

CIÊNCIAS DA COMPUTAÇÃO

“DESENVOLVIMENTO DE UM JOGO COM UTILIZAÇÃO DE INTERFACE GRÁFICA”

Alunos:

Chinnon Santos – RA: B5526C-7.

Leonardo Felipe da Silva - RA: B5680D-5.

Lucas Aleixo Mendonça - RA: B25444-0.

Magdiel Alves de Oliveira - RA: B4707G-0.

Goiânia 11/2013

SUMÁRIO



OBJETIVO E MOTIVAÇÃO DO TRABALHO.....3 – 4.



INTRODUÇÃO..... 5.



REGRAS DO JOGO (CONCEITOS GERAIS).....5 – 6.



PLANO DE DESENVOLVIMENTO DO JOGO.....6 – 7.



PROJETO DO PROGRAMA..... 8.



CÓDIGO FONTE DO JOGO.....9 – 24.



FONTES/BIBLIOGRAFIA..... 25.



IMAGENS DO JOGO.....26 – 28.

Objetivo do trabalho e Motivação do Trabalho

Os jogos educativos são apresentados à humanidade desde que ela existe. O jogo está presente na natureza humana, seja ela na infância ou na fase adulta. Brincar é do cotidiano de todas as crianças e adultos, e não tem uma data de início.

Antigamente os jogos eram apresentados como forma de divertimento, mas algum tempo os jogos foram introduzidos para ajudar em abordagens como melhorar a concentração, habilidade motora e percepções de todos os sentidos, que são os jogos sensoriais, e porque não falarmos dos jogos que são ligados diretamente à educação, ou seja os jogos educativos.

Os jogos educativos começaram a crescer dentro das escolas e também entre as famílias. Atualmente os jogos são divididos em várias categorias, dependendo da proposta do jogo. É importante que o jogo educativo chegue às crianças como uma proposta de brincadeira. A brincadeira com os jogos educativos precisa ser prazerosa; proposta e nunca imposta.

Muitos jogos estão sendo deixado de lado com o advento da internet, porém é de extrema importância que a criança trabalhe o movimento e pensamento, através de jogos corporais, eletrônicos e sensoriais. O jogo é um instrumento de aprendizagem. A criança e o adulto aprendem por aquilo que ela vive e não pelo que ela escuta.

Por que voltado para o meio ambiente?

No Brasil, para Dias (2004), a década de 1990 é a década das maiores transformações dessa temática. A nova mentalidade do Ministério da Educação, os surgimentos de centros de educação ambiental e encontros regionais e nacionais foram importantes para aproximar as discussões mundiais a respeito dos fundamentos de uma comprometida Educação Ambiental. A ECO-92 reforça as ideias da Conferência de Tbilisi e propõe a capacitação de educadores e a mobilização social acerca da problemática ambiental. Na Educação a maior contribuição se deu com a reforma curricular

no ensino fundamental prevista nos Parâmetros Curriculares Nacionais (PCNs), o qual estabelece a Educação Ambiental como um tema transversal que preza pela interdisciplinaridade.

A importância da educação ambiental

A Educação Ambiental é necessária para indivíduos que ainda estão em processo de formação de valores, pois pode contribuir para uma mudança ou o direcionamento dos valores, e conseqüentemente, dinamizar a reprodução de valores integrativos. A prática necessita de projetos complementares e atividades didáticas de campo para atingir seus objetivos. Somente com estas atividades práticas serão valorizados conceitos importantes para uma cidadania ambientalmente responsável.

Lixo Certo

O referido jogo tem por objetivo geral, ensinar de forma divertida o assunto reciclagem. Vemos no nosso cotidiano vários problemas neste sentido, papel que poderia ser reaproveitado, plástico que poderia retornar novamente a nós, e assim por diante, contudo, este reaproveitamento não acontece por falta de um descarte consciente. Ah, sem falar no descarte incorreto de eletrônicos, o que acaba então por poluir de forma inevitável o habitat natural do homem e dos animais, comprometendo assim o futuro da Terra. Jogo simples, fácil de ser jogado, mas quando colocado na prática no dia a dia, traz a certeza de uma missão, e porque não falar de uma responsabilidade cumprida.

INTRODUÇÃO

Partindo do proposto pela Instituição, desenvolvemos um programa em JAVA com o intuito de demonstrar a vida em uma grande metrópole visando os parâmetros ambientais envolvidos.

Desenvolvido em caráter educacional visando aumentar a percepção das pessoas aos problemas que são causados pelos cidadãos de grandes cidades pela negligência e falta de informação em soluções inteligentes como coleta seletiva e reciclagem do lixo.

Sentimos a responsabilidade social de provocar a mudança no pensamento e na cultura do nosso público-alvo, que é as pessoas compreendidas nas metrópoles e circunvizinhanças.

Através da inclusão digital proporcionada pelo avanço da tecnologia, esperamos atingir um público vasto e variado, com todas as idades e níveis sociais, uma vez que o problema atinge a todos.

Como informação técnica, o software foi desenvolvido através da linguagem JAVA, como sugerido pela Instituição.

Regras do jogo (conceitos gerais)

- Acerta o número máximo de lixo exatamente na boca da lixeira correta de acordo com o lixo que aparecer na tela no **prazo de 1 (um) minuto**.
- O acerto do lixo corretamente na boca da lixeira e o tipo do lixo ser correspondente ao tipo da lixeira, **soma 10 (dez) pontos** à sua pontuação total.
- O acerto do lixo corretamente na boca da lixeira, mas e o tipo do lixo NÃO ser correspondente ao tipo da lixeira **subtrai 5 (cinco) pontos** à sua pontuação total.
- O não acerto do lixo corretamente na boca da lixeira, **subtrai 10 (dez) pontos** à sua pontuação total.
- O resultado da pontuação total no término de 1 minuto for positivo (maior que

zero), você recebe os parabéns e cumpre o objetivo do jogo.

- O resultado da pontuação total no término de 1 minuto for negativo (menor ou igual a zero), você recebe uma mensagem de fracasso.

- A qualquer momento do jogo é possível voltar ao menu inicial e encerrar a partida atual.

- A quantidade de lixos exibidas são ilimitadas.

- O aparecimento de tipos de lixo é definida de forma aleatória pelo sistema do jogo, podendo ocorrer repetições seguidas do mesmo tipo de lixo, elevando o nível de dificuldade.

- Após o término dos 60 segundos não é possível mais jogar lixos nas lixeiras sendo que o mesmo não é mais exibido.

- Você pode jogar quantas vezes quiser, não há limites de jogada, também poderá abandonar a partida quantas vezes quiser, não há limitações para abandono de jogo.

Plano de desenvolvimento do jogo

O plano de jogo foi elaborado em 4 fases, tendo como base a gerencia de desenvolvimentos de grandes jogos do mercado atual.

1ª Fase – Roteiro:

Cada vez mais se assemelham a roteiros de filmes. Este é um item fundamental para o processo de criação e será o elemento crucial para convencer os investidores da potencialidade do nosso jogo. É nesse item que o jogo deve mostrar seu diferencial em relação aos outros. Chamam-se aos roteiros de jogos de roteiros interativos, pois diferentemente que os roteiros de filmes, devem ter espaço para interferência do usuário no desencadeamento da história.

2ª Fase - Game Design:

Entende-se por game design a conceituação artística do jogo. Hoje em dia, dada a complexidade das histórias e dos cenários elaborados é importante que esta parte do documento seja escrita por um artista. Dentro deste item deverão ser expostos quais as principais características dos cenários, esboços de personagens, descrição das texturas fundamentais, mapas e descrições dos combates. (também denominado de *level design*).

3ª Fase - Game Play:

Nesta parte do documento deve descrever-se como será a jogabilidade. PRINCIPAL ETAPA DE UM JOGO, Por jogabilidade entendem-se as regras do jogo e o balanceamento das regras (*game balancing*). Nesta descrição deve ficar claro que o jogo é divertido e irá proporcionar desafios interessantes. Esta parte do documento é muito importante para guiar os programadores principalmente na etapa de *scripting*. Uma mal elaboração dessa etapa pode influenciar seriamente em um mal desenvolvimento (bugs) e atrasos em todo o projeto.

4ª Fase - Interface Gráfica:

Pode-se dividir a interface em *in game* e *out game*. A primeira consiste na instrumentação disponível durante o jogo e é responsável pela entrada de dados do jogador para a aplicação. A interface *out game* é a forma de apresentar a introdução do jogo, sua configuração, instruções, carregar o jogo, entre outras operações de suporte. Costuma-se dizer que a melhor interface é aquela que passa despercebida para o jogador, permitindo que o mesmo possa focar-se no desenrolar da história e das ações. Resumindo, o *in game* é a parte do jogo que se é jogável, a parte onde você possui um tempo para atingir o objetivo que é fazer pontuação acertando corretamente o tipo de lixo em sua lixeira, já o *out game* é a nossa abertura do Marinho correndo e a tela onde você escolhe jogar ou sair do jogo.

Projeto do programa

O projeto foi simples, e consistiu na sequência das fases citadas no plano de desenvolvimento do jogo, após reunir todas as informações necessárias e deixar tudo definido do jogo, partimos para o desenvolvimento onde usamos a ferramenta de desenvolvimento em JAVA chamada ECLIPSE para criar os algoritmos e gerar a aplicação do jogo. Baseado em nosso conhecimento adquirido em boa parte nas aulas de LPOO e ALPOO, e também em alguns tutoriais na internet, conseguimos concluir com sucesso o que estava previsto e planejado dentro dos prazos estipulados. Nas primeiras horas de desenvolvimento foi um pouco complexa, iniciamos a parte visual para facilitar é ter visão de como o jogo se sairia com o decorrer do desenvolvimento, aplicando métodos e classe necessários de acordo com o caminhar da programação.

Passamos por algumas dificuldades que devemos citar, está no qual se refere ao manipula mento de áudio do jogo, inicialmente tentamos trabalhar com o formato de áudio em. MP3, foi um fracasso pois ainda não existem bibliotecas satisfatórias que possa trabalhar em conjunto com algoritmos de **Buffer**, que no caso os sistemas de Buffer são a peça-chave para o funcionamento de todo o jogo. Passamos a fazer testes com o formato. OGG e também não obtemos sucesso, até então irmos direto ao formato. MIDI do qual o Java possui uma biblioteca nativa de suporte a esse formato, e também é o formato mais recomendado para pequenos jogos.

Quanto ao resto foi bem produtivo o tempo, sem complicações, uso de cálculos simples, a maioria métodos nativos do próprio Java e claro um bom uso de belos gráficos para chamar a atenção do jogador.

CÓDIGO FONTE DO JOGO



```
package jogo;

import java.awt.*;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.awt.event.MouseMotionListener;

import java.awt.image.BufferedImage;

import javax.sound.midi.MidiSystem;

import javax.sound.midi.Sequence;

import javax.sound.midi.Sequencer;

import javax.swing.ImageIcon;

import javax.swing.JFrame

//A CLASSE GAME HERDA AS FUNCIONALIDADES DE JFRAM

@SuppressWarnings("serial"

public class Jogo extends JFrame implements KeyListener, MouseListener,

MouseMotionListener {

BufferedImage backBuffer; //ESSE É O NOSSO BUFFER

int FPS = 30; //ESSA É A TAXA DE ATUALIZAÇÃO DA TELA

int width = Toolkit.getDefaultToolkit().getScreenSize().width;

int height = Toolkit.getDefaultToolkit().getScreenSize().height;

int sizeW = 600; // Largura da janela

int sizeH = 500; // Altura da janela

//AQUI DECLARAMOS O NOSSO MENU COM:

//4 itens, coordenadas x e y = 100, a ativo = true

//agora olhe esse mÃ©todo abaixo cenarios()

Menu menuPrincipal = new Menu(2, 340, 384, true, this.cenario);

ImageIcon menu = new ImageIcon(getClass().getResource("imgs/menu.png"));
```

```

Imagelcon jogoPlay = new Imagelcon(getClass().getResource("imgs/jogo.png"));
Imagelcon fundo = new Imagelcon(getClass().getResource("imgs/abertura.png"));

Imagelcon marinhoJogando = new
Imagelcon(getClass().getResource("imgs/marinhoJogando.png"));

Imagelcon marinhoTriste = new
Imagelcon(getClass().getResource("imgs/marinhoTriste.png"));

Imagelcon marinhoFeliz = new
Imagelcon(getClass().getResource("imgs/marinhoFeliz.png"));

boolean desce = true; //ESSA VARIÁVEL PODERIA ESTAR NA CLASSE Sprite
boolean sobe = false; //ESSA VARIÁVEL PODERIA ESTAR NA CLASSE Sprite
boolean corre = true;

int cenario = 0;

int xMira;

int yMira;

int codigoLixo = 0;

int pontos = 0;

int acertos = 0;

int erros = 0;

int tempo = 59;

int frames = 0;

int frameUp = 1000/FPS;

boolean stopAudioFundo = false;

boolean stopAudioJogo = true;

boolean stopAudioSucesso = true;

boolean stopAudioFracaso = true;

Sprite marinhoPiscando = new Sprite(10, 50, 63);

Sprite marinhoCorrendo = new Sprite(5, -190, 197);

Sprite logoMovimento = new Sprite(4, 238, 63);

Sequencer player; //ESSE É O NOSSO TOCADOR

String musicaFundo = "audio/skrillex-scary_monsters_and_nice_sprites.mid"; //ISSO É
SÓ A URL DA NOSSA MÚSICA

String musicaJogo = "audio/Beethoven-Piano_Sonata_23.mid";

```

```

String musicaSucesso = "audio/sucesso.mid";
String musicaFracaso = "audio/fracaso.mid"
public void tocarMusica(String nome, int repetir){
    try{
        player = MidiSystem.getSequencer(); //INICIA O TOCADOR
        //Sequence musica = MidiSystem.getSequence(new File(nome));//CARREGA
A MÚSICA A SER TOCADA
        Sequence musica =
MidiSystem.getSequence(getClass().getResource(nome));//CARREGA A MÚSICA A
SER TOCADA
        player.open(); //ABRE O TOCADOR
        player.setSequence(musica); //PASSA A MÚSICA QUE VAI SER TOCADA
PARA O TOCADOR
        player.setLoopCount(repetir); //DEFINE QUANTAS VEZES VAI TOCAR A
MÚSICA OBS: 0 = 1 vez; 1 = 2 vezes; 2 = 3 vezes...
        player.start(); //COMEÇA A TOCAR.... CASO QUEIRA PARAR DE TOCAR A
MÚSICA BASTA: player.stop();
    }catch(Exception e){
        System.out.println("Erro ao tocar: "+nome);
    }
}

public void mover(){
    if(desce){ logoMovimento.y += 5; }
    if(sobe){ logoMovimento.y -= 5; }
    if(logoMovimento.y>100){ desce = false; sobe = true; }
    if(logoMovimento.y<20){ desce = true; sobe = false; }
}

public void corre(){
    if(corre){ marinhoCorrendo.x += 5;
        if((marinhoCorrendo.x%2)==0 && marinhoCorrendo.x<300){
            marinhoCorrendo.y -= 1;
        }
    }
}

```

```

    }

    if(marinhoCorrendo.x>610){ corre = false; this.cenario = 1; }
}

//NO NOSSO METODO ATUALIZAR VAMOS CHAMAR OS METODOS
//QUE SERÃO EXECUTADOS O TEMPO INTEIRO...

public void atualizar() {
    mover();
    if(this.cenario == 0){
        corre();
    }
}

//NESSE MÉTODO VAMOS DESENHAR
//FORMAS GEOMETRICAS, IMAGENS E TEXTOS NA TELA E ETC..

public void desenharGraficos() {
    if(this.cenario == 0)
        Graphics g = getGraphics(); //ISSO JÁ ESTAVA AQUI
        Graphics bbg = backBuffer.getGraphics();//ISSO TAMBÉM JÁ ESTAVA
AQUI...

        //AQUI VAMOS MANDAR DESENHAR ALGUNS IMAGENS NA TELA
        bbg.drawImage(fundo.getImage(),0,0,this);//QUI DESENHAMOS O
FUNDO

        //AS DIMENSÕES ORIGINAIS DO FUNDO SÃO: 600X500 QUE É O
TAMANHO DA NOSSA TEL
        bbg.drawImage(marinhoCorrendo.cenas[marinhoCorrendo.cena].getImage(),
marinhoCorrendo.x, marinhoCorrendo.y, this);

        marinhoCorrendo.animarLento();
//=====
=====

        g.drawImage(backBuffer, 0, 0, this);//OBS: ISSO DEVE FICAR
SEMPRE NO FINAL!

    }

    if(this.cenario == 1)

        // COMO AQUI É EXECUTADO O TEMPO TODO, CRIEI UM
CONTROLE DE EXECUÇÃO DE SOMENTE UMA VEZ, ATRAVES DE UMA
VARIÁVEL BOOLEAN

```

```

        if(this.stopAudioFundo){
            this.stopAudioJogo = true;
            player.stop();
            this.stopAudioFundo = false;
            tocarMusica(musicaFundo, 999);
        }
        setTitle("Lixo Certo - APS 4º Semestre UNIP!")
        Graphics g = getGraphics(); //ISSO JÁ ESTAVA AQUI
        Graphics bbg = backBuffer.getGraphics();//ISSO TAMBÉM JÁ ESTAVA
AQUI..

        //AQUI VAMOS MANDAR DESENHAR ALGUNS IMAGENS NA TELA
        bbg.drawImage(menu.getImage(),0,0,this);//QUI DESENHAMOS O
FUNDO

        //AS DIMENSÕES ORIGINAIS DO FUNDO SÃO: 600X500 QUE É O
TAMANHO DA NOSSA
TELAbbg.drawImage(logoMovimento.cenas[logoMovimento.cena].getImage(),
logoMovimento.x, logoMovimento.y, this);

        logoMovimento.animarMaisLento(); //AQUI CHAMEI O MÉTODO
ANIMAR MAIS LENTO
bbg.drawImage(marinhoPiscando.cenas[marinhoPiscando.cena].getImage(),
marinhoPiscando.x, marinhoPiscando.y, this);

        marinhoPiscando.animarMaisLento(); //AQUI CHAMEI O MÉTODO
ANIMAR MAIS LENTO

        this.cenario = menuPrincipal.desenharMenu();//isso desenharÁj nosso
men
//=====
=====

        g.drawImage(backBuffer, 0, 0, this);//OBS: ISSO DEVE FICAR
SEMPRE NO FINAL!

    }

    if(this.cenario == 2){

        // COMO AQUI É EXECUTADO O TEMPO TODO, CRIEI UM
CONTROLE DE EXECUÇÃO DE SOMENTE UMA VEZ, ATRAVES DE UMA
VARIÁVEL BOOLEAN

        if(!stopAudioFundo){

            this.stopAudioFundo = true;

            player.stop();

```

```

        this.stopAudioJogo = false;
        tocarMusica(musicaJogo, 0);
    }
    Graphics g = getGraphics(); //ISSO JÁ ESTAVA AQUI
    Graphics bbg = backBuffer.getGraphics();//ISSO TAMBÉM JÁ ESTAVA
AQUI...
    bbg.drawImage(jogoPlay.getImage(),0,0,this);//QUI DESENHAMOS O
FUNDO
    //AQUI VAMOS MANDAR DESENHAR ALGUNS TEXTOS NA TELA
    if(this.tempo == 0){
        setTitle("Tempo esgotado!!!");
        bbg.setColor(new Color(225,61,0));
    }else{
        setTitle("Rápido Marinho! Você só tem um minuto...");
        bbg.setColor(new Color(255,255,255));
    }
    bbg.setFont(new Font("Arial", Font.PLAIN,30));
    if(this.tempo<10){
        bbg.drawString("00:0"+this.tempo, 265, 78);
    }else{
        bbg.drawString("00:"+this.tempo, 265, 78);
    }
    bbg.setColor(new Color(255,255,255));
    bbg.setFont(new Font("Arial", Font.PLAIN,30));
    bbg.drawString(String.valueOf(this.pontos), 27, 110);
    bbg.setColor(new Color(255,255,255));
    bbg.setFont(new Font("Arial", Font.PLAIN,30));
    bbg.drawString(String.valueOf(this.acertos), 522, 115);
    bbg.setColor(new Color(255,255,255));
    bbg.setFont(new Font("Arial", Font.PLAIN,30));
    bbg.drawString(String.valueOf(this.erros), 522, 152);

```

```

        ImagemCon mira = new
        ImagemCon(getClass().getResource("imgs/lixo"+this.codigoLixo+".png"));

        // Mapeado a área de acerto, somente para controle de posição!!!
        /*

        bbg.setColor(new Color(100,200,100));
        bbg.fillRect(34, 249, 118, 26);
        bbg.setColor(new Color(100,200,100));
        bbg.fillRect(176, 233, 118, 26);
        bbg.setColor(new Color(100,200,100));
        bbg.fillRect(318, 224, 118, 26);
        bbg.setColor(new Color(100,200,100));
        bbg.fillRect(462, 240, 118, 26);

        */

        if(this.tempo>0){
            this.frames += 1;
            if(this.frameUp == this.frames){
                this.tempo -= 1;
                this.frameUp += 1000/FPS;
            }

            bbg.drawImage(marinhoJogando.getImage(),219,310,this);//AQUI
            DESENHAMOS O MARINHO

            //AQUI DESENHAMOS A MIRA, O -25 Ã‰ PARA FICAR
            CENTRALIZADA COM O CURSOR

            bbg.drawImage(mira.getImage(), this.xMira-25, this.yMira-25,
            this);

        }else{
            if(this.pontos>0){

                if(this.stopAudioSucesso){
                    this.stopAudioJogo = true;
                    player.stop();
                }
            }
        }
    }
}

```

```

        this.stopAudioSucesso = false;

        tocarMusica(musicaSucesso, 0);

    }

    bbg.drawImage(marinhoFeliz.getImage(),219,310,this);//AQUI DESENHAMOS O
    MARINHO Feliz com sua pontuação

        }else{

            if(this.stopAudioFracaso){

                this.stopAudioJogo = true;

                player.stop();

                this.stopAudioFracaso = false;

                tocarMusica(musicaFracaso, 0);

            }

            bbg.drawImage(marinhoTriste.getImage(),219,310,this);//AQUI
            DESENHAMOS O MARINHO Triste por não ter pontuado

        }

    }

//=====
=====

        g.drawImage(backBuffer, 0, 0, this);//OBS: ISSO DEVE FICAR
        SEMPRE NO FINAL!

    }

    if(this.cenario == 9){

        System.exit(0);//esse comando fecha o nosso game!

    }

}

//AQUI É AONDE A MAGICA DA TROCA DE LIXO NA MIRA ACONTECE, USEI UM
CÓGIGO DE GERAÇÃO DE VALOR ALEATÓRIA (ok, ok, Copei pronto do meu
professor... )

public void trocaLixo(){

    int[] digitos = {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1};

    int e = 0;

    int n = 0;

    while (e < 10) {

```



```

        n = (int)(Math rint(Math.random() * 10) - 1);
        for    (int j = 0;j < 10;j++) {
            if    (digitos[j] == n) {
                break;
            }
            if    (digitos[j] == -1) {
                digitos[j] = n;
                e++;
                break;
            }
        }
    }
    this.codigoLixo = digitos[0];
}

//ESSE É O NOSSO MÉTODO INICIALIZAR

//AQUI VAMOS INICIALIZAR ALGUMAS CONFIGURAÇÃO DO frame E OUTRAS CONFIGURAÇÕES

public void inicializar() {
    setTitle("Corra Marinho, o jogo vai começar!");//SETANDO O TITULO DA JANELA

    setSize(sizeW,sizeH); // atribui ao tamanho da janela da janela

    setLocation(width/2-(sizeW/2), height/2-(sizeH/2)); //posiciona (X,Y) a janela

    setResizable(false);//TIRANDO A PERMISSÃO DO USUÁRIO REDIMENSIONAR A JANELA

    setDefaultCloseOperation(EXIT_ON_CLOSE);//QUANDO FECHARMOS O frame A APLICAÇÃO PARA DE EXECUTAR

    setLayout(null);//COM ISSO PODEREMOS DEFINIAR COORDENADA E DIMESÕES DE ELMENTOS DE FORMULARIO NO NOSSO FRAME

    setVisible(true);//MUDANDO A VISIBILIDADE DO frame PARA TRUE, ASSIM ELE APARECERÁ

    setResizable(false);

    backBuffer = new BufferedImage(sizeW, sizeH,
    BufferedImage.TYPE_INT_RGB);//CRIANDO O NOSSO BUFFER DE IMAGEM

```

```

//AQUI ESTAMOS ADICIONANDO UM ESCUTADOR DE TECLAS
addKeyListener(this);

//AQUI ESTAMOS ADICIONANDO OS ESCUTADORES DE EVENTOS DO
MOUSE

addMouseListener(this);

addMouseMotionListener(this);

    //aqui definimos o texto de cada item do nosso menu
    menuPrincipal.itens[0] = "Jogar";
    menuPrincipal.itens[1] = " Sair";

    //aqui fazemos o método desenhaMenu() que fica lá em Menu.java
    //desenhar no nosso buffer

    //.. agora para finalizar observe o método de evento keyPressed()
mais abaixo...

    menuPrincipal.bbg = backBuffer.getGraphics();

    //AQUI CARREGAMOS AS IMAGENS DE NOSSA SPRITE!!!!!!

    //PARA O VETOR DE Imagem[] !!!

    marinhoPiscando.cenas[0] = new
    Imagem(getClass().getResource("imgs/marinhoPiscando1.png"));

    marinhoPiscando.cenas[1] = new
    Imagem(getClass().getResource("imgs/marinhoPiscando2.png"));

    marinhoPiscando.cenas[2] = new
    Imagem(getClass().getResource("imgs/marinhoPiscando3.png"));

    marinhoPiscando.cenas[3] = new
    Imagem(getClass().getResource("imgs/marinhoPiscando4.png"));

    marinhoPiscando.cenas[4] = new
    Imagem(getClass().getResource("imgs/marinhoPiscando1.png"));

    marinhoPiscando.cenas[5] = new
    Imagem(getClass().getResource("imgs/marinhoPiscando1.png"));

    marinhoPiscando.cenas[6] = new
    Imagem(getClass().getResource("imgs/marinhoPiscando1.png"));

    marinhoPiscando.cenas[7] = new
    Imagem(getClass().getResource("imgs/marinhoPiscando1.png"));

    marinhoPiscando.cenas[8] = new
    Imagem(getClass().getResource("imgs/marinhoPiscando1.png"));

```

```

        marinhoPiscando.cenas[9] = new
        ImageIcon(getClass().getResource("imgs/marinhoPiscando1.png"));

        logoMovimento.cenas[0] = new
        ImageIcon(getClass().getResource("imgs/logo1.png"));

        logoMovimento.cenas[1] = new
        ImageIcon(getClass().getResource("imgs/logo2.png"));

        logoMovimento.cenas[2] = new
        ImageIcon(getClass().getResource("imgs/logo3.png"));

        logoMovimento.cenas[3] = new
        ImageIcon(getClass().getResource("imgs/logo4.png"));

        marinhoCorrendo.cenas[0] = new
        ImageIcon(getClass().getResource("imgs/marinhoCorrendo1.png"));

        marinhoCorrendo.cenas[1] = new
        ImageIcon(getClass().getResource("imgs/marinhoCorrendo2.png"));

        marinhoCorrendo.cenas[2] = new
        ImageIcon(getClass().getResource("imgs/marinhoCorrendo3.png"));

        marinhoCorrendo.cenas[3] = new
        ImageIcon(getClass().getResource("imgs/marinhoCorrendo4.png"));

        marinhoCorrendo.cenas[4] = new
        ImageIcon(getClass().getResource("imgs/marinhoCorrendo5.png"));

//=====
=====

        //===== É AQUI QUE EU TO CHAMANDO O MÉTODO
        tocarMusica()=====

        tocarMusica(musicaFundo, 999); //CHAMAMOS O TOCADOR DENTRO DO
        MÉTODO inicializar()

        //ENTÃO AO INICAR O GAME, ESSA SERÁ A NOSSA MÚSICA DE FUNDO

//=====
=====

    }

    //AQUI É O NOSSO MÉTODO RUN()

    //NELE TEMOS O NOSSO GAME LOOP (UM LOOP INFINITO)

```

```

public void run() {

    inicializar();//AQUI CHAMAMOS O METODO INICIALIZAR SOMENTE UMA VEZ,
    POIS ELE ESTÁ FORA DO NOSSO LOOP

    trocaLixo();

    while (true) { //AQUI É O NOSSO LOOP INFINITO

        atualizar();//CHAMAMOS O METODO ATUALIZAR O TEMPO INTEIRO

        desenharGraficos();//ATUALIZAREMOS O GRÁFICO QUE APARECE NA TELA O
        TEMPO INTEIRO

        try {

            Thread.sleep(1000/FPS); //TAXA DE ATUALIZAÇÃO NA TELA, FUNCIONA COMO
            UM DELAY

        } catch (Exception e) {

            System.out.println("Thread interrompida!");

        }

    }

}

//AQUI É O NOSSO MÉTODO PRINCIPAL

public static void main(String[] args) {

    Jogo game = new Jogo();//CRIAMOS UM OBJETO A PARTIR DESSA PROPRIA
    CLASSE

    game.run();//CHAMAMOS O METODO RUN(), O MÉTODO RUN() EXECUTA O
    INICIALIZAR(), ATUALIZAR() E DESENHARGRAFICOS()

}

public void keyPressed(KeyEvent e) {

    if(this.cenario == 1){

        this.pontos = 0;

        this.acertos = 0;

        this.erros = 0;

        this.tempo = 59;

        trocaLixo();

    }

    //aqui, chamamos os métodos que irão controlar o menu pelo teclado

```

```

        this.cenario = menuPrincipal.controlar(e); //esse controla o menu

        this.cenario = menuPrincipal.voltarAoMenu(e); //esse faz voltar para o
menu quando pressionarmos "Esc"
    }

    public void keyReleased(KeyEvent e) {
    }

    public void keyTyped(KeyEvent e) {
    }

    public void mouseClicked(MouseEvent e) {

        // Versão do menu clicavel, mas não limpa a pontuação, não vou perder
tempo com isso não.....

        /*if(this.cenario == 1){

            if(this.xMira >= 247 && this.xMira <= 493 && this.yMira >= 356
&& this.yMira <= 395){

                this.cenario = 2;

            }

            if(this.xMira >= 247 && this.xMira <= 493 && this.yMira >= 410
&& this.yMira <= 449){

                int resp = JOptionPane.showConfirmDialog(null, "Deseja
fazer sair do jogo?", "Vai abandonar o Marinho?", JOptionPane.YES_NO_OPTION, -1);

                if(resp == 0){

                    this.cenario = 9;

                }

            }

        }*/

        if(this.cenario == 2 && this.tempo > 0){

            boolean acerto = false;

            boolean papel = false;

            boolean metal = false;

            boolean plastico = false;

            boolean vidro = false;

```

```

        // Verifica em qual lixeira ele acertou o lixo...
        if(this.xMira >= 34 && this.xMira <= 152 && this.yMira >= 249 &&
this.yMira <= 275){

            // Acertou a lixeira de PAPEL

            papel = true;

        }else{

            if(this.xMira >= 176 && this.xMira <= 294 && this.yMira >=
233 && this.yMira <= 259){

                // Acertou a lixeira de METAL

                metal = true;

            }else{

                if(this.xMira >= 318 && this.xMira <= 436 &&
this.yMira >= 224 && this.yMira <= 250){

                    // Acertou a lixeira de PLÁSTICO

                    plastico = true;

                }else{

                    if(this.xMira >= 462 && this.xMira <= 580
&& this.yMira >= 240 && this.yMira <= 266){

                        // Acertou a lixeira de VIDRO

                        vidro = true;

                    }

                }

            }

        }

        // Caso tenha acertado alguma lixeira, verifica se o lixo jogado
esta correto e pontua...

        if(papel || metal || plastico || vidro){

            if((this.codigoLixo == 2 || this.codigoLixo == 3 ||
this.codigoLixo == 4) && papel){

                acerto = true;

            }

            if((this.codigoLixo == 8 || this.codigoLixo == 9) && metal){

                acerto = true;

```

```

        }
        if((this.codigoLixo == 0 || this.codigoLixo == 1) &&
plastico){
            acerto = true;
        }
        if((this.codigoLixo == 5 || this.codigoLixo == 6 ||
this.codigoLixo == 7) && vidro){
            acerto = true;
        }
        if(acerto){
            this.acertos += 1;
            this.pontos += 5;
        }else{
            this.erros += 1;
            this.pontos -= 5;
        }
    }else{
        this.erros += 1;
        this.pontos -= 10;
    }
    trocaLixo(); // Acertando ou Errado o lixo e trocado...
}

}

public void mouseEntered(MouseEvent e) {
}

public void mouseExited(MouseEvent e) {
}

public void mousePressed(MouseEvent e) {
}

public void mouseReleased(MouseEvent e) {
}

```

```
public void mouseDragged(MouseEvent e) {  
    }  
public void mouseMoved(MouseEvent e) {  
    //AQUI X e Y DA MIRA RECEBE AS COORDENADAS DO CURSOR  
    this.xMira = e.getX();  
    this.yMira = e.getY();  
}  
//-----  
} // FIM DO CÓDIGO FONTE
```


FONTES/BIBLIOGRAFIA

Game:

<http://www.lwjgl.org/>

<http://docs.oracle.com/javase/7/docs/api/index-files/index-1.html>

<http://seumestredaweb.blogspot.com.br/2012/06/java-games-2d-tutor-1.html>

<http://desenvolvimentodejogos.wikidot.com/>

<http://beatnupproject.nostalgia.eng.br/como-criar-um-enredo/>

<http://www.selectgame.com.br/iniciando-em-desenvolvimento-de-jogos-parte-1/>

Áudio:

<http://www.soundeffectpack.com/>

<http://www.soundjay.com/>

<http://www.free-midi.org/>

Livros

MANUAL DE PRODUÇÃO DE JOGOS DIGITAIS - Heather M. Chandler - 1º ed
- 2012



Imagens do Jogo



Lixo Certo



Que pena!



