



INSTITUTO FEDERAL
SANTA CATARINA

Construindo um Jogo para a Web – *River Raid*

Programação para a Internet

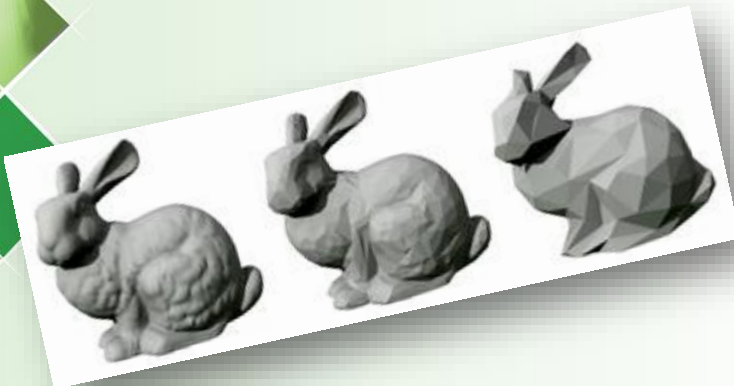
Prof. Vilson Heck Junior

Tecnologias Necessárias

- Tecnologias já Estudadas:
 - HTML;
 - CSS;
 - JavaScript;
- **Tecnologias Novas:**
 - **Computação Gráfica Básica;**
 - **Noções de Geometria;**
 - **Noções de Física;**
 - **Reprodução de Sons;**
 - **Enredo;**

Computação Gráfica

- É um campo da Ciência da Computação que estuda métodos para sintetizar e manipular digitalmente conteúdo visual:
 - Geração de imagens 2D;
 - Geração de imagens 3D (renderização);
 - Com ou sem animação;



Noções de Geometria

- Gráficos 2D ou 3D são na verdade a composição de pequenas peças geométricas:
- A relação espacial dada entre diferentes objetos existentes em uma cena deve ser respeitada:
 - Dois corpos não podem ocupar um mesmo lugar no espaço!

Noções de Física

- Objetos podem possuir algum tipo de movimento ou interação com outros objetos;
- Para isto, geralmente respeitam alguma(s) regras físicas:
 - Próximas a real: Simulação;
 - Diferenciadas: Arcade;

Reprodução de Sons

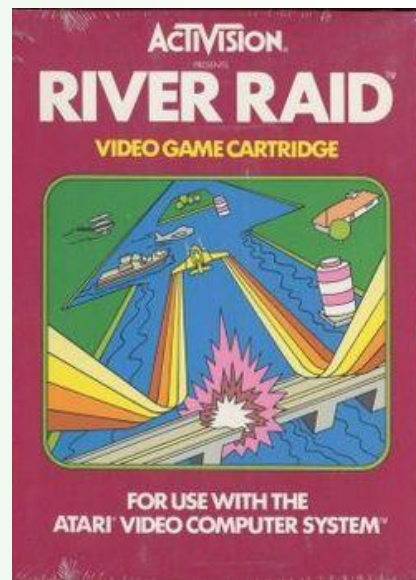
- O som é o elemento responsável por estimular o sentido da audição;
- Não tanto quanto os gráficos, mas os sons são responsáveis por completar uma boa sensação de imersão em jogos e entretenimento;
- Geralmente os sons (músicas ou barulhos) serão escolhidos conforme um determinado contexto ou acontecimento.

Enredo

- O enredo irá explicar ao usuário o que deverá ser feito e deve ser o principal responsável por atrair a atenção do jogador:
 - História;
 - Diversão;
 - Desafios;
 - Passatempo;
 - ...

Enredo

- River Raid:
 - Lançado em 1982;
 - Plataforma: Atari 2600;
 - Desenvolvido por Carol Shaw;
 - Publicado pela Activision;
 - Mais de 1 milhão de cartuchos vendidos;
 - Portado para outros consoles;



Enredo

- River Raid:
 - Jogo de rolagem;
 - Avião avança ao longo do curso de um rio;
 - Destruindo inimigos;
 - Mantendo o tanque com combustível.

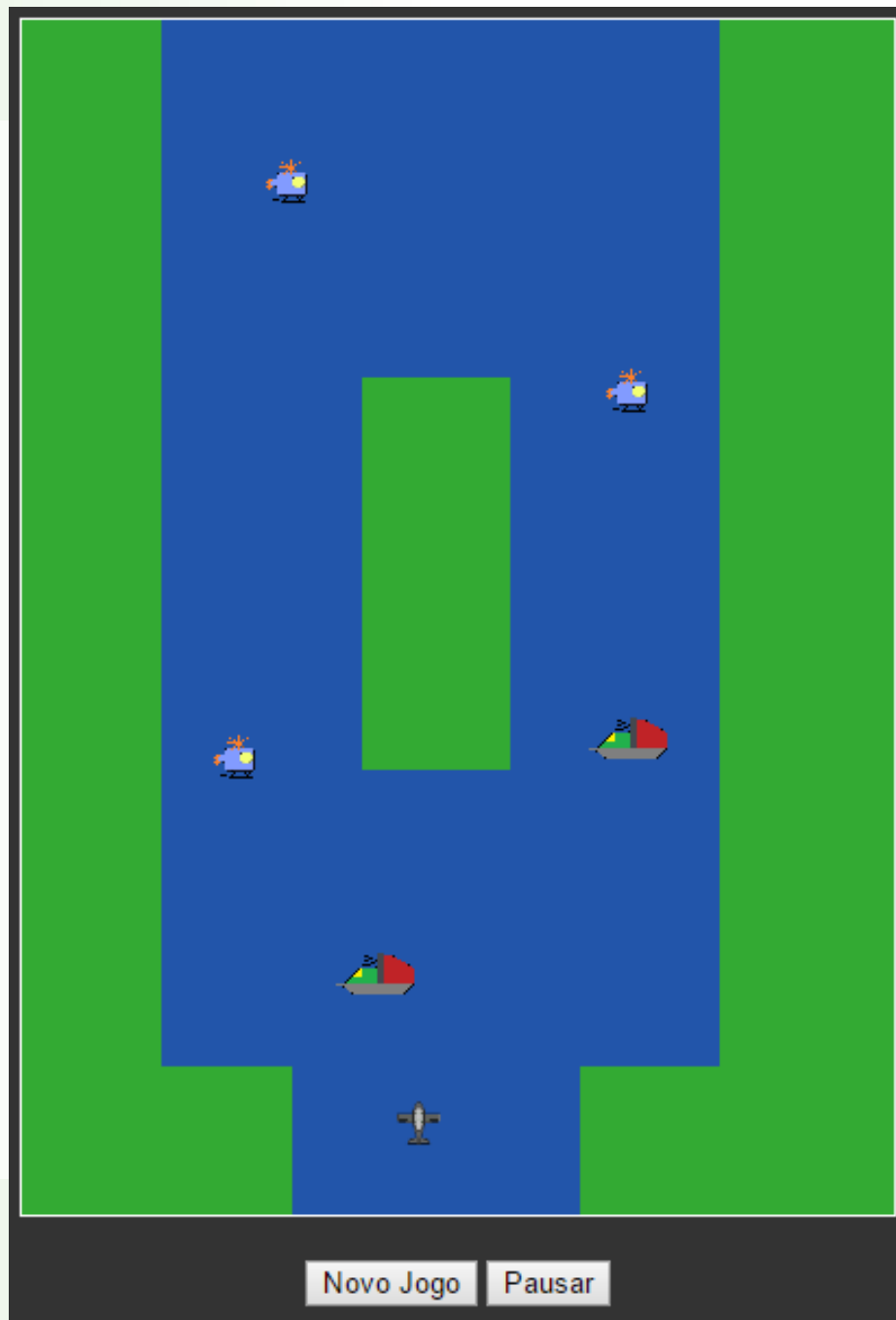




INSTITUTO FEDERAL
SANTA CATARINA



Nosso Conceito



105
ANOS

REDE FEDERAL
DE EDUCAÇÃO
PROFISSIONAL
E TECNOLÓGICA
1909-2014



INSTITUTO FEDERAL
SANTA CATARINA



River Raid

LISTA DE RECURSOS INICIAIS



Recursos Iniciais

- Pasta: “River Raid”:
 - index.html
 - Construiremos de um documento web, inserindo todos os demais elementos necessários;
 - css/estilo.css
 - Definiremos algumas configurações de cores, bordas e outros para nossa interface;
 - **js/** RiverRaid.js & Jogador.js & Cenario.js & Inimigo.js & Tiro.js
 - Faremos todo o processamento e configurações do jogo, dando ações aos acontecimentos e eventos do jogo.
 - js/mapa01.js & js/mapa02.js
 - Representam as variações de mapas existentes em nosso jogo. Serão gerados por um aplicativo gerados de mapas.

index.html

- Crie o arquivo como **doctype** para **html 5**;
- Crie as tags para:
 - **<html>**, **<head>**, **<body>** e **<title>**;
- Estipule um **<link>** com arquivo de estilo (css/estilo.css);
- Adicione os arquivos de **<script>** dentro do **<head>**:
 - js/mapa01.js
 - js/mapa02.js
 - js/Tiro.js
 - js/Jogador.js
 - js/Inimigo.js
 - js/Cenario.js
- Adicione o arquivo de **script** RiverRaid.js **ao fim do <body>**:
 - ***Importante: adicionar os outros arquivos js antes, pois o último precisa dos primeiros já executados.***

index.html

- Adicione os seguintes Tags com seus atributos dentro do <body>:
 - <canvas>Navegador não suportado!</canvas>
 - id = “tela” width=400 height=550
 - <button>Novo Jogo</button>
 - onclick=“RiverRaid.novoJogo(‘tela’)” id=“btnNovo”
 - <button>Pausar</button>
 - onclick=“RiverRaid.pausar()” id=“btnPausa”
 - <p id=“tecla”></p>
 - <p>A,S,D,W: Mover
G: Atirar</p>



estilo.css

```
body { background-color: #333333;
        text-align: center; }

#tela {
    background-color: #2255AA;
    border: 1px solid #FFFFFF; }

button {
    background-color: #EE9999;
    color: white;
    border: 1px solid red;
    width: 100px;
    height: 30px;
    font-weight: bold;
    cursor: pointer; }

button:hover { background-color: #EEBBBB; }

button:disabled {
    background-color: #DDDDDD;
    cursor: not-allowed; }
```



INSTITUTO FEDERAL
SANTA CATARINA



River Raid

DESENHANDO NO CANVAS



105
ANOS

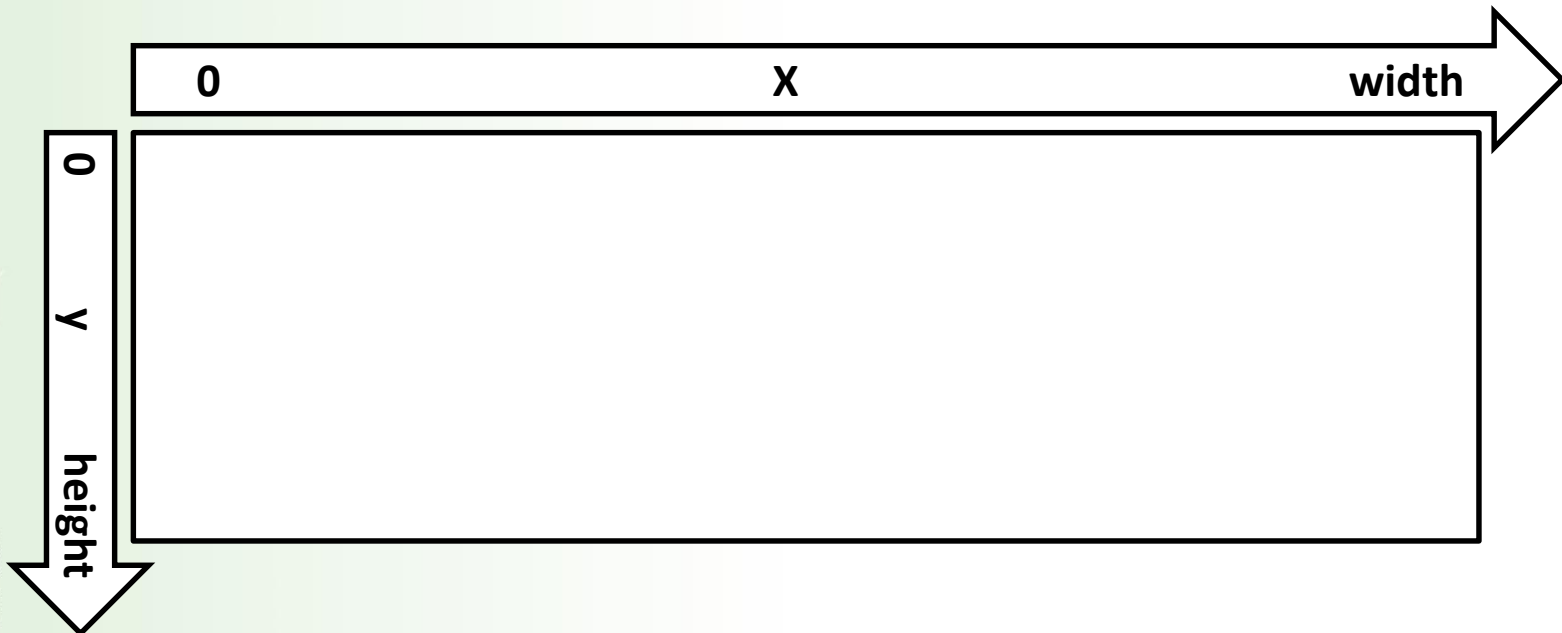
REDE FEDERAL
DE EDUCAÇÃO
PROFISSIONAL
E TECNOLÓGICA
1909-2014

<canvas>

- Canvas é um termo inglês dado a alguns tipos de tela para pintura;
- No nosso caso, será uma área dentro do documento HTML onde poderemos pintar o que precisarmos;
- Nosso pincel e paleta de cores estão disponíveis através de código JavaScript.

<canvas>

- O Canvas é feito para oferecer suporte a rápido desenho de cenas bidimensionais ou tridimensionais:
 - Geralmente acelerado por Hardware;



js/RiverRaid.js

//Recuperando referência dos objetos no documento

var canvas = document.getElementById("tela");

var ctx = canvas.getContext("2d");

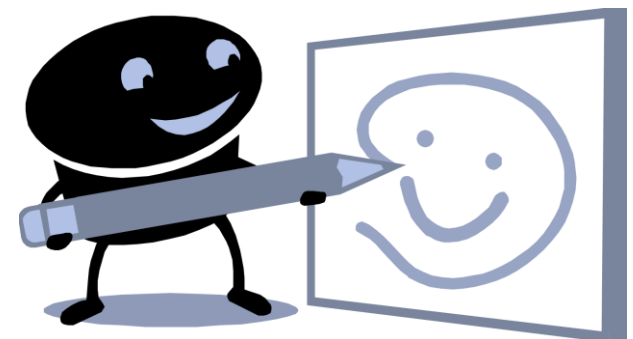
//Um pequeno teste (remover depois de testar)

ctx.fillStyle = "#FF0000"; //Usar cor vermelha

ctx.fillRect(20, 30, 50, 100); //x=20, y=30, w=50 e h=100

Desenhando

- Temos uma tela para desenho;
- Conhecemos uma primeira ferramenta para pintar algo;
- Temos que utilizar esta ferramenta de forma a construir o cenário inicial do nosso jogo;

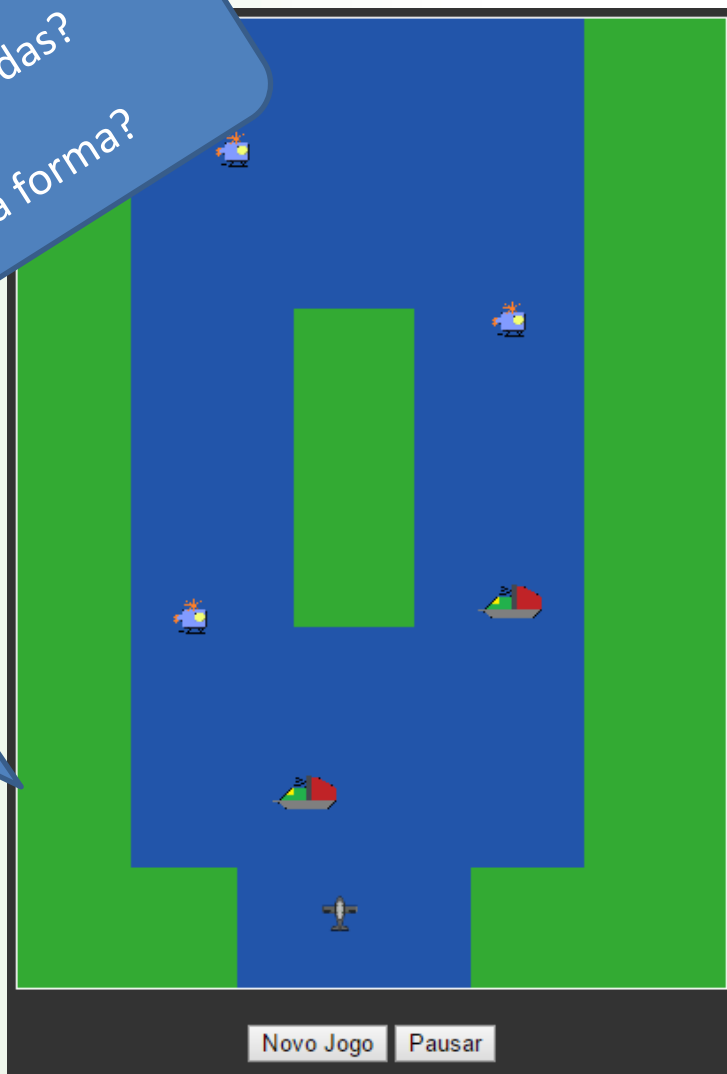




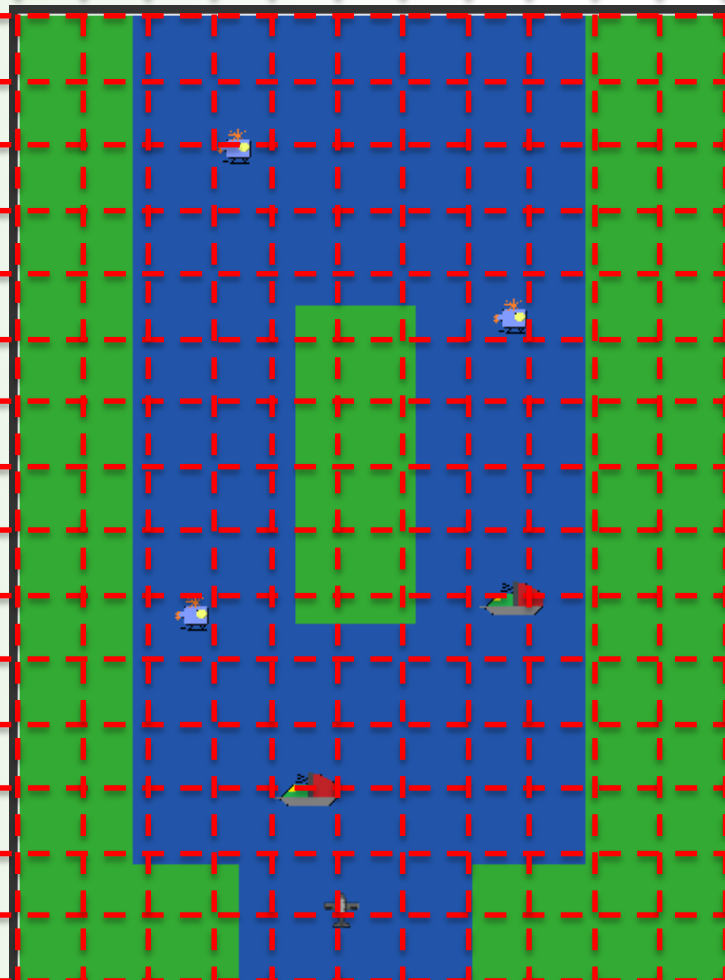
INSTITUTO FEDERAL
SANTA CATARINA

Relembrando

- Quais são as formas geométricas utilizadas?
- Qual é a quantidade de cada forma?
- Qual é a posição e tamanho de cada forma?



Estrutura de Dados: matriz



Posições da matriz

Descrição	Representação Gráfica	Código (cor)
Água		0 - (preto)
Terra/ grama		1 - (branco)
Início	N/A	2 - (vermelho)
Inimigo parado	 ou 	3 - (verde)
Inimigo móvel	N/A	4 - (azul)
Posto	N/A	5 - (cinza)

Matriz de Dados

0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

Matriz de Dados

Posição = 0

0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

linhasPorTela = 5

Sentido de avanço

Matriz de Dados

Posição = 1

0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

linhasPorTela = 5

Sentido de avanço

Matriz de Dados

Posição = 8

0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

linhasPorTela = 5

Sentido de avanço

Matriz de Dados

Posição = 9

5					
6					
7					
8					
9					
10					
11					
12					

linhasPorTela = 5

Sentido de avanço

Matriz de Dados

Posição = 9

linhasPorTela = 5

5					
6					
7					
8					
9					
10					
11					
12					
0					
1					
2					
3					
4					

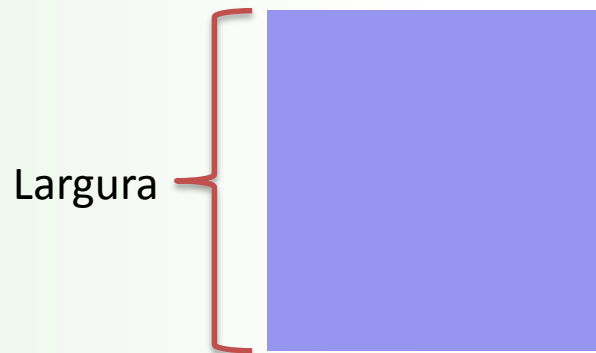
mapa[0]

mapa[1]

Sentido de avanço

Dimensões

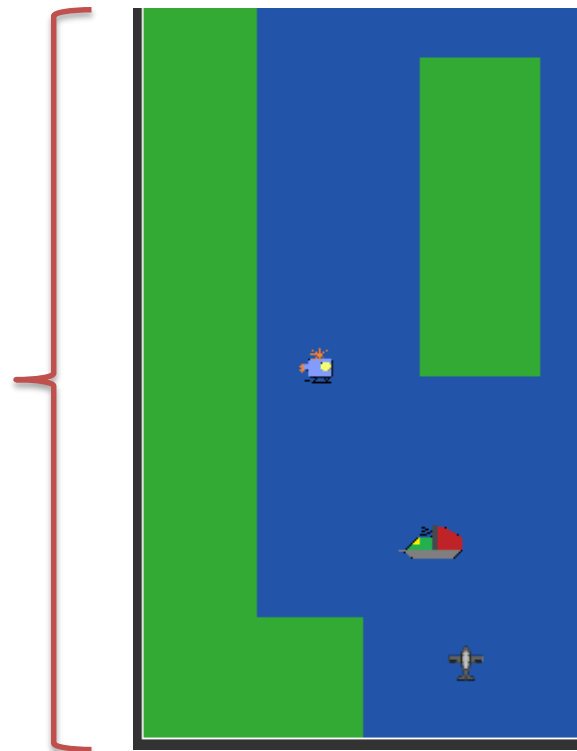
- Cada elemento da matriz terá uma largura e altura, sendo elementos quadrados: largura = altura.



Dimensões

- A largura, ou altura, geral do jogo, será definida pela largura de cada elemento x o número de elementos:

$$\text{AlturaTela} = \text{Largura} * \text{nLinhas}$$





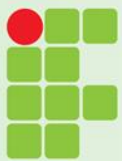
js/mapa01.js

```
var mapa01 = [  
  [1,1,1,1,1],  
  [1,0,0,0,1],  
  [1,0,1,0,1],  
  [1,0,0,0,1],  
  [1,1,1,1,1]  
];
```




js/Cenario.js

```
function Cenario(riverRaid) {  
    "use strict";  
    this.listaMapas = [];  
    this.listaMapas.push(mapa01);  
    this.mapaAtual = 0;  
    this.posicao = 0;  
    this.riverRaid = riverRaid;  
    this.colunasPorTela = mapa01[0].length;  
    Cenario.largura = Math.floor(riverRaid.canvas.width /  
                                this.colunasPorTela);  
    riverRaid.canvas.height = Math.floor(riverRaid.canvas.height /  
                                         Cenario.largura) * Cenario.largura;  
    riverRaid.canvas.width = Math.floor(riverRaid.canvas.width /  
                                         Cenario.largura) * Cenario.largura;  
    this.linhasPorTela = Math.floor(riverRaid.canvas.height /  
                                    Cenario.largura);  
    this.totalLinhasMatriz = mapa01.length;  
};
```



js/Cenario.js

...

```
Cenario.todosMapas = [mapa01];
```

```
Cenario.agua = 0; //Preto
```

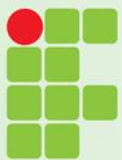
```
Cenario.terra = 1; //Branco
```

```
Cenario.ponte = 2; //Vermelho
```

```
Cenario.inimigoParado = 3; //Verde
```

```
Cenario.inimigoMovel = 4; //Azul
```

```
Cenario.posto = 5; //Cinza
```



js/Cenario.js

```
function Cenario(riverRaid) {  
    ...  
    this.desenhar = function (ctx) {  
        for (var ym = 0; ym < this.linhasPorTela; ym++) {  
            for (var xm = 0; xm < this.colunasPorTela; xm++) {  
                switch (this.getPosicaoMapa(xm, ym)) {  
                    case Cenario.terra:  
                        ctx.fillStyle = "#33AA33";  
                        ctx.fillRect(this.matrizParaPixelX(xm), this.matrizParaPixelY(ym,  
                            true), Cenario.largura, Cenario.largura);  
                        break;  
                }  
            }  
        }  
    };  
};
```



js/Cenario.js

```
function Cenario(riverRaid) {  
    ...  
  
    this.matrizParaPixelX = function (mx) {  
        return Math.floor(mx * Cenario.largura);  
    };  
  
    this.matrizParaPixelY = function (my, calcDif) {  
        var dif = 0;  
        if (calcDif) {  
            dif = this.posicao - Math.floor(this.posicao);  
        }  
        return ((this.linhasPorTela - 1) - my) * Cenario.largura + dif *  
            Cenario.largura;  
    };  
};
```



js/Cenario.js

```
function Cenario(riverRaid) {  
    ...  
    this.getPosicaoMapa = function (x, y) {  
        var mapa = Math.floor((this.posicao + y) / this.totalLinhasMatriz);  
        var sobra = Math.floor(this.posicao + y) % this.totalLinhasMatriz;  
        if (mapa >= this.listaMapas.length) {  
            this.incluirNovoMapa();  
        }  
        return this.listaMapas[mapa][sobra][x];  
    };  
  
    this.incluirNovoMapa = function () {  
        var n = Math.floor(Math.random() * Cenario.todosMapas.length);  
        var proximoMapa = Cenario.todosMapas[n];  
        this.listaMapas.push(proximoMapa);  
    };  
};
```



js/RiverRaid.js

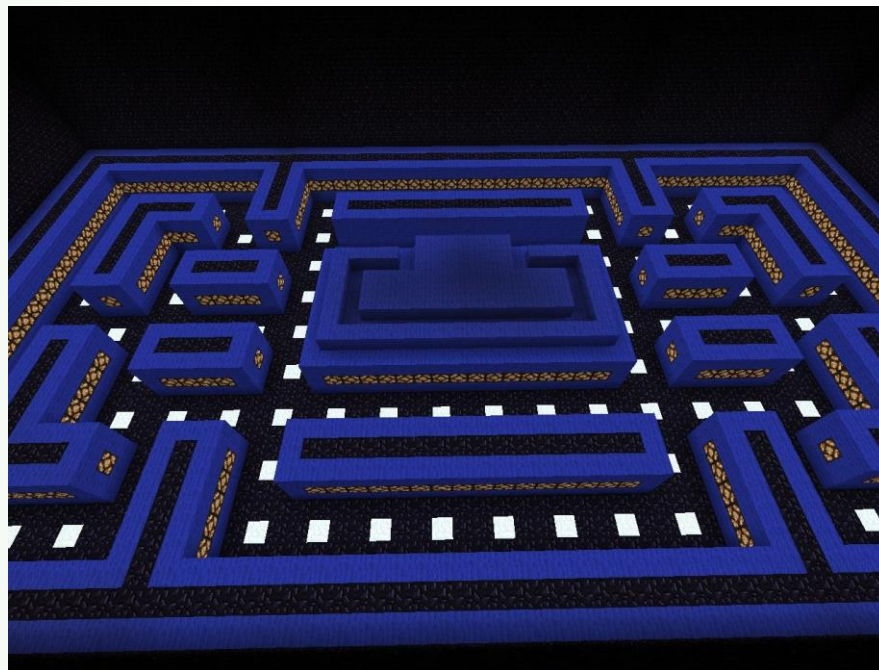
```
function RiverRaid (canvasID) {  
    "use strict";  
    this.canvas = document.getElementById(canvasID);  
    this.ctx = this.canvas.getContext("2d");  
    this.temporizador = null;  
    this.emPausa = true;  
    this.gameOver = false;  
    this.cenario = new Cenario(this);  
    this.desenharTudo = function () {  
        this.ctx.clearRect(0, 0, this.canvas.width, this.canvas.height);  
        this.cenario.desenhar(this.ctx);  
    };  
    this.desenharTudo();  
}; //Fim do construtor
```

```
RiverRaid.instancia = new RiverRaid("tela");
```

Testar!



INSTITUTO FEDERAL
SANTA CATARINA



River Raid

GERAÇÃO DE MAPAS

Geração de Mapas

- Em “Arquivos para Atividades Práticas”
 - Na pasta “riverraid”;
 - Baixe RiverRaidMapGenerator.zip;
 - Isto é um projeto NetBeans!
- Gerar um arquivo BMP no MS Paint!
 - Observar cores que irão definir objetos.
- Converter com o software Java!

Geração de Mapas

- Em “Arquivos para Atividades Práticas”
 - Na pasta “riverraid”;
 - Baixe “mapa01.png” para qualquer pasta
 - Converta o arquivo com o Gerador de Mapas baixado anteriormente;
 - salve o arquivo gerado com nome **js/mapa01.js** ;
 - Coloque o arquivo gerado dentro da pasta do seu projeto, substituindo o arquivo digitado anteriormente.

RiverRaid

- Teste sua página!



Testar!



INSTITUTO FEDERAL
SANTA CATARINA



River Raid

COLOCANDO VIDA



105
ANOS

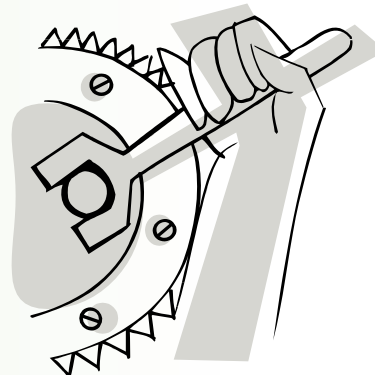
REDE FEDERAL
DE EDUCAÇÃO
PROFISSIONAL
E TECNOLÓGICA
1909-2014

O que precisamos?

- Fazer o “jogador se movimentar” ao iniciar e pressionar tecla?
- Inserir e fazer os inimigos se movimentarem:
 - Com qual intervalo de tempo?
 - Para qual direção?
- O que acontece se o jogador ordenar que o Avião se mova contra uma parede?
- O que ocorre quando o Avião atira?
- O que ocorre quando o Avião encontrar um inimigo?
 - E se for o tiro que encontrar um inimigo?
- E o combustível?

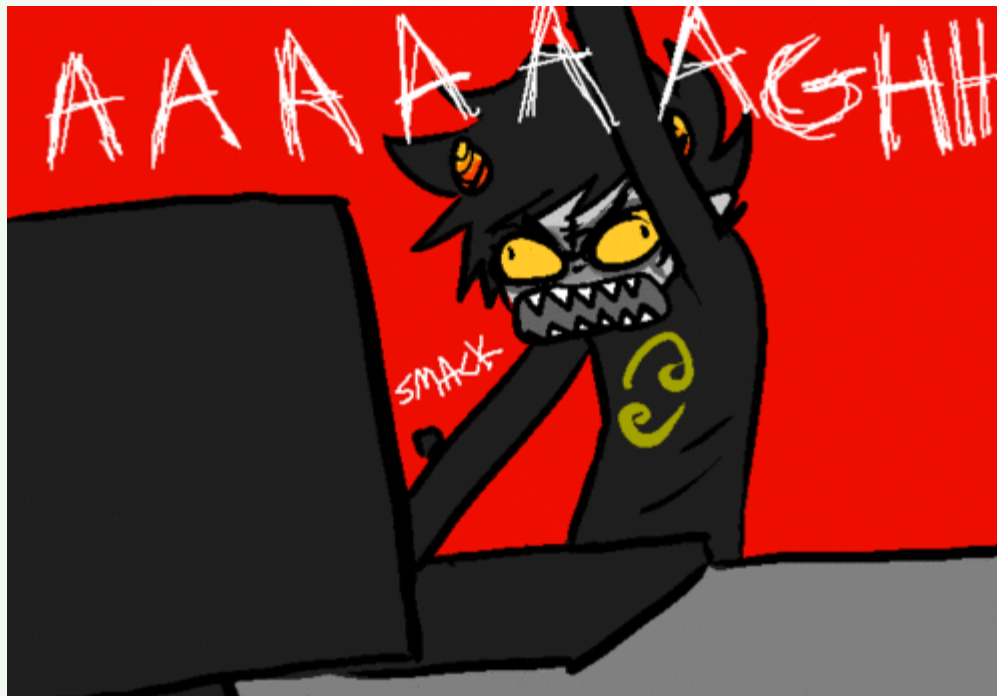
Movimentação do Jogo

- O Avião irá se movimentar continuamente para frente e irá responder conforme teclas que o usuário pressionar:
 - A, S, D, W, G;
 - Outras teclas configuradas;





INSTITUTO FEDERAL
SANTA CATARINA



River Raid

INTERAGINDO COM O USUÁRIO

Eventos!

- A interação é dada por uma troca entre a máquina e o usuário;
- A máquina fornece principalmente imagens que descrevem uma situação, onde pode ser necessária a intervenção do usuário;
- O usuário irá intervir basicamente através de comandos!
 - Comandos são captados através de eventos.

Eventos!

- Nosso **document** possui propriedades de eventos que podem ser associadas à funções quaisquer;
- Estas funções determinam algo a ser feito quando aquele evento ocorrer:
 - **document.onkeydown**
 - Ao descer uma tecla qualquer;
 - **document.onkeyup**
 - Ao soltar uma tecla qualquer;



INSTITUTO FEDERAL
SANTA CATARINA

js/RiverRaid.js

```
document.onkeydown = function (evt) {  
    document.getElementById("tecla").innerHTML = evt.keyCode;  
}
```

Teste!



js/RiverRaid.js

No final do arquivo

```
document.onkeydown = function (evt) {  
    "use strict";  
    RiverRaid.instancia.pressionarTecla(evt);  
};  
  
document.onkeyup = function (evt) {  
    "use strict";  
    RiverRaid.instancia.liberarTecla(evt);  
};
```

Apagar o
<p id="tecla">
do HTML



js/RiverRaid.js

```
function RiverRaid (canvasID) {
```

```
...
```

```
}; //Fim do construtor
```

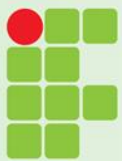
```
RiverRaid.teclaEsquerda = 65; //A
```

```
RiverRaid.teclaDireita = 68; //D
```

```
RiverRaid.teclaCima = 87; //W
```

```
RiverRaid.teclaBaixo = 83; //S
```

```
RiverRaid.teclaTiro = 71; //G
```



js/RiverRaid.js (1/3)

```
function RiverRaid (canvasID) {  
  ...  
  this.pressionarTecla = function (evt) {  
    switch (evt.keyCode) {  
      case RiverRaid.teclaEsquerda:  
        this.jogador.setMoverEsquerda(true);  
        if (this.emPausa) {  
          this.pausar();  
        }  
        evt.preventDefault();  
        break;  
    }  
  }  
}
```



js/RiverRaid.js (2/3)

```
case RiverRaid.teclaDireita:
    this.jogador.setMoverDireita(true);
    if (this.emPausa) {
        this.pausar();
    }
    evt.preventDefault();
    break;
case RiverRaid.teclaCima:
    this.jogador.setAcelerar(true);
    if (this.emPausa) {
        this.pausar();
    }
    evt.preventDefault();
    break;
```



js/RiverRaid.js (3/3)

```
case RiverRaid.teclaBaixo:
    this.jogador.setFrear(true);
    if (this.emPausa) {
        this.pausar();
    }
    evt.preventDefault();
    break;
```

```
case RiverRaid.teclaTiro:
    this.jogador.setAtirar(true);
    if (this.emPausa) {
        this.pausar();
    }
    evt.preventDefault();
    break;
```

```
}
```

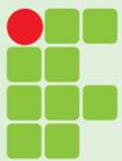
```
};
```

```
} //Fim do construtor
```



js/RiverRaid.js (1/2)

```
function RiverRaid (canvasID) {  
    ...  
    this.liberarTecla = function (evt) {  
        switch (evt.keyCode) {  
            case RiverRaid.teclaEsquerda:  
                this.jogador.setMoverEsquerda(false);  
                evt.preventDefault();  
                break;  
            case RiverRaid.teclaDireita:  
                this.jogador.setMoverDireita(false);  
                evt.preventDefault();  
                break;  
            case RiverRaid.teclaCima:  
                this.jogador.setAcelerar(false);  
                evt.preventDefault();  
                break;  
        }  
    }  
}
```



js/RiverRaid.js (2/2)

...

```
case RiverRaid.teclaBaixo:
    this.jogador.setFrear(false);
    evt.preventDefault();
    break;
case RiverRaid.teclaTiro:
    this.jogador.setAtirar(false);
    evt.preventDefault();
    break;
} //Fim do switch
};
```



js/Jogador.js (1/3)

```
function Jogador(nomeArq, inicioX, posicaoY, riverRaid) {  
    "use strict";  
    this.imagem = new Image();  
    this.imagem.src = nomeArq;  
    this.imagem.onload = RiverRaid.desenharTudo;  
    this.x = inicioX;  
    this.y = posicaoY;  
    this.esquerda = false;  
    this.direita = false;  
    this.acelerar = false;  
    this.frear = false;  
    this.atirar = false;  
    this.velocidade = 1;  
    this.riverRaid = riverRaid;  
    ...  
}
```




js/Jogador.js (2/3)

...

```
this.desenhar = function (ctx) {  
    ctx.drawImage(this.imagem, this.x, this.y, Jogador.largura,  
                  Jogador.largura);  
};  
this.setMoverEsquerda = function (booleano) {  
    this.esquerda = booleano;  
    if (this.esquerda) {  
        this.direita = false;  
    }  
};  
this.setMoverDireita = function (booleano) {  
    this.direita = booleano;  
    if (this.direita) {  
        this.esquerda = false;  
    }  
};
```

...



js/Jogador.js (3/3)

```
this.setAcelerar = function (booleano) {  
    this.acelerar = booleano;  
    if (this.acelerar) {  
        this.frear = false;  
    }  
};  
  
this.setFrear = function (booleano) {  
    this.frear = booleano;  
    if (this.frear) {  
        this.acelerar = false;  
    }  
};  
  
this.setAtirar = function (booleano) {  
    this.atirar = booleano;  
};  
  
} //Fim do construtor  
  
Jogador.largura = 20;
```



Baixar
arquivo

js/RiverRaid.js

```
//No inicio do construtor (após declarar this.canvas)
this.jogador = new Jogador("img/aviao.png", this.canvas.width / 2 -
    5, this.canvas.height - this.canvas.height / 4, this);

...

//Dentro do construtor - atualizar o método
this.desenharTudo = function () {
    this.ctx.clearRect(0, 0, this.canvas.width, this.canvas.height);
    this.jogador.desenhar(this.ctx);
    this.cenario.desenhar(this.ctx);
};

...

//Fora do construtor - estático
RiverRaid.desenharTudo = function () {
    RiverRaid.instancia.desenharTudo();
};
```

Testar!



INSTITUTO FEDERAL
SANTA CATARINA



River Raid

CONTROLE GLOBAL DO MOVIMENTO

105
ANOS

REDE FEDERAL
DE EDUCAÇÃO
PROFISSIONAL
E TECNOLÓGICA
1909-2014

Movendo com Tempo

- Todo tipo de movimento tem uma velocidade;
- Como determinamos a velocidade de algum objeto?
 - Medida Espacial / Tempo!
 - KM/h
 - m/s
 - ...



Controlando o Tempo

- Como já definimos, a unidade de espaço de cada movimento do cenário e dos inimigos será por posição da matriz;
- Agora precisamos determinar o *intervalo de tempo* que nosso jogo irá usar para fazer cada movimento dos elementos;
- Como nosso jogo gira em torno do “avanço” do jogador este tempo será um guia para todo o jogo.



Controlando o Tempo

- Função JavaScript:
 - `relogio = setInterval(“NomeFuncao()”, intervalo);`
 - **relogio** é uma referência ao timer/clock que foi criado;
 - **NomeFuncao()** é a função que será executada a cada intervalo;
 - **intervalo** é um número inteiro representando a quantidade em milissegundos de intervalo entre uma execução e outra da função NomeFuncao().
 - `clearInterval(relogio);`
 - Para o relógio de repetição;





js/Jogador.js

```
this.mover = function () {  
    if (this.esquerda) {  
        this.x -= 2;  
    } else if (this.direita) {  
        this.x += 2;  
    }  
    if (this.acelerar) {  
        this.velocidade = 1;  
    } else if (this.frear) {  
        this.velocidade = 0.25;  
    } else {  
        this.velocidade = 0.5;  
    }  
    this.riverRaid.cenario.andar(this.velocidade);  
};
```



js/Cenario.js

```
this.andar = function (espaco) {  
    this.posicao += espaco;  
};
```



js/RiverRaid.js

```
1 RiverRaid.novoJogo = function (canvasID) {  
    "use strict";  
    if (!RiverRaid.instancia.emPausa) {  
        RiverRaid.instancia.pausar();  
    }  
    RiverRaid.instancia = new RiverRaid(canvasID);  
};  
RiverRaid.pausar = function () {  
    "use strict";  
    RiverRaid.instancia.pausar();  
};  
RiverRaid.atualizar = function () {  
    "use strict";  
    RiverRaid.instancia.atualizarJogo();  
};
```



js/RiverRaid.js

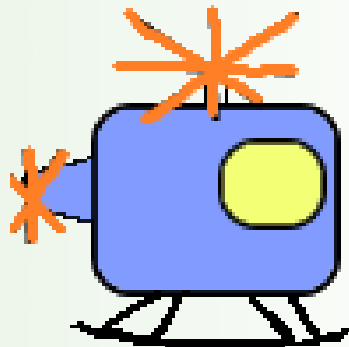
```
this.atualizarJogo = function () {  
    this.jogador.mover();  
    this.desenharTudo();  
};  
  
this.pausar = function () {  
    this.emPausa = !this.emPausa;  
    if (this.emPausa) {  
        clearInterval(this.temporizador);  
        this.temporizador = null;  
    } else if (!this.gameOver){  
        this.temporizador = setInterval(RiverRaid.atualizar,  
                                         RiverRaid.intervalo);  
    }  
};
```

```
RiverRaid.intervalo = 25;//ms
```

Testar!



INSTITUTO FEDERAL
SANTA CATARINA



River Raid

OS VILÕES!

Os Vilões

- No River Raid, os vilões são “inimigos” que são encontrados ao longo do rio;
- Cada mapa terá um número e uma disposição diferente de inimigos.
- Em nossa implementação, cada inimigo será uma instância de uma classe chamada Inimigo;
- Cada vez que encontramos **Cenario.inimigoParado** no mapa, instanciamos um novo objeto Inimigo.



js/Inimigo.js (1/3)

```
function Inimigo(px, my, riverRaid) {  
    "use strict";  
    if (Inimigo.imagens == null) {  
        Inimigo.imagens = [];  
        Inimigo.imagens[0] = new Image();  
        Inimigo.imagens[0].src = "img/inimigo1.png";  
        Inimigo.imagens[0].onload = RiverRaid.desenharTudo;  
        Inimigo.imagens[1] = new Image();  
        Inimigo.imagens[1].src = "img/inimigo2.png";  
        Inimigo.imagens[1].onload = RiverRaid.desenharTudo;  
    }  
} //Fim do construtor  
  
Inimigo.imagens = null;
```

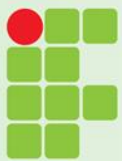
Baixar
arquivo

Baixar
arquivo



js/Inimigo.js (2/3)

```
function Inimigo(px, my, riverRaid) {  
    ...  
    this.x = px;  
    this.y = my;  
    this.riverRaid = riverRaid;  
    this.altura = Jogador.largura;  
    this.largura = 0;  
    this.tipo = Math.floor(Math.random() * 2);  
    this.imagem = Inimigo.imagens[this.tipo];  
    if (this.tipo == 0) {  
        this.largura = Jogador.largura;  
    } else { //Barco  
        this.largura = Jogador.largura * 2;  
    }  
}
```



js/Inimigo.js (3/3)

```
function Inimigo(px, my, riverRaid) {  
    this.desenhar = function (ctx) {  
        ctx.drawImage(this.imagem, this.x,  
            riverRaid.cenario.matrizParaPixelY(this.y),  
            this.largura, this.altura);  
    };  
    Inimigo.todos.push(this);  
} //Fim do construtor
```

Inimigo.todos = []; //Vetor de todos os inimigos existentes.

```
Inimigo.desenharTodos = function (ctx) {  
    for (var i = 0; i < Inimigo.todos.length; i++) {  
        Inimigo.todos[i].desenhar(ctx);  
    }  
};
```



js/Cenario.js (1/2)

```
//inicialização (últimas linhas do construtor)
for (var j = 0; j < this.totalLinhasMatriz; j++) {
    for (var i = 0; i < this.colunasPorTela; i++) {
        var pos = this.getPosicaoMapa(i, j);
        switch(pos) {
            case Cenario.ponte:
                riverRaid.jogador.x = this.matrizParaPixelX(i);
                riverRaid.jogador.y = this.matrizParaPixelY(j);
                break;
            case Cenario.inimigoParado:
                new Inimigo(this.matrizParaPixelX(i), j,
                           this.riverRaid);
                break;
        }
    }
}
```



js/Cenario.js (2/2)

```
this.incluirNovoMapa = function () {  
    var n = Math.floor(Math.random() * Cenario.todosMapas.length);  
    var proximoMapa = Cenario.todosMapas[n];  
    this.listaMapas.push(proximoMapa);  
    for (var j = 0; j < this.totalLinhasMatriz; j++) {  
        for (var i = 0; i < this.colunasPorTela; i++) {  
            var posMapaAtual = Math.floor(this.posicao) %  
                                     this.totalLinhasMatriz;  
            var jnovoMapa = j + (this.totalLinhasMatriz -  
                                posMapaAtual);  
            switch(proximoMapa[j][i]) {  
                case Cenario.inimigoParado:  
                    new Inimigo(this.matrizParaPixelX(i), jnovoMapa,  
                                this.riverRaid);  
                    break;  
            }  
        }  
    }  
};
```



js/RiverRaid.js (2/2)

```
this.desenharTudo = function () {  
    this.ctx.clearRect(0, 0, this.canvas.width, this.canvas.height);  
    this.jogador.desenhar(this.ctx);  
    Inimigo.desenharTodos(this.ctx);  
    this.cenario.desenhar(this.ctx);  
};
```

Testar!



INSTITUTO FEDERAL
SANTA CATARINA

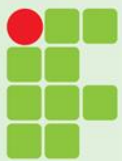


River Raid

MOVIMENTAÇÃO DOS INIMIGOS

Movimentação dos Inimigos

- Em nossa primeira implementação, os inimigos não irão, de fato, se movimentar.
- Precisamos implementar apenas a movimentação deles em relação à tela, pois à medida em que o jogador avança pelo cenário, estes inimigos devem avançar juntos.



js/Inimigo.js

```
//Dinâmico
```

```
  this.mover = function (espaco) {  
    this.y -= espaco;  
    if (this.y < 0) {  
      var minhaPos = Inimigo.todos.indexOf(this);  
      Inimigo.todos.splice(minhaPos, 1);  
    }  
  };  
};
```

```
//Estático
```

```
Inimigo.andarTodos = function (espaco) {  
  for (var i = Inimigo.todos.length - 1; i >= 0; i--) {  
    Inimigo.todos[i].mover(espaco);  
  }  
};
```



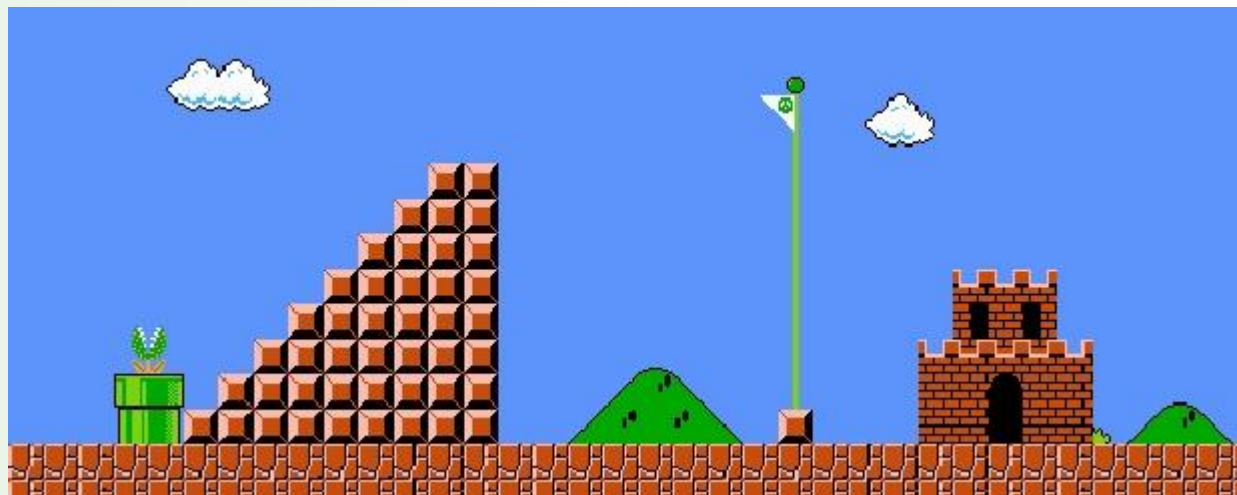
js/Cenario.js

```
this.andar = function (espaco) {  
    this.posicao += espaco;  
    Inimigo.andarTodos(espaco);  
};
```

Testar!

Alterações Restantes

- O que falta alterar?
 - Atirar!
 - E quando acerta:
 - Cenário?
 - Inimigo?
 - **Fim de jogo?**
 - Sons?





js/Inimigo.js (1/2)

```
this.detectarColisao = function () {  
    var jxi = riverRaid.jogador.x;  
    var jxf = jxi + Jogador.largura;  
    var jyi = riverRaid.jogador.y;  
    var jyf = jyi + Jogador.largura;  
    var ixi = this.x;  
    var ixf = ixi + this.largura;  
    var iyi = this.riverRaid.cenario.matrizParaPixelY(this.y, true);  
    var iyf = iyi + this.altura;  
    var fora = jxf < ixi || jxi > ixf || jyf < iyi || jyi > iyf;  
    if (!fora){  
        return Cenario.jogador;  
    }  
    return Cenario.agua;  
};
```



js/Inimigo.js (2/2)

```
Inimigo.detectarColisaoTodosJogador = function () {  
    for (var i = 0; i < Inimigo.todos.length; i++) {  
        if (Inimigo.todos[i].detectarColisao() == Cenario.jogador) {  
            return Cenario.jogador;  
        }  
    }  
    return Cenario.agua;  
};
```



js/RiverRaid.js

```
this.atualizarJogo = function () {  
    this.jogador.mover();  
    this.desenharTudo();  
    if (Inimigo.detectarColisaoTodosJogador() == Cenario.jogador) {  
        this.gameOver = true;  
    }  
    if (this.gameOver) {  
        this.pausar();  
        console.log("Game Over!");  
    }  
};
```



GAME OVER

Testar!



```
this.detectarColisaoTerra = function (x, y, lar, alt) {
    return this.detectarColisaoTerraXY(x, y) ||
        this.detectarColisaoTerraXY(x + lar, y) ||
        this.detectarColisaoTerraXY(x, y + alt) ||
        this.detectarColisaoTerraXY(x + lar, y + alt) ||
        this.detectarColisaoTerraXY(Math.round(x + lar / 2),
            Math.round(y + alt / 2));
};
```




js/Cenario.js (2/2)

```
this.pixelParaMatrizX = function (px) {  
    return Math.floor(px / Cenario.largura);  
};
```

```
this.pixelParaMatrizY = function (py) {  
    return (this.linhasPorTela - 1) - Math.floor(py / Cenario.largura);  
};
```



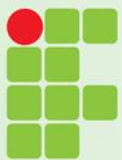
js/RiverRaid.js

```
this.atualizarJogo = function () {  
    this.jogador.mover();  
    this.desenharTudo();  
  
    this.gameOver = this.cenario.detectarColisaoTerra(this.jogador.x,  
        this.jogador.y, Jogador.largura, Jogador.largura);  
  
    if (Inimigo.detectarColisaoTodosJogador() == Cenario.jogador) {  
        this.gameOver = true;  
    }  
    if (this.gameOver) {  
        this.pausar();  
        console.log("Game Over!");  
    }  
};
```



GAME OVER

Testar!



INSTITUTO FEDERAL
SANTA CATARINA



River Raid

OS TIROS!



Os Tiros

- Depois de ser disparado(criado), um tiro saíra do ponto central do jogador;
- Os tiros deverão se movimentar para frente mais rápido do que o jogador;
- Ao movimentar o jogador para os lados, o os tiros já disparados se movem junto;
- Caso um determinado tiro atinja o cenário, ele deverá parar e ser removido;
- Caso um tiro atinja um determinado inimigo, ambos deverão ser eliminados (removidos).
- Deve haver um limite para quantos tiros poderão ser disparados por segundo.





js/Tiro.js (1/3)

```
function Tiro(riverRaid) {  
    "use strict";  
    this.riverRaid = riverRaid;  
    this.y = riverRaid.jogador.y;  
    this.varX = Math.floor(Jogador.largura / 2);  
    this.velocidade = 10;  
    this.altura = 6;  
    Tiro.todos.push(this);  
}  
  
Tiro.todos = [];
```



js/Tiro.js (2/3)

```
this.getX = function () {  
    return this.riverRaid.jogador.x + this.varX;  
};
```

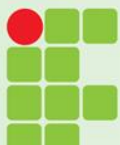
```
this.desenhar = function (ctx) {  
    ctx.strokeStyle = "#FFAA00";  
    ctx.lineWidth = 3;  
    ctx.beginPath();  
    ctx.moveTo(this.getX(), this.y);  
    ctx.lineTo(this.getX(), this.y + this.altura);  
    ctx.stroke();  
    ctx.lineWidth = 1;  
};
```

```
this.mover = function () {  
    this.y -= this.velocidade;  
};
```



js/Tiro.js (3/3)

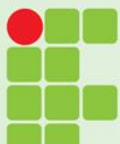
```
Tiro.desenharTodos = function (ctx) {  
    "use strict";  
    for (var i = 0; i < Tiro.todos.length; i++) {  
        Tiro.todos[i].desenhar(ctx);  
    }  
};  
  
Tiro.moverTodos = function () {  
    "use strict";  
    var cenario = RiverRaid.instancia.cenario;  
    for (var i = Tiro.todos.length - 1; i >= 0; i--) {  
        var ta = Tiro.todos[i];  
        ta.mover();  
        if (ta.y + ta.altura <= 0 ||  
            cenario.detectarColisaoTerra(ta.getX(), ta.y, 3, ta.altura)) {  
            Tiro.todos.splice(i, 1);  
        }  
    }  
};
```

js/Jogador.js

```
//Incluir como atributos do objeto
this.ultimoTiro = -100;
this.entreTiros = 15;
this.repeticoes = 0;

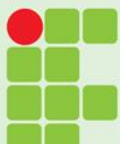
//Incluir como verificação ao fim do método mover
if (this.atirar && this.repeticoes - this.ultimoTiro >=
                                     this.entreTiros) {
    new Tiro(this.riverRaid);
    this.ultimoTiro = this.repeticoes;
}
this.repeticoes++;
```



js/RiverRaid.js

```
//Incluir dentro do this.atualizarJogo, após this.jogador.mover()  
Tiro.moverTodos();  
  
//Incluir ao fim do método this.desenharTudo()  
Tiro.desenharTodos(this.ctx);
```

Testar!



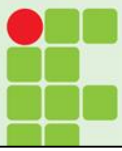
js/Tiro.js (1/2)

```
this.detectarColisaoInimigo = function (px, py, largura, altura) {  
  if (this.getX() >= px && this.getX() <= px + largura &&  
    ((this.y >= py && this.y <= py + altura) ||  
    (this.y + this.altura >= py &&  
    this.y + this.altura <= py + altura))) {  
    return true;  
  }  
  return false;  
};
```



js/Tiro.js (2/2)

```
1 Tiro.detectarColisaoTodos = function () {  
    "use strict";  
    var cen = RiverRaid.instancia.cenario;  
    for (var t = Tiro.todos.length - 1; t >= 0; t--) {  
        for (var i = Inimigo.todos.length - 1; i >= 0; i--) {  
            var iA = Inimigo.todos[i];  
            if (Tiro.todos[t].detectarColisaoInimigo(iA.x,  
                cen.matrizParaPixelY(iA.y, true), iA.largura,  
                iA.altura)) {  
                Inimigo.todos.splice(i, 1);  
                Tiro.todos.splice(t, 1);  
                i = -1;  
            }  
        }  
    }  
};
```



js/RiverRaid.js

//Incluir no inicio do construtor

Inimigo.todos.length = 0; //zerar número de inimigos em novo jogo

Tiro.todos.length = 0;

//Incluir após Tiro.moverTodos dentro do método this.atualizarJogo()

Tiro.detectarColisaoTodos();

Testar!



INSTITUTO FEDERAL
SANTA CATARINA



River Raid

ESTÍMULOS SONOROS

Estímulos Sonoros

- Conforme comentado anteriormente, quanto mais estimularmos, de forma positiva, os sentidos dos jogadores, maior a probabilidade dele se sentir como parte do jogo;
- Para isto, iremos adicionar alguns pequenos sons associados a eventos como colisões;
- Baixe os arquivos e salve na subpasta **snd**:
 - explosao.mp_
 - explosao.ogg
 - tiro.mp_
 - tiro.ogg

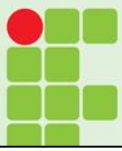
<audio> e <source>

- HTML 5!
- MIME Types:
 - MP3 – audio/mpeg
 - Ogg – audio/ogg
 - Wav – audio/wav
- Suporte:
 - Ps.: Múltiplos <source> fornecem redundância!
- Suporte dos diferentes navegadores

index.html

...

```
<audio controls id="sndExplosao">  
  <source src="snd/explosao.mp_" type="audio/mpeg">  
  <source src="snd/explosao.ogg" type="audio/ogg">  
</audio>  
<audio controls id="sndTiro">  
  <source src="snd/tiro.mp_" type="audio/mpeg">  
  <source src="snd/tiro.ogg" type="audio/ogg">  
</audio>  
<script src="RiverRaid.js"></script>
```

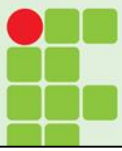


js/Inimigo.js

```
//Incluir no fim do arquivo
```

```
Inimigo.somExplosao = document.getElementById("sndExplosao");
```

```
Inimigo.somExplosao.volume = 0.5;
```



js/Tiro.js

```
//Incluir dentro do construtor
```

```
Tiro.som.cloneNode().play();
```

```
//Incluir dentro do IF do Tiro.detectarColisaoTodos
```

```
Inimigo.somExplosao.cloneNode().play();
```

```
//Incluir no fim do arquivo
```

```
Tiro.som = document.getElementById("sndTiro");
```

Testar!



INSTITUTO
SANTO ANTÔNIO

Trabalho

- | | | |
|---|----|---|
| 1 | 1. | Customize cores e outras configurações do arquivo de estilo; |
| | 2. | Customize cores, tamanhos e disposição dos objetos do jogo (dentro do <i>Javascript</i>). Utilize gradientes e/ou imagens; |
| | 3. | Complete o HTML informando o nome da disciplina, o nome do instituto e o seu nome, dispondo os elementos com layouts CSS; |
| 1 | 4. | Crie um placar com pontuação; |
| 1 | 5. | Crie uma indicação visual dentro do Canvas de fim de jogo ; |
| 1 | 6. | Adicione novos sons a eventos diferentes no jogo; |
| 1 | 7. | Adicione teclas de atalho para “Pausa” e para “Novo Jogo”; |
| 5 | 8. | Ao evoluir no jogo, crie novos desafios para o jogador;
– <u>Adicione outros elementos a seu critério</u> ;
– Exemplos: Combustível e postos, inimigos móveis, novos tipos de inimigos, novos tipos de armas, itens que podem ser coletados, chefões, pontes, múltiplas vidas, |
| | 9. | Entregue os arquivos por e-mail ao Professor junto com uma descrição/resposta para cada item do trabalho . |