

## Laboratório de Programação com Games

### Professor:

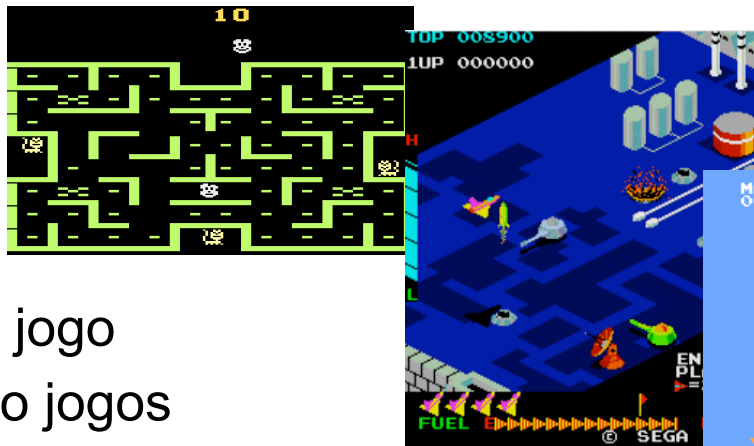
Anselmo Montenegro  
[www.ic.uff.br/~anselmo](http://www.ic.uff.br/~anselmo)

### Conteúdo:

- Conceitos sobre video jogos

# Roteiro

- O que é um jogo?
- O que é um vídeo jogo
- Histórico dos vídeo jogos
- Os diferentes tipos de vídeo jogos
- *Game Design Document*
- A arquitetura de um jogo
- O Game Loop



# O que é um jogo?

- “A **game** is a form of *interactive entertainment* where *players* must overcome *challenges*, by taking actions that are governed by *rules*, in order to meet a *victory condition*”

—Rollings & Adams



## O que é um vídeo jogo?

- Jogo eletrônico: é dispositivo que usa eletrônica para construir um sistema interativo com o qual o usuário possa jogar
- Um vídeo game é um tipo de jogo eletrônico que utiliza uma interface com o usuário para enviar resposta em display de vídeo

## Histórico de vídeo jogos



<https://www.youtube.com/watch?v=GoyGlyrYb9c>

## Tipos de vídeo jogos: categorização da Wikipedia

Action-adventure game  
Action role-playing games  
Adventure games  
Beat 'em up  
Cinematic platformer  
Computer  
Console role-playing video  
Fighting games  
Hack and slash  
Interactive movies  
Platform games  
Scrolling platformers  
Scrolling shooters  
Isometric platformer  
Isometric shooter

Light gun shooter  
Maze games  
Platform-adventure games  
Racing games  
Rail shooter  
Laserdisc video game  
Real-time strategy  
Run & gun shooters  
Rhythm game  
Stealth games  
Survival horror  
Vehicle simulation games  
Visual novels

## Tipos de vídeo jogos: action- adventure

- Tipo de jogo que combina elementos de vários generos incluindo:

- exploração
- quebra-cabeça
- uso de inventório de itens
- sistema monetário
- níveis de desenvolvimento
- e componentes de ação

- Outras características:

- ambiente aberto
- jogabilidade não linear

[Legend of Zelda](#) (1986)



## Tipos de vídeo jogos: Action role-playing games

- Jogo onde personagens possuem atributos estatísticos e voltado a exploração em um ambiente de combate de ação

- Exemplos:

- [Dragon Slayer II: Xanadu](#) (1985)
- [Zelda II: The Adventure of Link](#) (1987)





## Tipos de vídeo jogos: Adventure

- Jogos de aventura que narram uma estória em um ambiente a ser explorado:
  - Inicialmente possuíam interface puramente textual
  - Posteriormente surgiram versões com resposta gráfica mas não interativa

### Mistery House



## Tipos de vídeo jogos: Adventure

- Uma versão mais moderna utiliza interface point and click
- Exemplos:
  - [Zork](#) (1980) - [text adventure](#)
  - [Mystery House](#) (1980), de [Roberta Williams](#) para o [Apple II](#) - [graphic adventure game](#)
  - [King's Quest](#) (1984) da [Sierra](#),
  - [Maniac Mansion](#) (1987) [LucasArts](#), com interface [point and click](#)

Mystery House



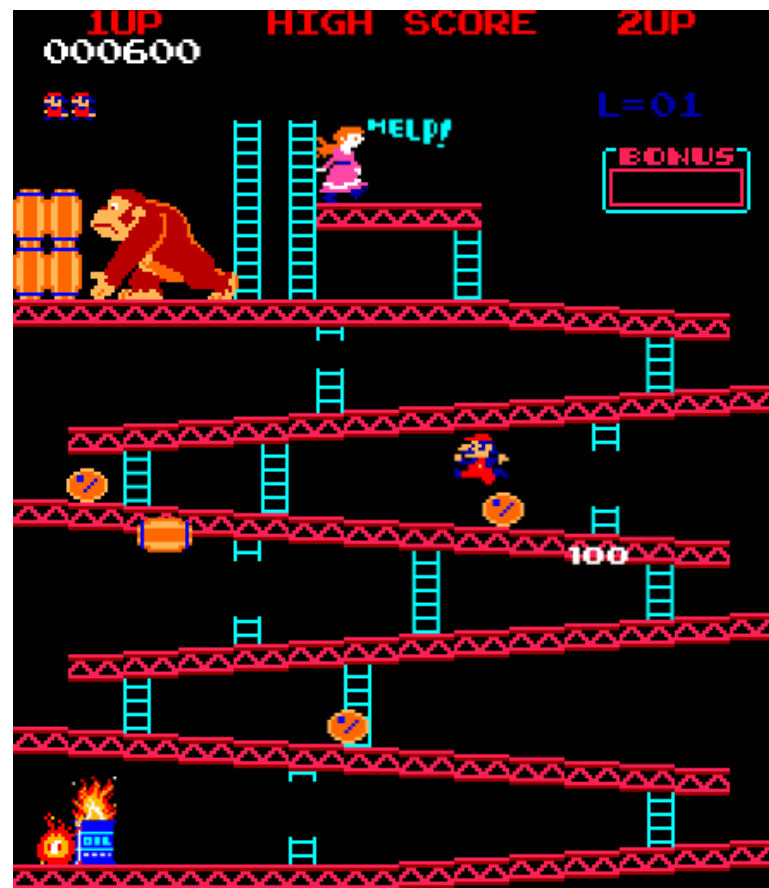
## Tipos de vídeo jogos: Beat' em up

- Jogo em side-scrolling envolvendo combate
- Exemplos:
  - [Karateka](#) (1984)
  - [Kung-Fu Master](#) (1984)
  - [Renegade](#), introduziu a movimentação tanto na horizontal quanto vertical
  - [Final Fight](#)
  - [Cadillac and Dinosaurs](#)



## Tipos de vídeo jogos: Platform

- Jogos centrados em subir escadas entre diferentes andares
- Exemplos:
  - Space Panic (1980)
  - [Donkey Kong](#) (1981), jogo que definiu o estilo com saltos sobre obstáculos e buracos
  - [Mario Bros.](#) (1983), modo cooperativo com dois jogadores



## Tipos de vídeo jogos: Cinematic Platformer

- *Prince of Persia* (1989): animação por rotação



## Tipos de vídeo jogos: Maze games

- Jogos que ocorrem em labirintos com paredes, armadilhas e perseguições
- Exemplos:
  - [Pac-Man](#) (1980)
  - [3D Monster Maze](#) (1981) primeiro jogo 3D para computador (ZX-81)
  - [Dungeons of Daggorath](#) (1982)



## Tipos de vídeo jogos: Maze games

- Jogos que ocorrem em labirintos com paredes, armadilhas e perseguições
- Exemplos:
  - [Pac-Man](#) (1980)
  - [3D Monster Maze](#) (1981) primeiro jogo 3D para computador (ZX-81)
  - [Dungeons of Daggorath](#) (1982)

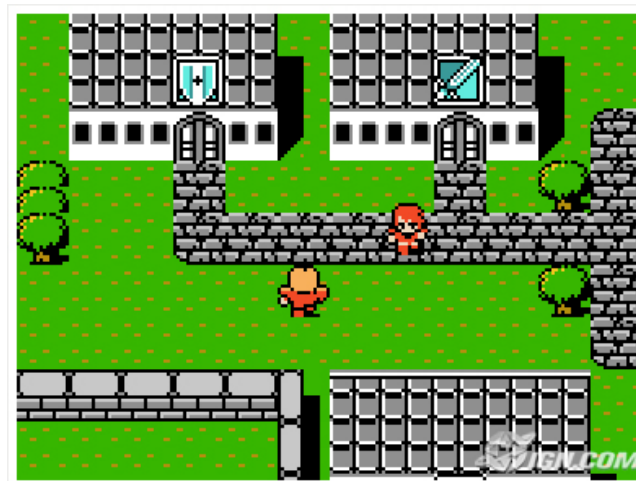


## Tipos de vídeo jogos: Role-playing

- Introdução de jogos de RPG em computadores e consoles. Ambientações típicas giram em torno de temas medievais e futuristas

- Exemplos:

- [Dragon Warrior](#) (1986)
- [Dragon Quest](#)
- [Phantasy Star](#) (1987)
- [Final Fantasy](#) (1987) – sistema em turno de batalha com vista lateral
- [Megami Tensei](#) (1987)





## Tipos de vídeo jogos: Scrolling Platformers

- Jogo de plataforma que rola continuamente na tela. Ações típicas são saltar e desviar de obstáculos, inimigos
- Exemplos:
  - [Jump Bug](#) (1981),
  - [Jungle King](#) (1982)
  - [Pac-Land](#) (1984), multi-layered [parallax scrolling](#)
  - [Super Mario Bros.](#) (1985)
  - [Wonder Boy](#) (1986)



## Tipos de vídeo jogos: Scrolling Platformers

- Jogo de plataforma que rola continuamente na tela. Ações típicas são saltar e desviar de obstáculos, inimigos
- Exemplos:
  - [Jump Bug](#) (1981),
  - [Jungle King](#) (1982)
  - [Pac-Land](#) (1984), multi-layered parallax scrolling
  - [Super Mario Bros.](#) (1985)
  - [Wonder Boy](#) (1986).



[http://img.gamefaqs.net/screens/6/c/f/gfs\\_22701\\_2\\_2.jpg](http://img.gamefaqs.net/screens/6/c/f/gfs_22701_2_2.jpg)

## Tipos de vídeo jogos: Scrolling shooter

- Jogos de tiro em que a tela rola continuamente sendo que diferentes níveis surgem horizontalmente ou verticalmente com o prosseguimento do jogo
  - Defender (1980)
  - [Vanguard](#) (1981) movimentos verticais e horizontais
  - [Xevious](#) (1982) scroller vertical
  - [Moon Patrol](#) (1982) introduziu a técnica de [parallax scrolling](#)
  - [Gradius](#) (1985) controle de armas; memorização de fases
  - [Thrust](#) (1986) cavernas, gravidade e inércia



## Tipos de vídeo jogos: Scrolling shooter

- Jogos de tiro em que a tela rola continuamente sendo que diferentes níveis surgem horizontalmente ou verticalmente com o prosseguimento do jogo
  - Defender (1980)
  - [Vanguard](#) (1981) movimentos verticais e horizontais
  - [Xevious](#) (1982) scroller vertical
  - [Moon Patrol](#) (1982) introduziu a técnica de [parallax scrolling](#)
  - [Gradius](#) (1985) controle de armas; memorização de fases
  - [Thrust](#) (1986) cavernas, gravidade e inércia



## Game Design Document

- O Game Design Document (GDD) é o documento que descreve conceitualmente o jogo e como ele será desenvolvido ao longo do projeto

# Game Design Document

- Um GDD deve conter:
  1. Nome do Projeto
  2. High concept - Descreva em apenas um parágrafo as principais características do jogo, incluindo os aspectos diferenciais e inovadores do projeto.
  3. Gameplay e Enredo - Descreva todos os elementos de jogo e sua interação com o jogador. Fluxo de jogo, fases, personagens, enredo, storyboards e cenário, são todos elementos desta seção. Se necessário utilize imagens, tabelas e fluxogramas.

# Game Design Document

4. Interface de usuário - Mostre com imagens e diagramas como os elementos visuais são dispostos para o usuário. Inclua exemplos de interfaces gráficas e imagens do jogo.
5. Audio e música - Descreva as características de áudio e música do jogo.
6. Arte Conceito e referências - Nesta seção, inclua arte conceito ou referências de outros jogos, que ilustram algum aspecto específico ou amplo deste jogo.
7. Ideias adicionais e observações - Inclua aqui funcionalidades extras que podem ou não ser incluídas no jogo, ou outras observações quanto ao projeto.

# Arquitetura de Jogos

- Primeiro vejamos o que é uma arquitetura de software:

“A arquitetura de um software é uma descrição da **estrutura** do sistema (a forma como está organizado), os seus componentes, as **propriedades visíveis externamente** destes componentes e suas **relações**”

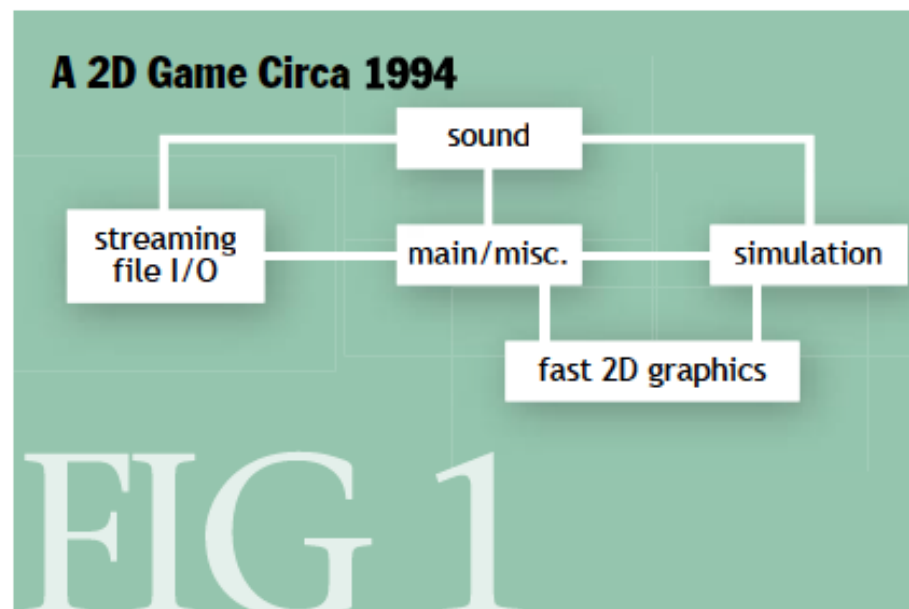


# Arquitetura de Jogos

- Estrutura – organização dos elementos de um programa ou sistema computacional
- Visibilidade Externa – abstração dos detalhes que são relevantes para o estabelecimento de restrições e inter-relações entre os elementos
- Relações – papéis dos elementos que compõem a arquitetura e como eles se relacionam entre si

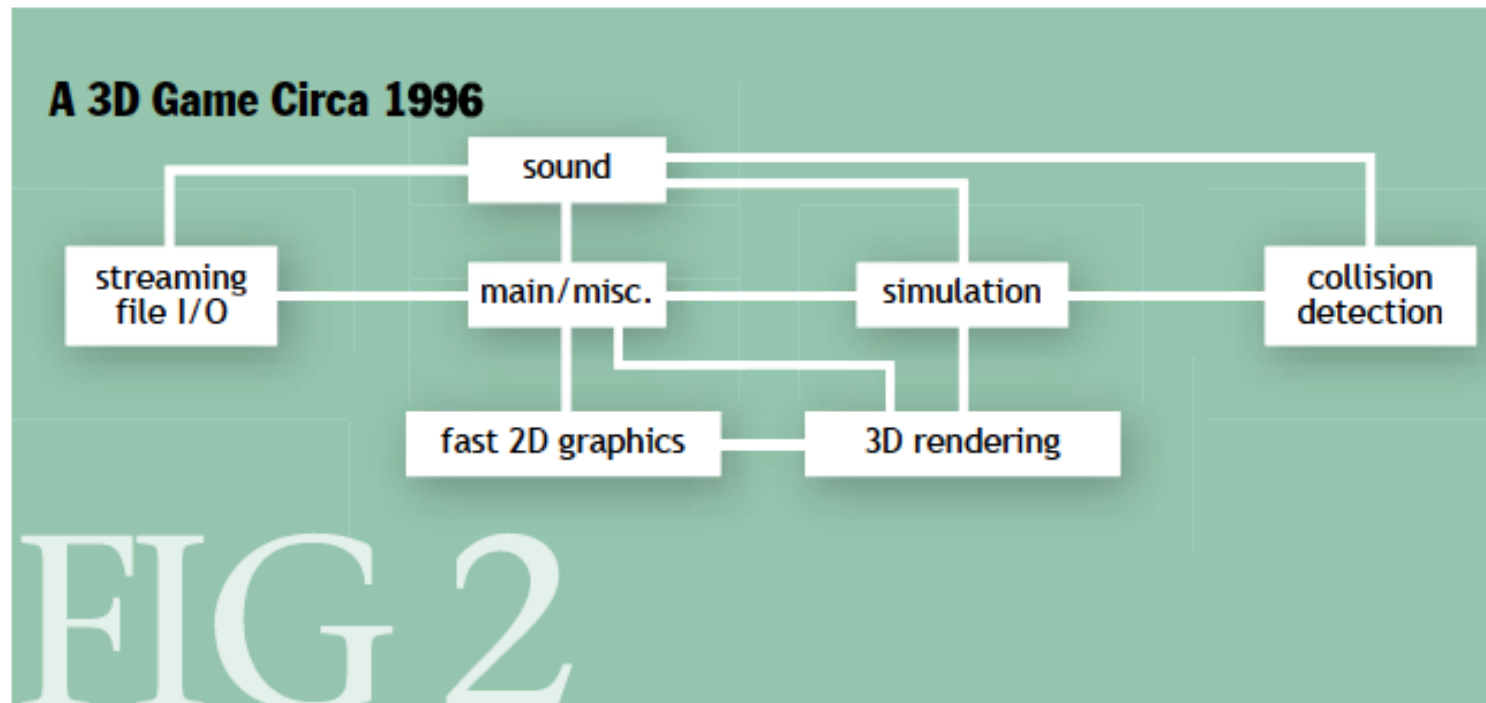
## Arquiteturas de jogos

- Os primeiros sistemas de jogos eram bem simples com poucos e limitados componentes de software
- Entretanto algumas decisões arquiteturais já permitiam identificar alguns submódulos e suas responsabilidades



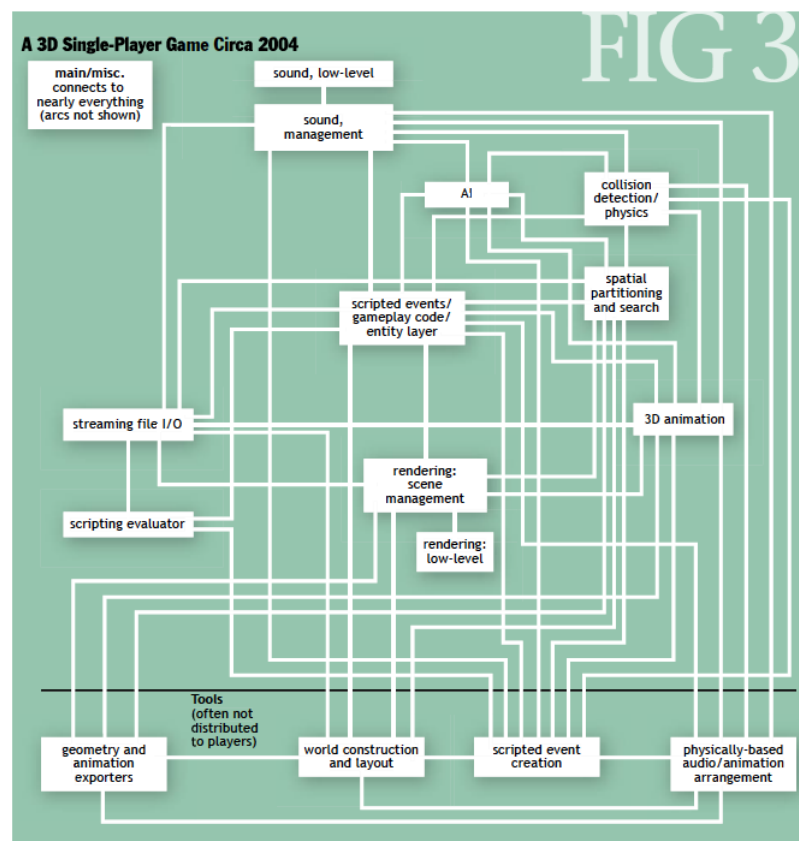
## Exemplos de arquiteturas de jogos

- Na medida em que gráficos 3D se tornaram importantes, componentes responsáveis por tal funcionalidade passaram a se revelar na arquitetura



## Exemplos de arquiteturas de jogos

- Um jogo 3D, por volta de 2004, alcançou um grau de complexidade tão alto que torna-se inconcebível projetar um software sem considerar a sua arquitetura



## Game Engines

- Game Engine: é um conjunto de componentes de software reutilizáveis e ferramentas para desenvolvimento de vídeo jogos
- Criam uma abstração dos detalhes presentes nas atividades e processos comuns a maioria dos jogos

## Game Engines

- Qual a diferença entre um Game Engine e um jogo (game)?
  - Um Game Engine possui uma arquitetura voltada a dados
  - Um jogo possui lógica e regras do jogo codificadas rigidamente no código

# Game Engines

- Componentes:
  - [Rendering engine](#)
  - [Physics engine](#)
  - Detecção e tratamento de colisão
  - Audio
  - [Scripting](#)
  - Animação
  - Inteligência artificial
  - Rede
  - Streaming
  - Gerenciamento de memória,
  - Threading
  - Suporte a localização
  - Grafo de cena

# Game Engines

Nome	Linguagem	Scripting	Cross-Platform	2D/3D	Licença	Obs
Blender	C++	Python	Sim	2D/3D	GLP	
idTech 4	C++	C++ via dll	Sim	3D	GLP	Engine do Doom 3
JMonkeyEngine	Java		Sim	3D	GLP	
Ogre3D	C++	C++	Sim	3D	MIT	
Unreal Engine 3		UnrealScript			Proprietaria	
Crytek 3					Livre até a versão 3.5.8	
Unity		Javascript, C#	Sim	2D/3D	Livre, Pro é proprietária	



## Game Engines

- É praticamente impossível reutilizar o software de um jogo para criar um jogo distinto
- Por outro lado, um Game Engine é um software extensível que pode servir utilizado para criar diferentes jogos sem muitas modificações no seu sistema

## Game Engines

- Neste curso não desenvolveremos Game Engines
- Tão pouco usaremos um Game Engine
- Que ferramentas usaremos então?
  - Linguagem de programação
  - API para desenvolvimento de jogos: PPlay

## Game Engines

- O que é uma API (Application Programming Interface)?

## Game Objects

- Em algumas arquiteturas e engines de jogos, todos os objetos ou entidades do jogo são tratadas segundo o conceito de **Game Object**
- Um Game Object é um container que armazena diferentes **componentes** que caracterizam uma entidade específica no jogo
- Exemplos de componentes são:
  - Geometria
  - Navegação
  - Transformações
  - Física
  - Audio
  - Rendering
  - Scripts

## Game Objects

- Componentes são os ingredientes que são usados para compor um objeto
- Um Game Object é o recipiente onde os componentes são combinados
- Um personagem, por exemplo, é criado a partir de um game object vazio combinando diferentes componentes como:
  - a forma do personagem,
  - seu esquema de navegação,
  - sons
  - sua resposta a colisões,
  - scripts que determinam seu comportamento e etc...

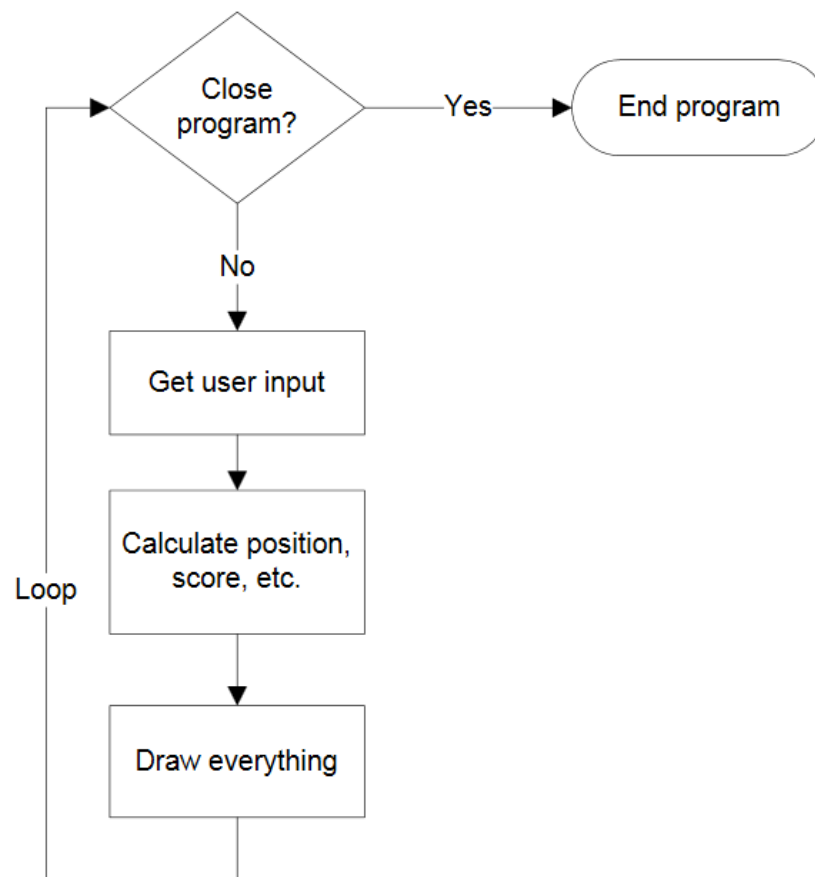
## O Game Loop

- O Game Loop é uma estrutura que descreve um conjunto de processos que se repetem indefinidamente a cada frame que é exibido
- Isto ocorre em média de 30 a 60 vezes por segundo um ciclo se repete

## O Game Loop

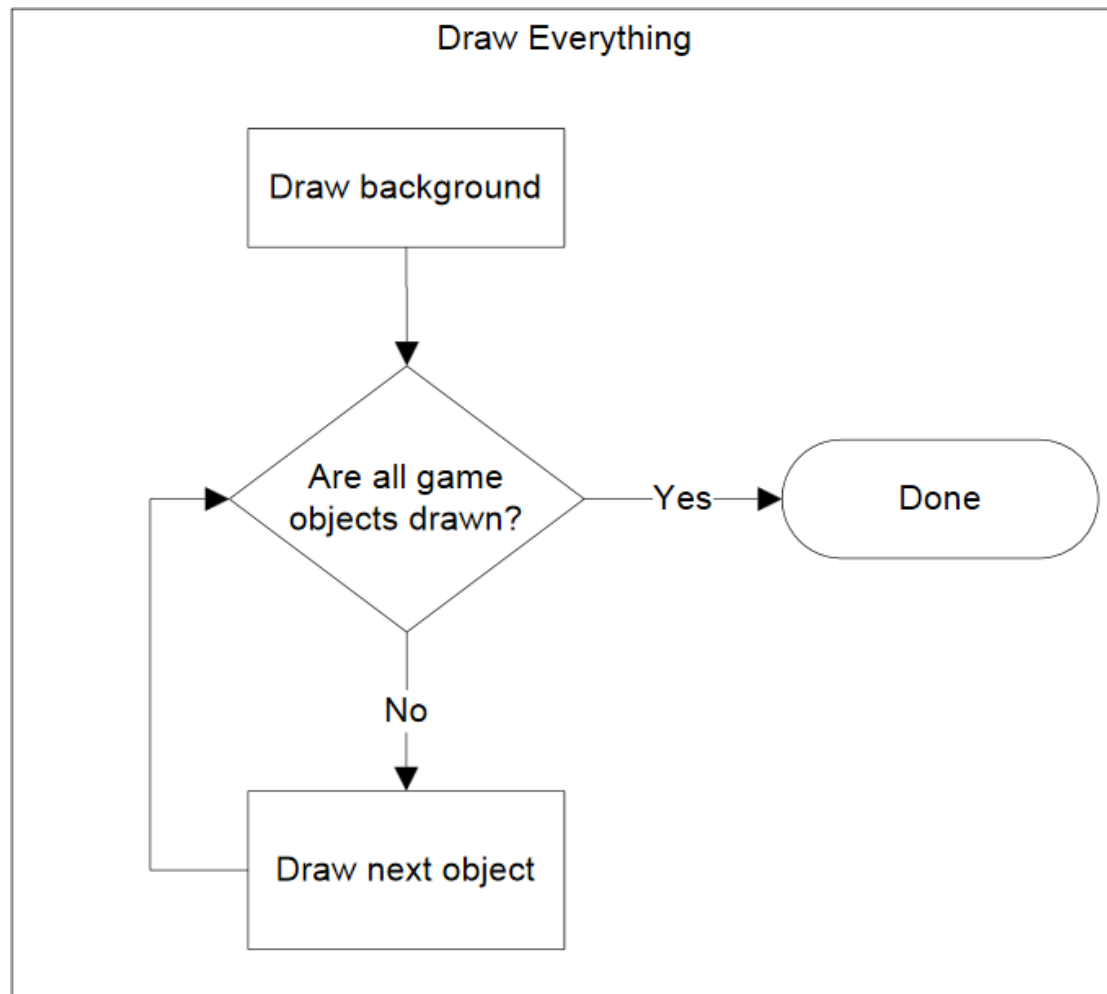
- Em um game loop são executados os seguintes processos de um jogo:
  1. Captura e tratamento de entrada de dados(input)
  2. Atualização da cena (update)
    - a) IA
    - b) Simulação Física
    - c) Atualização dos game objects
    - d) Atualização dos estados do jogo
    - e) Atualização do passo de tempo
  3. Rendering (desenho da cena)

# O Game Loop





# O Game Loop



## Referências

- Geral
  - Steve Rabin. *Introduction to Game Development*. Second Edition. Cengage Learning.
- Histórico
  - [http://wirelesslab.sjtu.edu.cn/resource/seminar/080411/History\\_of\\_Computer\\_and\\_Video\\_Games.ppt](http://wirelesslab.sjtu.edu.cn/resource/seminar/080411/History_of_Computer_and_Video_Games.ppt)
  - [http://inventors.about.com/library/inventors/blcomputer\\_videogames.htm](http://inventors.about.com/library/inventors/blcomputer_videogames.htm)
- Gêneros de Jogos
  - [http://en.wikipedia.org/wiki/Video\\_game\\_genres](http://en.wikipedia.org/wiki/Video_game_genres)

## Referências

- Arquitetura de jogos
  - Software Architecture and the Creative Process in Game Development. Njal Nordmark. Master of Science in Computer Science. June 2012. Supervisor: Alf Inge Wang, IDI. Norwegian University of Science and Technology. Department of Computer and Information Science.
- Game Engine
  - [http://www.gamecareerguide.com/features/529/what\\_is\\_a\\_game\\_.php](http://www.gamecareerguide.com/features/529/what_is_a_game_.php)

## Referências

- Game Objects
  - <http://docs.unity3d.com/Manual/GameObjects.html>
- Game Loop
  - [http://www.programarcadegames.com/index.php?chapter=loops&lang=pt#section\\_4](http://www.programarcadegames.com/index.php?chapter=loops&lang=pt#section_4)