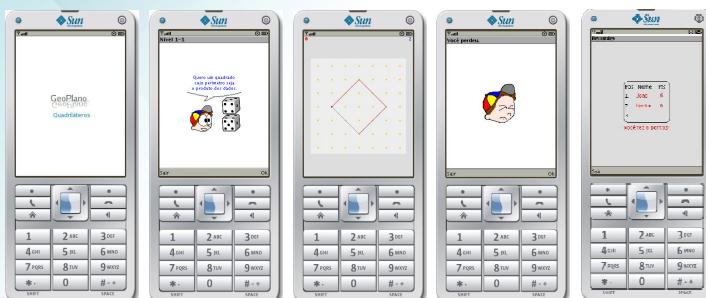
Jogos no LabTEVE Educacional

- GeoplanoPEC
 - Geometria plana
 - Inteligência no oponente
 - Sistema de partidas
 - Personagem
 - Interação



Jogos no LabTEVE Educacional

- GeoplanoMOB
 - geometria plana
 - interativo
 - abordagem adaptada para dispositivos móveis
 - desafio = tempo



Jogos no LabTEVE Educacional

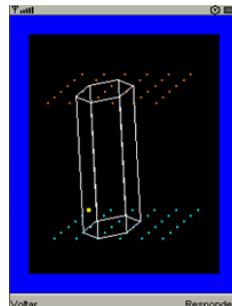
- GeoespaçoPEC
 - geometria espacial
 - RPG
 - enredo desafiador
 - estória
 - tempo



jogo

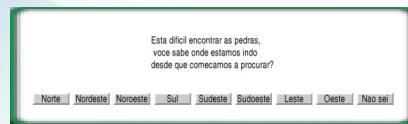
Jogos no LabTEVE Educacional

- GeoespaçoMOB
 - geometria espacial
 - RPG
 - explora recursos gráficos do *hardware*



Jogos no LabTEVE Educacional

- SilvesterPEC
 - geografia
 - RPG
 - enredo desafiador
 - ET em busca de elementos
 - inteligência



Jogos no LabTEVE Educacional

- SilvesterMOB
 - aventura em 2D
 - adaptação do jogo
 - mesmo conteúdo educacional
 - batalhas para conquistar itens



Jogos no LabTEVE Educacional

- Odontologia
- Ginecologia
- Cirurgia
 - novos dispositivos
 - desenvolvimento de ferramentas
 - técnicas de avaliação
 - treinamento



Mais informações

<http://www.de.ufpb.br/~labteve>

Obrigado!