



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

Jean Cláudio Kaneko Lobo

**Desenvolvimento de Jogos para Ensino de Programação
usando Java**

Assis
2013

Jean Cláudio Kaneko Lobo

Desenvolvimento de Jogos para Ensino de Programação usando Java

Trabalho de Conclusão de Curso apresentado
ao Instituto Municipal de Ensino Superior de
Ensino de Assis, como requisito do Curso de
Graduação.

ORIENTADOR: Felipe Alexandre Cardoso Pazinatto

ÁREA DE CONCENTRAÇÃO: Informática

Assis
2013

FICHA CATALOGRÁFICA

LOBO, Jean Cláudio Kaneko

Desenvolvimento de Jogos para Ensino de Programação usando Java / Jean Cláudio Kaneko Lobo. Fundação Educacional do Município de Assis – FEMA – Assis, 2013.

Numero de paginas: 53

Orientador: Felipe Alexandre Cardoso Pazinato.

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA.

1. Jogo. 2. Educacional.

CCD: 001.6
Biblioteca da FEMA

Desenvolvimento de Jogos para Ensino de Programação usando Java

Jean Cláudio Kaneko Lobo

Trabalho de Conclusão de Curso apresentado
ao Instituto Municipal de Ensino Superior de
Ensino de Assis, como requisito do Curso de
Graduação.

Orientador: Felipe Alexandre Cardoso Pazinato

Analizador: Luiz Carlos Begosso

Assis
2013

Dedicatória

Dedico este trabalho aos meus pais, que me apoiaram durante toda a minha vida.

Agradecimentos

Ao meu orientador Felipe Alexandre Cardoso Pazinato, pela ajuda, orientação e incentivo para desenvolver um trabalho com o tema de jogos.

Aos meus pais que nunca me deixaram desistir.

Resumo

Entender como um algoritmo funciona é a base para aprender a programar, porém por ser diferente de qualquer outra disciplina, existe uma dificuldade por parte de alguns alunos para alcançarem o objetivo didático, que é o aprendizado de algoritmos. É objetivo deste projeto criar um serious game, servindo como ferramenta didática para o ensino de algoritmos. Jogos sempre tem um apelo mais lúdico, quebrando um pouco o formalismo de uma aula.

A proposta desse trabalho é desenvolver uma ferramenta para auxiliar a apresentação dos algoritmos mais simples aos alunos, e também apresentar o uso de fluxogramas para representar esses algoritmos.

Palavras-chave: Jogo, Educacional.

ABSTRACT

Understand how an algorithm works is the base to learn to program, however this is very different from the other disciplines, exist the difficulty by the students to reach the goal, which is learning the algorithms. The objective of this project is developing a game, who will work like a tool to teach algorithm. The games have an appeal more playful, breaking the formalism of the classes.

The proposal of this paper is develop a tool to help the introduction of simple algorithms to the students, and introduce the use of flowcharts to represent those algorithms.

Keywords: Game, Educational

Lista de Ilustrações

Figura 1: Imagem da aplicação tênis para dois.....	14
Figura 2: Jogo Spacewar.....	15
Figura 3: Jogo 3D Driving Academy.....	17
Figura 4: Jogo Lemonade Tycoon 2.....	18
Figura 5: Jogo Xilo.....	21
Figura 6: Jogo O código secreto.....	22
Figura 7: Jogo A questão.....	22
Figura 8: Jogo ProGame.....	23
Figura 9: Fluxograma simples.....	25
Figura 10: Estruturas de sequencia de ações, seleção e loop.....	26
Figura 11: Exemplos de primitivas geométricas do Java 2D.....	28
Figura 12: Classes do Java Swing.....	31
Figura 13: Exemplo de utilização de Java Swing.....	32
Figura 14: Resultado do código.....	32
Figura 15: Diagrama de classes do projeto.....	33
Figura 16: Diagrama de estados.....	34
Figura 17: Diagrama de casos de uso.....	34
Figura 18: Tabuleiro do jogo.....	42

Sumário

1. INTRODUÇÃO	12
1.1. OBJETIVO DO TRABALHO.....	12
1.2. PÚBLICO ALVO.....	13
1.3. JUSTIFICATIVA.....	13
1.4. METODOLOGIA DE DESENVOLVIMENTO.....	13
2. JOGOS ELETRÔNICOS	14
2.1. HISTÓRIA DOS JOGOS.....	14
2.2. SERIOUS GAMES.....	16
2.3. UTILIZAÇÕES DE JOGOS ELETRÔNICOS EM DIVERSAS ÁREAS.....	19
2.3.1. Jogos Eletrônicos na Medicina	19
2.3.2. Jogos Didáticos	20
2.3.2.1. Jogos Didáticos na Área de Informática.....	21
3. LÓGICA DA PROGRAMAÇÃO	24
3.1 Fluxograma.....	24
4. JAVA	27
4.1. HISTÓRIA DO JAVA.....	27
4.2. Java 2D.....	28
4.3. JAVA SWING.....	30
5. ANÁLISE DO PROJETO	33
5.1. DIAGRAMA DE CLASSES.....	33
5.2. DIAGRAMA DE ESTADOS.....	34
5.3. CASOS DE USO.....	34
5.3.1. Descrição dos Casos de Uso	35
5.3.1.1. Mover Peças.....	35

5.3.1.2. Requisitar Correção.....	36
5.3.1.3. Iniciar Novo Problema.....	36
5.3.1.4. Limpar Tabuleiro.....	37
5.3.1.5. Corrigir Problema.....	38
5.3.1.6. Gerenciar Tempo.....	39
6. RESULTADO.....	40
6.1. CONSTRUTOR DA CLASSE JOGO.....	40
6.2. MÉTODO ACTIONPERFORMED.....	42
6.3. MÉTODO DE TEMPO.....	44
6.4. MÉTODO DE INICIAÇÃO DE UMA NOVA FASE.....	45
6.5. A CLASSE DE CORREÇÃO.....	46
7. CONCLUSÃO.....	47
8. REFERÊNCIAS BIBLIOGRÁFICAS.....	48
ANEXO.....	50

1. INTRODUÇÃO

Em todas as áreas do conhecimento são utilizados jogos educacionais para auxiliar ou mesmo motivar os alunos numa determinada matéria. Na área de programação não é diferente, porém como nas outras áreas do conhecimento ainda ha uma falta de incentivo para os jogos didáticos.

Mesmo com a falta de incentivo ainda é de grande ajuda uma ferramenta parra auxiliar o ensino, por isso foi desenvolvido um jogo didático, onde foram apresentados problemas por meio de desenhos de fluxogramas e o aluno deverá resolvê-los ou completá-los.

Esse jogo foi desenvolvido para ensinar de maneira mais agradável o pensamento lógico necessário para se aprender a programar, pois mesmo aqueles que não se tornarão programadores precisarão de um pensamento lógico desenvolvido para melhor atuar em suas respectivas áreas.

1.1. OBJETIVO DO TRABALHO

O objetivo deste trabalho é desenvolver um jogo para ensinar, de maneira lúdica, as estruturas lógicas que são usadas para representar a elucidação de alguns problemas em programação de computadores (Introdução à Programação).

Atualmente enfrenta-se na área da educação grandes desafios. Percebe-se que os atuais alunos de muitos cursos, possuem certa deficiência de atenção, devido ao uso indiscriminado de tecnologias que exigem atenção por curtos espaços de tempo, e imediata. Devem-se buscar as tecnologias possíveis para cativar estes alunos, como a utilização do lúdico no ensino, na forma de jogos para computador.

1.2. PÚBLICO ALVO

O público alvo deste trabalho são os alunos que estão começando a aprender programação e sua lógica, como uma ferramenta de auxílio ao aprendizado.

1.3. JUSTIFICATIVA

Este trabalho se justifica pela necessidade de produzir ferramentas que apoiem o ensino introdutório de programação de computadores e que utilizem do lúdico para envolver os alunos no aprendizado. Qualquer ferramenta que facilite o ensino é muito bem vinda.

1.4. METODOLOGIA DE DESENVOLVIMENTO

Serão usadas para desenvolvimento do sistema a linguagem Java, com a IDE NetBeans, utilizando a tecnologia Swing e alguns conceitos do Java 2D.

Não será utilizado nenhum banco de dados, pois a aplicação não necessitará armazenar dados dos usuários.

2. JOGOS ELETRÔNICOS

2.1 HISTÓRIA DOS JOGOS

Segundo Gonzalez Clua e Bittencourt[1], o primeiro uso de imagens eletrônicas manipuladas por pessoas com o objetivo de entretenimento foi a aplicação do professor William Higinbotham em 1958. Utilizando-se de um osciloscópio e um computador desenvolveu uma aplicação chamada de tênis para dois, que consistia em uma simulação de uma quadra de tênis onde cada jogador era representado por um traço vertical cada um de uma cor e a bola era representada por uma bolinha branca. Esta aplicação funcionava genericamente da mesma maneira que um jogo de tênis convencional, com cada jogador tendo que impedir que a bola cair no seu campo e rebater-la de volta para o adversário. A Figura 1 mostra a aplicação tênis para dois em um osciloscópio.

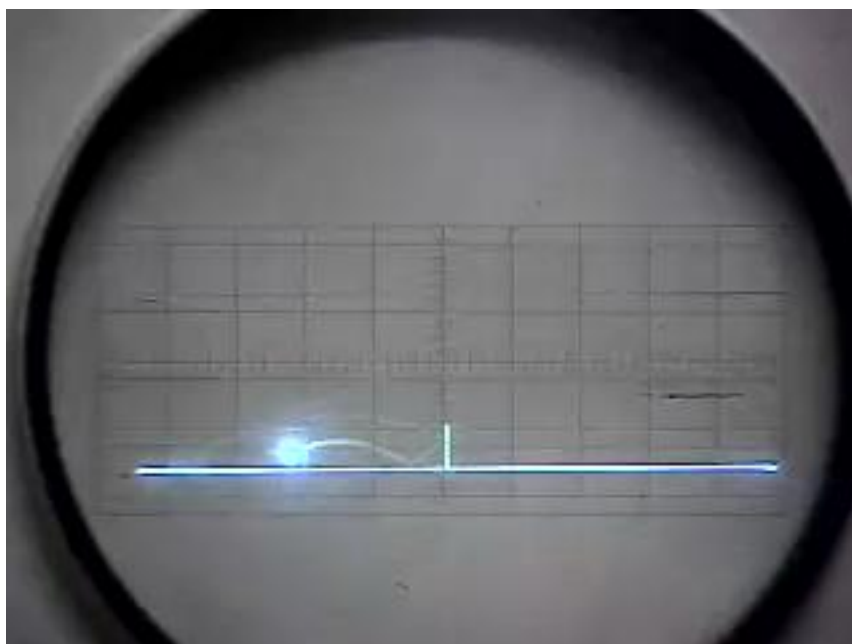


Figura 1: Imagem da aplicação tênis para dois. Fonte [2].

Porém, tênis para dois ainda não pode ser considerado o primeiro jogo eletrônico pois este nunca foi patenteado ou vendido pelo seu criador, e era usado apenas para entreter alguns visitantes do seu laboratório.

Considerado o primeiro jogo eletrônico, *Space War!* foi desenvolvido por um grupo de estudantes do MIT, onde a maior parte foi feita por Steve Russel. Este jogo foi testado pela primeira vez em 1961, porém foi oficialmente finalizado em 1962 tendo apenas 2KB de tamanho [1]. O jogo era basicamente uma batalha entre duas naves no espaço onde cada jogador controlava uma nave e deveria destruir a outra atirando nela, não havia efeitos sonoros por conta das limitações tecnológicas da época e também não foi desenvolvida uma inteligência artificial pelo mesmo motivo, por isso só era possível jogar se houvesse dois jogadores. Uma curiosidade sobre esse jogo era o fato de seu código ser aberto. A figura 2 mostra uma imagem do jogo.

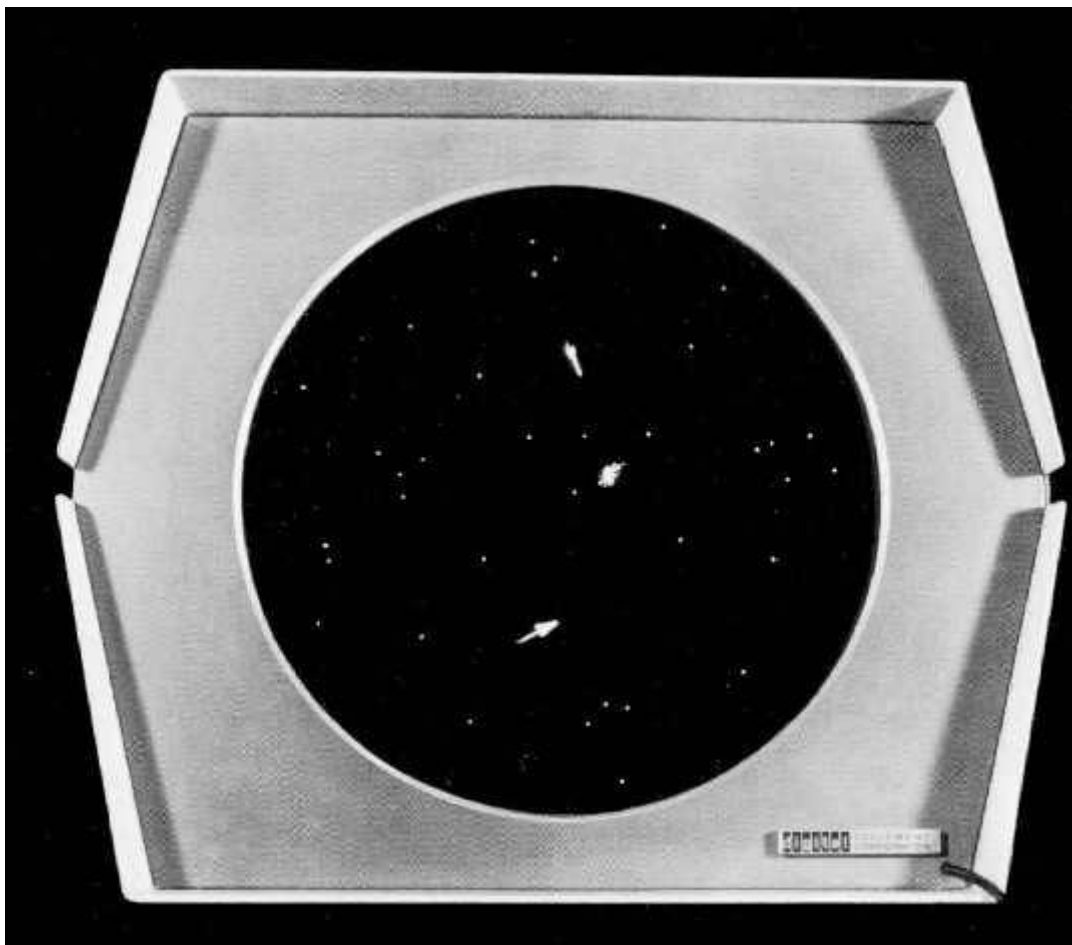


Figura 2: Jogo Space War. Fonte [3].

Apesar de Space War ser uma revolução de sua época seus criadores acabaram não ganhando dinheiro com ele, pois a popularidade do jogo ficou limitada aqueles que viviam da computação em 1962.

Porém Space War despertou a atenção de um estudante da Universidade de Utah, chamado Nolan Bushnell, que acredita que os videogames poderiam se tornar um grande mercado e se espalhar pelo seu país [3].

No ano de 1972 Nolan Bushnell junto com um amigo fundou a maior empresa de jogos eletrônicos da época, a Atari. No mesmo ano o engenheiro Ralf Baer desenvolveu o primeiro console de videogame, solicitado pelo exército para treinar soldados, chamado de Odyssey.

O ano de 1984 foi marcado pelo fim da empresa Atari e junto com o seu fim uma grande crise se instaurou no mercado ocidental de consoles. Os motivos do fim da empresa e da crise foram: defasagem tecnológica, crise de conteúdo e início da informática doméstica [4].

Em 1985 são lançados os consoles das empresas japonesas Nintendo e SEGA. Em 1994 a Sony, outra empresa japonesa que atuava na produção de outros produtos eletrônicos, entra para o mercado de consoles competindo com as outras empresas também japonesas. A empresa SEGA em 1998, após o lançamento de um console excepcional que infelizmente não atraiu público o suficiente, desiste do mercado de consoles e decide produzir apenas softwares. No ano de 2001 a gigante Microsoft entra neste mercado com o seu console, e com sua entrada tem-se as três empresas que atualmente dominam o mercado de consoles: Nintendo, Sony e Microsoft[1].

2.2 SERIOUS GAMES

Alguns jogos não são feitos com o intuito de entreter e sim de passar um conhecimento ou até mesmo dar algum treinamento por meio de simulações. Esses tipos de jogos são denominados jogos sérios ou *serious games*.

Esse tipo de jogo normalmente utiliza simulações de situações diárias com o intuito de proporcionar treinamento profissional, conscientização e treinamento para como se portar em situações críticas. Os jogos conhecidos como *serious games* utilizam a abordagem lúdica das indústrias de jogos para tornar suas simulações mais divertidas e atraentes, desta forma tornando a absorção de conceitos mais proveitosa e desenvolvendo as habilidades psicomotoras. Por isso, o termo *serious games* é usado para classificar os jogos com um propósito específico de treinamento ou aprendizado [13].

Um exemplo deste tipo de jogo é o 3D Driving Academy que teve como objetivo principal recriar alguns distritos de cidades da Europa, para criar um simulador onde seria possível dirigir sobre as diferentes leis de trânsito de alguns países. Esse jogo utiliza inteligência artificial para controlar os outros carros, pessoas, semáforos e motociclistas nas ruas da cidade escolhida. A figura 3 mostra uma imagem do jogo.



Figura 3: Jogo 3D Driving Academy[13].

Outro exemplo dessa categoria de jogo é direcionado para a administração de negócios, mesmo tendo um tema incomum. O jogo, *Lemonade Tycoon 2*, simula a administração de uma empresa de vendas ambulante de limonada, neste jogo é necessário administrar a matéria prima, localização das barracas, os preços e melhorar sua receita. O jogador pode comparar seu desempenho com outros jogadores via internet [13]. A figura 4 mostra uma imagem do jogo.



Figura 4: Jogo Lemonade Tycoon 2 [13].

2.3 UTILIZAÇÕES DE JOGOS ELETRÔNICOS EM DIVERSAS ÁREAS

2.3.1 JOGOS ELETRÔNICOS NA MEDICINA

Os jogos eletrônicos que normalmente são vistos como apenas uma brincadeira ou uma espécie de atividade lúdica usada para se distrair vem sendo amplamente usado na medicina, com diversos objetivos, como citado na reportagem do site da revista Isto É [5].

Podem-se citar duas iniciativas usando jogos para ajudar os pacientes:

O primeiro foi feito no Hospital Infantil Nationwide, nos Estados Unidos, que basicamente se tratava de usar um capacete de realidade virtual para emergir o paciente em um mundo virtual deste modo a criança fica menos ansiosa na hora de trocar os curativos. Este recurso é usado nas vítimas de queimaduras que, antes da introdução do jogo, eram distraídas usando músicas e livros na hora de trocar os curativos [5].

A segunda iniciativa está sendo usada em Uganda com portadores de paralisia cerebral causado por malária. Os pesquisadores da universidade de Michigan, também dos Estados Unidos, estão utilizando um jogo para melhorar as capacidades cognitivas, coordenação motora e atenção das vítimas de paralisia cerebral. Um dos responsáveis do projeto, o doutor Michael Bovin afirma que o tratamento com o jogo, além desses benefícios, ainda melhora a autoestima e autocontrole do paciente [5].

Assim como nessas áreas da medicina os jogos eletrônicos vêm contribuindo com a fisioterapia. O console da empresa Nintendo chamado Wii lançado em 2006 revolucionou com o uso de movimentos do próprio jogador interagindo com o jogo. Esta nova tecnologia de movimento foi usada pela fisioterapia, como dito na reportagem de 2011 do site da Folha de São Paulo [6]:

“... usando o console fisioterapeutas usam jogos para complementar o tratamento de pacientes que tiveram derrame. Além do fato do paciente se divertir ao mesmo tempo em que faz sua terapia os jogos motivam os pacientes e os fazem querer voltar logo.”

2.3.2 JOGOS DIDÁTICOS

Além dos benefícios para alguns campos da medicina os jogos podem também auxiliar crianças e jovens no aprendizado. Os jogos utilizam uma linguagem mais dinâmica que desafiam e recompensam os jogadores, além de utilizar uma atividade lúdica para expor o conhecimento para o jogador.

É possível utilizar os jogos didáticos no ensino de qualquer área como, por exemplo, o jogo Xilo que foi desenvolvido pelos alunos do curso de Jogos Digitais da Facisa em Campina Grande. Este jogo mostra a cultura e folclore do nordeste do Brasil por meio de um jogo em plataforma em 2D e com o visual baseado em xilogravuras [7]. Também é possível encontrar jogos para quase todas as áreas do conhecimento espalhados pela internet. A figura 5 mostra a tela de seleção de fases do jogo.



Figura 5: Jogo Xilo. Fonte [7].

2.3.2.1 JOGOS DIDÁTICOS NA ÁREA DE INFORMÁTICA

Como nas outras áreas do conhecimento e possível utilizar-se de jogos para ensinar o conteúdo aos alunos, e na área de informática não é diferente porém como esta área é muito vasta existem jogos voltados para áreas diversas da informática como o ensino da lógica.

E neste campo temos jogos como “O código secreto” que consiste em o jogador descobrir a combinação da senha que abre a porta para o detetive entrar na casa, e para isso ele deve informar uma combinação que é formada pela soma das linhas, colunas e diagonais. Para que a resposta seja correta todas as somas devem ser iguais a 15 e todos os números de 1 a 9 devem ser usados sem repetições [8]. A figura 6 mostra o jogo “O código secreto”.



Figura 6: Jogo O código secreto. Fonte [8]

Outro exemplo de jogo para desenvolver a lógica é o jogo “A questão” onde o jogador deve acertar uma sequência numérica de 7 números, os números podem ser repetidos, e com esses números deve ser possível representar todos os números de 1 até 24. Os números da sequência podem ser somados ou usados individualmente para cumprir o problema [8]. A figura 7 mostra o jogo “A questão”.

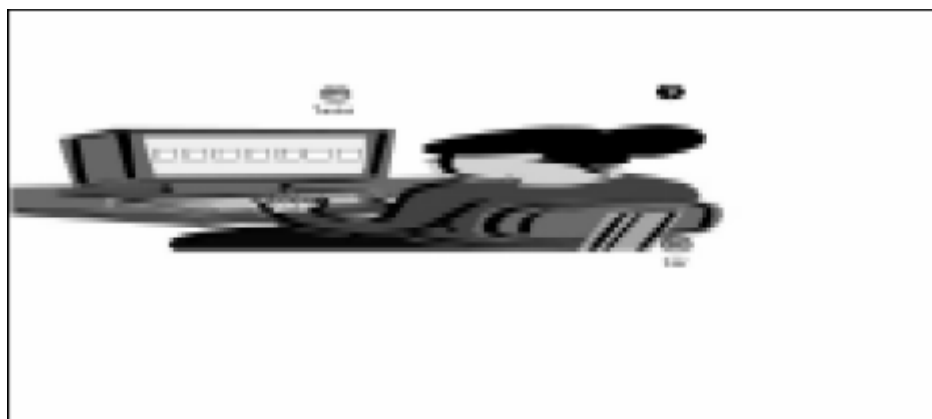


Figura 7: Jogo A questão. Fonte [8]

Um aplicativo para o ensino de programação é o jogo ProGame que passa conceitos de programação de forma lúdica além de desafiar o jogador ao longo das fases para que este pratique seus conhecimentos de programação.

“O objetivo é conduzir o aluno a um ambiente em que possa desenvolver suas habilidades em solução de problemas, estando esses explícitos ou não.” [9]. A figura 8 mostra o jogo ProGame.

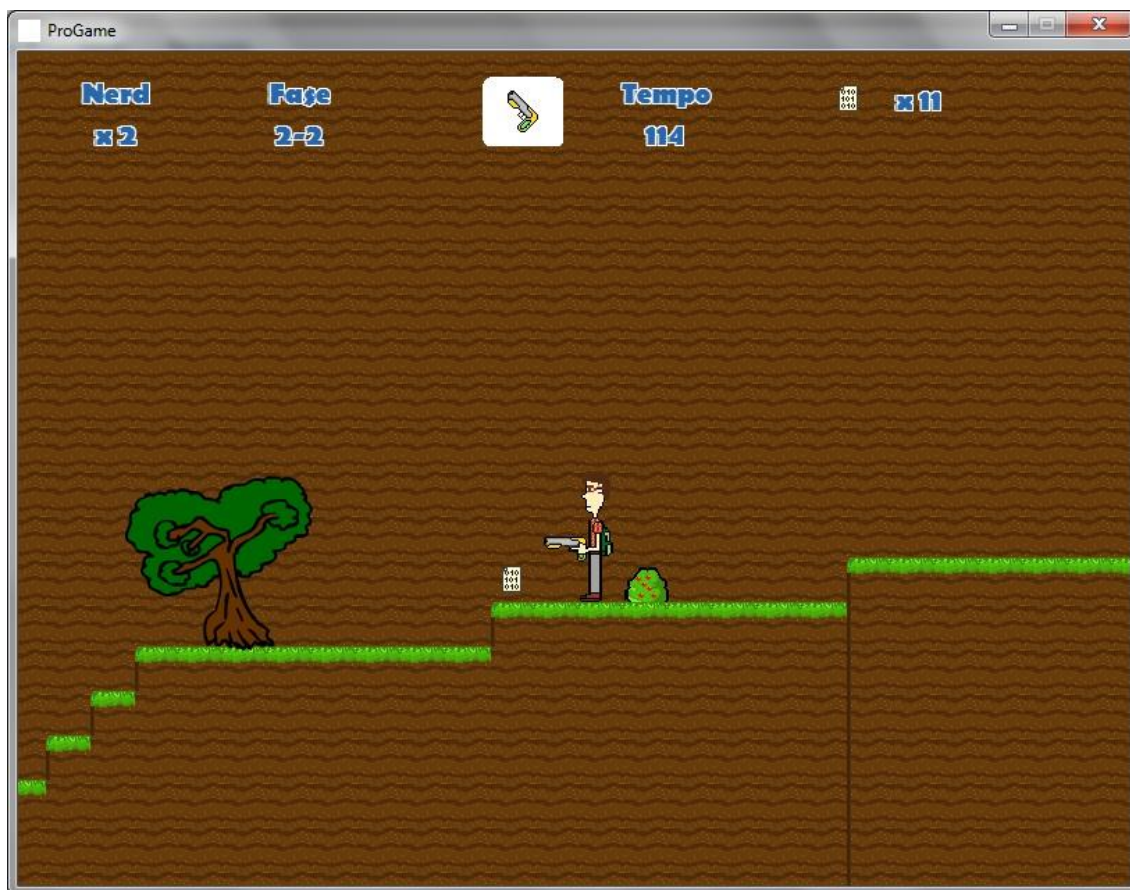


Figura 8: Jogo ProGame. Fonte [9].

3. LÓGICA DA PROGRAMAÇÃO

Independente da linguagem que será utilizada para desenvolver os programas, o programador deve ter planejado o seu projeto antes de começar a digitar o código. Para planejar um programa segundo Joyce Farrel [14] é preciso entender o problema e planejar a lógica antes de começar a codificar.

Para se planejar a lógica é necessário organizar as instruções que serão utilizadas para resolver o problema que foi entendido anteriormente. Para melhor organizar a lógica pode-se usar um fluxograma.

3.1. FLUXOGRAMA

Um fluxograma é um diagrama que representa os passos necessários para se resolver um problema. A figura 9 mostra um fluxograma simples.

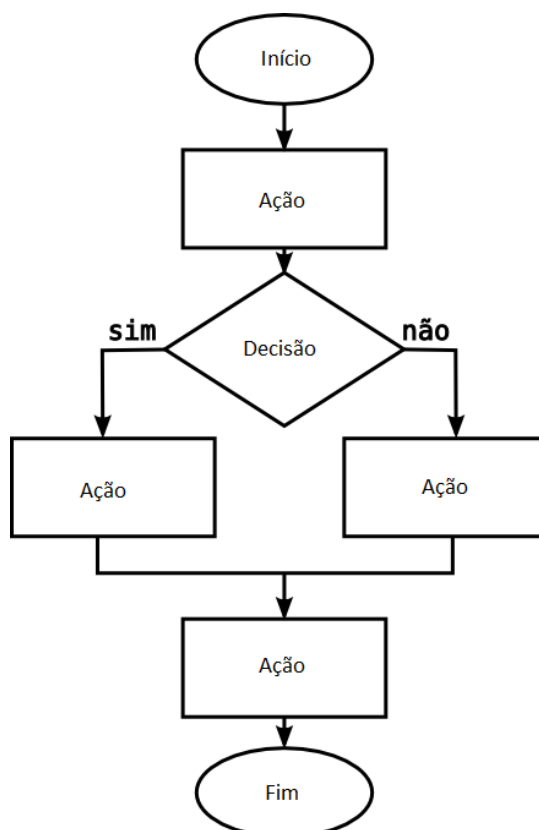


Figura 9: Fluxograma simples.

Na figura9, os retângulos com o termo “Ação” representam alguma tarefa. Por exemplo, pode ser a soma de dois valores, e este campo é representado por um retângulo. A Decisão que sempre é representada por um hexágono e dentro de um programa essa decisão se refere a testar uma variável e tomar uma decisão de acordo com o resultado.

Usando esses dois componentes é possível construir três estruturas básicas que possibilitam representar um código. Essas estruturas são: sequência de ações, seleção e loop. A figura 10 mostra essas estruturas.

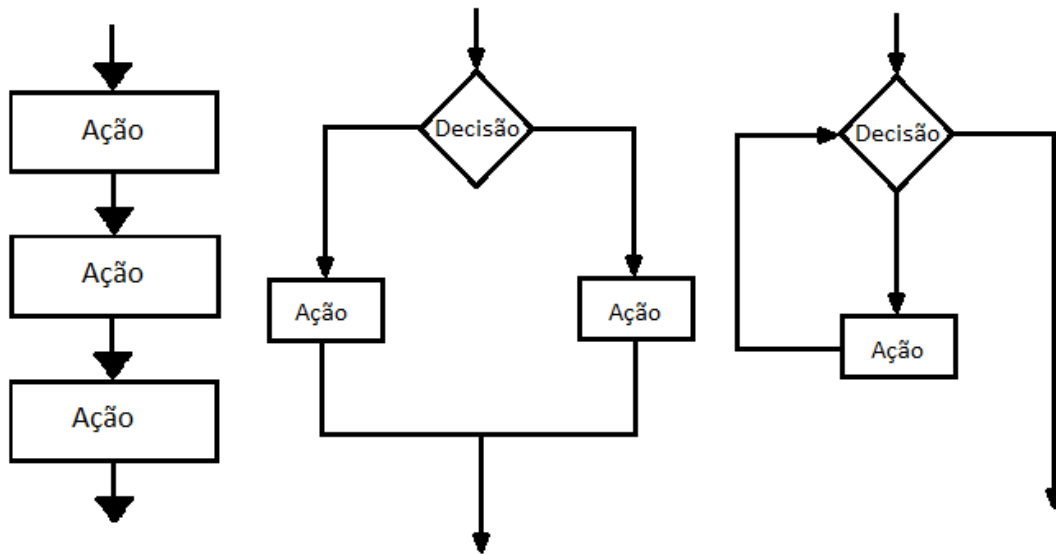


Figura 10: Estruturas de sequencia de ações, seleção e loop respectivamente.

A estrutura de sequencia de ações representa as operações que acontecem de modo sequencial. A seleção ou decisão representa a escolha de um determinado caminho a partir da resposta da pergunta que será feita. O loop é usado para se repetir a ação ate que uma exigência seja cumprida.

4. JAVA

Java basicamente é composto por uma linguagem de programação e uma máquina virtual que é usada para executar aplicativos desenvolvidos com sua linguagem de programação.

4.1 HISTÓRIA DO JAVA

Inicialmente chamado de Green Project quando começou a ser criado, no ano de 1991, este projeto estava sendo desenvolvido para ser usado como a próxima geração de software embarcado. Em 1992 surge a linguagem Oak. Diversas tentativas de venda da linguagem foram feitas, mas nenhuma foi aceita. No ano de 1994 surge a internet e com ela a empresa Sun percebe uma nova possibilidade para o Green Project e desenvolve uma linguagem para construir aplicativos Web baseada na linguagem Oak como Java. No ano seguinte, no dia de 23 de maio na SunWorld Expo 95 é oficialmente lançada a linguagem Java com a versão JDK 1.0 alpha. A empresa Netscape começa a implementar interpretadores Java em seus navegadores, e a linguagem começa a crescer muito daí em diante. Em 1999 surge a plataforma Java 2 *Enterprise Edition* (J2EE) que é uma plataforma para desenvolvimento corporativo, e a Java 2 *Mobile Edition* (J2ME) usada para dispositivos móveis. Atualmente Java é usado em diversas áreas entre elas: aplicações web, desktop, jogos, chips de identificação e até na sonda Spirit que foi mandada a Marte.

4.2 Java 2D

O Java 2D é uma API padrão do JDK é versátil, pois esta dispõe de diversos recursos para desenhar direto na tela, manipular imagens, utilizar a impressora e desenhar textos. Ela é implementada como uma extensão do *Abstract Window Toolkit* (AWT), usado para desenhar as “janelas” dos programas. O Java 2D se tornou melhor após os projetistas do AWT perceberem que delegar a tarefa de desenhar janelas, botões e caixas de texto para o sistema operacional limitava a portabilidade e performance do Java. Após perceberem isso associaram o AWT ao Swing. Esta seria uma nova extensão responsável por desenhar todos os componentes na tela. Após o Swing tornar-se padrão para aplicações desktop, esta começou a receber críticas em relação à performance, o que obrigou a Sun a otimizar o Java 2D. Algumas das características do Java 2D são: grande número de primitivas geométricas, mecanismo para desenhar qualquer forma geométrica, mecanismo para detectar colisão, capacidade de combinar imagens que se interceptam, diretivas para controle de qualidade do desenho entre outras.

As primitivas geométricas em Java 2D são: pontos, linhas, retângulos, arcos, elipses e curvas. As classes que trabalham com essas primitivas estão definidas no pacote `Java.awt.geom`. A Figura 11 ilustra alguns exemplos das primitivas geométricas.

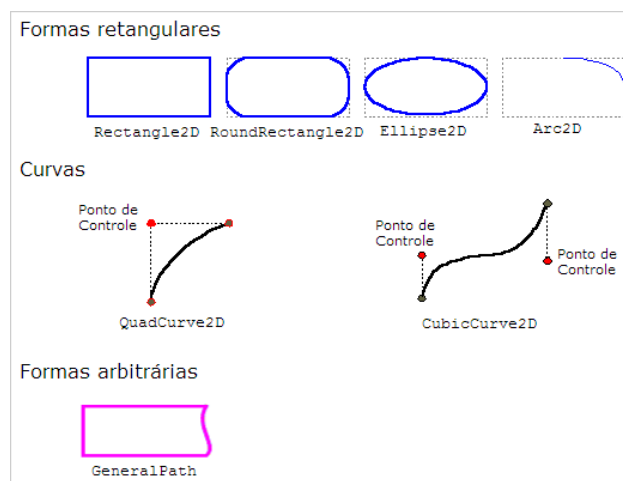


Figura 11: Exemplos de primitivas geométricas do Java 2D. Fonte [10].

Java 2D ainda oferece classes para trabalhar com textos. Mais frequentemente usada chamando-se os métodos `setFont` e `drawString` da classe `Graphics`. A API disponibiliza classes para manipulações mais avançadas, como posicionamento de texto. Um caractere dentro de uma fonte é chamado de `Glyph` e um caractere de texto pode ser composto por dois `glyphs`, como as letras acentuadas. Então, uma fonte é um conjunto de `glyphs` a qual pode ter variações como negrito e itálico. Cada uma dessas variações são chamadas de faces. Todas as faces de uma fonte são chamadas de família. Pode-se utilizar Java para alterar propriedades como altura, largura e outras métricas dessas fontes com propriedades encapsuladas na classe `FontMetrics`. Antes do texto ser desenhado na tela ele é formatado pela classe `TextLayout`, a qual oferece métodos para lidar com idiomas diferentes e textos bidirecionais, misturando assim fontes diversas. Uma classe para um controle mais aprimorado sobre opções de layout se encontra no Java 2D, e é chamada de `GlyphVector`, que é uma classe de mais baixo nível.

Analisando as funcionalidades do Java 2D para a manipulação de imagens, o primeiro recurso é a classe que permite a carga e escrita de imagens chamada `ImageIO`. A classe `BufferedImage` é usada para criar um imagem de tela que mais tarde será impressa inteira na tela, chamada de `buffer`. A funcionalidade do `buffer` é que se colocam todas as imagens e textos necessários no `buffer` e apenas o `buffer` é exibido na tela. Também são suportado diversos filtros através da classe `BufferedImageOP`. Funções de manipulação direta e eficaz dos pixels de uma imagem estão contidas na classe `Raster`. Esta manipulação avançada é normalmente usada apenas por quem vai trabalhar com aplicações que possuem filtros diretamente.

4.3 JAVA SWING

Nas primeiras versões do Java, para se trabalhar com programas gráficos, a única alternativa era uma biblioteca de baixo nível que dependia da plataforma onde era rodada, a AWT. Esta biblioteca tinha problemas de compatibilidade entre as plataformas, e isto fazia com que o programa tivesse aparências diferentes para cada plataforma usada.

Tendo em vista esse problema foi implementado a partir do Java 1.2 a biblioteca Swing que é mais fácil e estável para se usar. Esta biblioteca fornece componentes de mais alto nível, assim eliminando a dependência entre código e plataforma, deste modo tornando a aplicação com a aparência desejada em todas as plataformas. O Swing não possui nenhum código nativo, e isso possibilita que as aplicações tenham diferentes visuais. Ao contrário da AWT, onde todos os programas tinham a aparência do local onde eram rodadas.

Os componentes do Java Swing tem um “J” no começo dos seus nomes, como JButton. Algumas das vantagens do Swing sobre o AWT: áreas de textos que podem mostrar conteúdo HTML nativamente, botões com suporte a imagem, capacidade de alterar as bordas de quase todos os componentes, associar imagens a componentes e controle de como são desenhados em detalhes [11]. A figura 12 mostra os componentes do Java Swing.

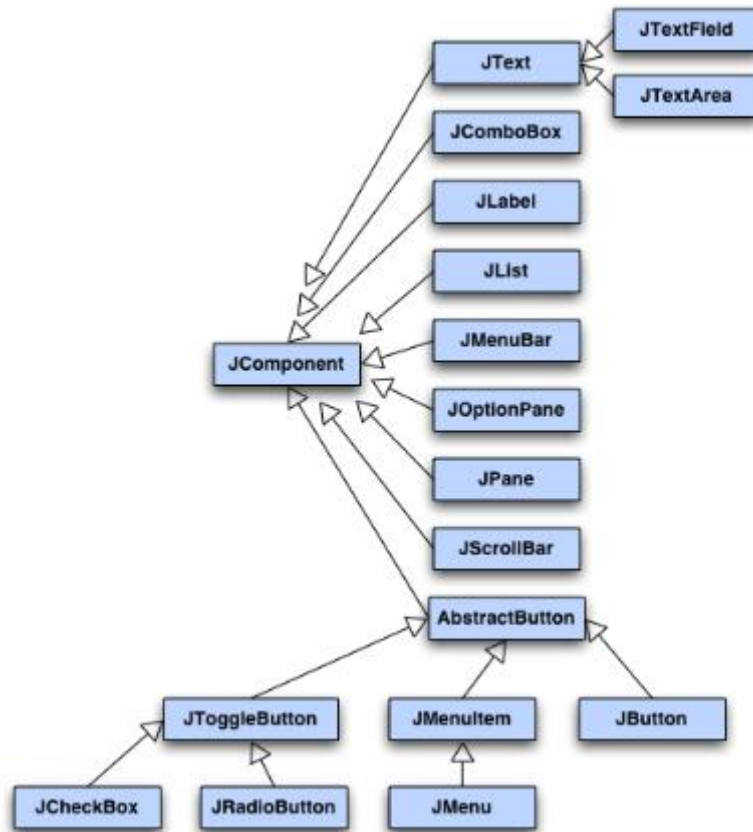


Figura 12: Classes do Java Swing. Fonte [12].

A figura 13 mostra um exemplo de código de Java Swing, neste exemplo e criado uma calculadora simples com as quatro operações matemáticas básicas.

```

public ExemploJavaSwing() {

    soma = new JButton("+");
    subtracao = new JButton("-");
    multiplicacao = new JButton("*");
    divisao = new JButton("/");
    label1 = new JLabel("Numero 1:");
    label2 = new JLabel("Numero 2:");
    label3 = new JLabel("Resultado");
    texto1 = new JTextField();
    texto2 = new JTextField();
    texto3 = new JTextField();

    painel1 = new JPanel();
    painel1.setLayout(new GridLayout(1, 3));
    add(painel1, BorderLayout.NORTH);
    painel1.add(label1);
    painel1.add(label2);
    painel1.add(label3);

    painel2 = new JPanel();
    painel2.setLayout(new GridLayout(1, 3));
    add(painel2, BorderLayout.CENTER);
    painel2.add(texto1);
    painel2.add(texto2);
    painel2.add(texto3);

    painel3 = new JPanel();
    painel3.setLayout(new GridLayout(1, 4));
    add(painel3, BorderLayout.SOUTH);
    painel3.add(soma);
    painel3.add(subtracao);
    painel3.add(multiplicacao);
    painel3.add(divisao);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(300, 100);
    setLocationRelativeTo(null);
    setTitle("exemplo");
    setVisible(true);
}

```

Figura 13: Exemplo de utilização de Java Swing

A figura 14 mostra o resultado deste código, gerando uma calculadora simples.

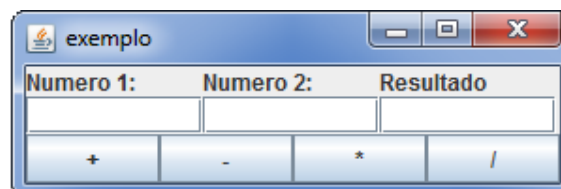


Figura 14: Resultado do código.

5. ANÁLISE DO PROJETO

5.1 DIAGRAMA DE CLASSES

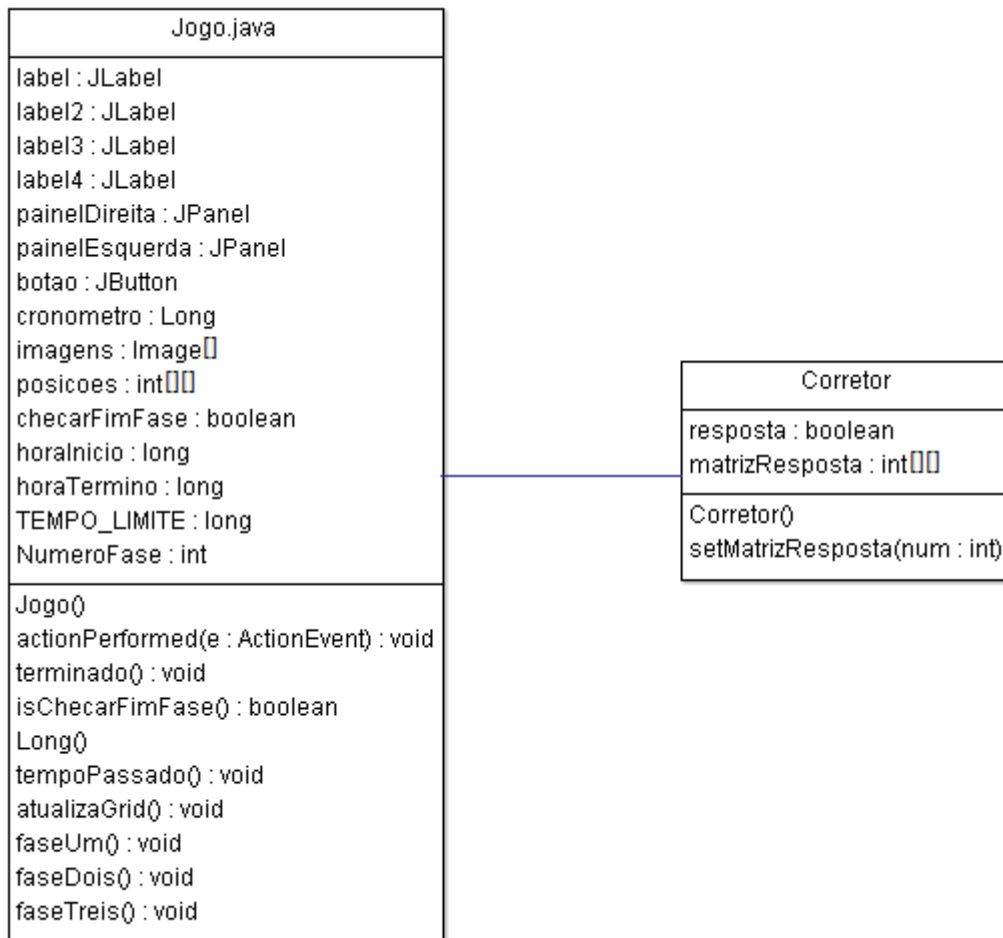


Figura 15: Diagrama de classes do projeto

5.2 DIAGRAMA DE ESTADOS

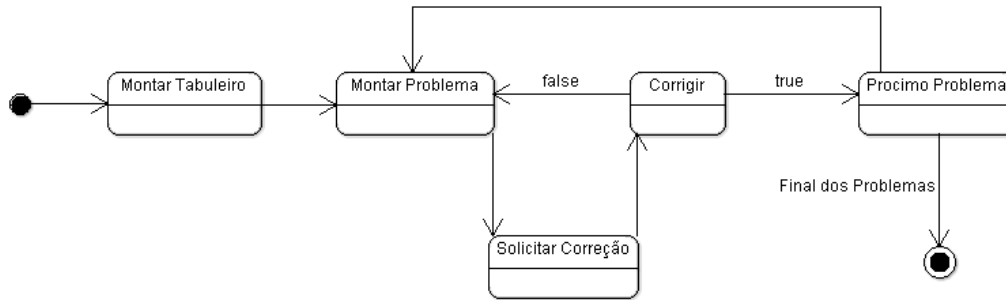


Figura 16: Diagrama de estados

5.3 CASOS DE USO

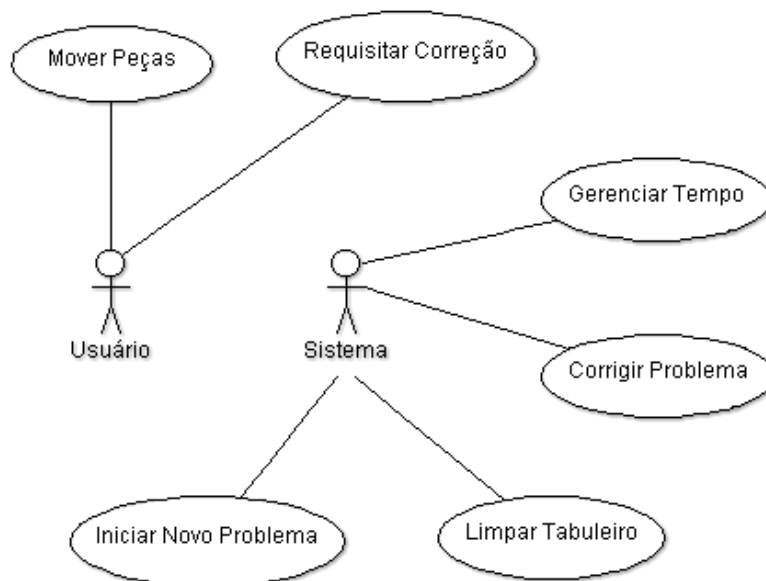


Figura 17: Diagrama de casos de uso

5.3.1 DESCRIÇÃO DOS CASOS DE USO

5.3.1.1 MOVER PEÇAS

1. Finalidade

- a. Alterar uma peça em um determinado local do tabuleiro.

2. Atores

- a. Usuario.

3. Pré-Requisito

- a. Iniciar alguma fase do jogo.

4. Fluxo Principal

- a. O Usuário clica com o mouse em um campo do tabuleiro.
- b. O Sistema verifica o campo.
- c. Caso o campo seja valido, se altera a peça desse campo.

5. Fluxo Alternativo

- a. O Usuário clica com o mouse em um campo do tabuleiro.
- b. O Sistema verifica o campo.
- c. Caso seja um campo invalido nada se altera.

6. Fluxo Exceção

7. Testes

- a. O Sistema verifica se algum campo foi clicado.

- b. O Sistema verifica se o campo e valido.

5.3.1.2 REQUISITAR CORREÇÃO

1. Finalidade

- a. Faz uma requisição para que o problema seja corrigido.

2. Atores

- a. Usuario.

3. Pré-Requisito

- a. Iniciar alguma fase do jogo.

4. Fluxo Principal

- a. O Usuário clica com o mouse no campo de finalização do tabuleiro.
- b. O Sistema inicia a correção.

5. Fluxo Alternativo

6. Fluxo Exceção

7. Testes

- a. O Sistema verifica se o campo de finalização foi clicado.

5.3.1.3 INICIAR NOVO PROBLEMA

1. Finalidade

- a. Prepara o tabuleiro para uma nova fase.
- 2. Atores
 - a. Sistema.
- 3. Pré-Requisito
 - a. Iniciar alguma fase do jogo.
- 4. Fluxo Principal
 - a. Uma nova fase é iniciada.
 - b. O Sistema limpa o tabuleiro das peças da fase anterior.
 - c. O Sistema posiciona as peças para a nova fase.
- 5. Fluxo Alternativo
 - a. O tempo da fase se esgota.
 - b. O Sistema reinicia a fase.
- 6. Fluxo Exceção
- 7. Testes

5.3.1.4 LIMPAR TABULEIRO

- 1. Finalidade
 - a. Limpa o tabuleiro para uma nova fase.
- 2. Atores
 - a. Sistema.
- 3. Pré-Requisito
 - a. Iniciar alguma fase do jogo.
- 4. Fluxo Principal

a. O Sistema limpa o tabuleiro quanto requisitado.

5. Fluxo Alternativo

6. Fluxo Exceção

7. Testes

5.3.1.5 CORRIGIR PROBLEMA

1. Finalidade

a. Corrigir a resposta do usuário para que ele possa prosseguir ou não.

2. Atores

a. Sistema.

3. Pré-Requisito

a. Iniciar alguma fase do jogo.

4. Fluxo Principal

a. O Usuário clica com o mouse no campo de finalização do tabuleiro.

b. O Sistema inicia a correção.

c. Se a resposta estiver certa prossegue para a próxima fase.

5. Fluxo Alternativo

a. O Usuário clica com o mouse no campo de finalização do tabuleiro.

b. O Sistema inicia a correção.

c. Se a resposta estiver errada uma mensagem de resposta errada e exibida e o tabuleiro e liberado para o Usuário tentar de novo.

6. Fluxo Exceção

7. Testes

- a. O Sistema verifica se o campo de finalização foi clicado.
- b. O Sistema verifica se a resposta do Usuário esta correta.

5.3.1.6 GERENCIAR TEMPO

1. Finalidade

- a. Gerencia o tempo de cada fase.

2. Atores

- a. Sistema.

3. Pré-Requisito

- a. Iniciar alguma fase do jogo.

4. Fluxo Principal

- a. A fase e iniciada.
- b. O tempo começa a ser contado de forma decrescente.
- c. Caso o tempo acabe uma mensagem e exibida e a fase e reiniciada.

5. Fluxo Alternativo

6. Fluxo Exceção

7. Testes

- a. O Sistema verifica se o tempo gasto pelo Usuário no problema e menor que a determinada.

6. RESULTADO

6.1 CONSTRUTOR DA CLASSE JOGO

Foi gerado um jogo onde é disponibilizado um tabuleiro e um problema que deve ser resolvido pelo jogador. Cada casa do tabuleiro e uma peça e pode ser modificada pelo jogador para outra peça para resolver o problema dado. A fase deve ser terminada antes de o tempo acabar ou esta será reiniciada.

O código abaixo mostra o construtor da classe jogo onde são iniciados os componentes que serão usados para a construção do tabuleiro.

```
public Jogo(){

    label = new JLabel("Tempo Restante:");
    label.setFont(new Font(null, 5, 20));
    label2 = new JLabel();
    label2.setFont(new Font(null, 5, 20));
    label3 = new JLabel("Problema:");
    label3.setFont(new Font(null, 5, 20));
    label4 = new JLabel("");
    label4.setFont(new Font(null, 5, 20));

    painelEsquerda = new JPanel();
    painelEsquerda.setLayout(new GridLayout(10, 10, 0, 0));
    painelDireita = new JPanel();
    painelDireita.setLayout(new GridLayout(10, 1));
    add(painelEsquerda, BorderLayout.CENTER);
    add(painelDireita, BorderLayout.EAST);
    painelDireita.add(label);
    painelDireita.add(label2);
    painelDireita.add(label3);
    painelDireita.add(label4);

    ImageIcon ii1 = new ImageIcon(this.getClass().getResource("imagens/inicio.jpg"));
    ImageIcon ii2 = new ImageIcon(this.getClass().getResource("imagens/retangulo.jpg"));
    .
    .
    .
    ImageIcon ii23 = new ImageIcon(this.getClass().getResource("imagens/reta5.jpg"));
```



```
ImageIcon ii24 = new ImageIcon(this.getClass().getResource("imagens/reta6.jpg"));

imagens[0] = ii5.getImage();
imagens[1] = ii4.getImage();
imagens[2] = ii1.getImage();
.
.
.
imagens[23] = ii24.getImage();
imagens[24] = ii6.getImage();

faseUm();

for(int i=0;i<10;i++){
    for(int j=0;j<10;j++){

        botao = new JButton();
        botao.addActionListener(this);
        painelEsquerda.add(botao);
        botao.setIcon(new ImageIcon(imagens[posicoes[i][j]]));

    }
}

setSize(1200, 1000);
setTitle("TCC - Jogo Atualmente Sem Nome");
setResizable(false);
setLocationRelativeTo(null);
setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
setVisible(true);

}
```

A figura 18 mostra resultado do código acima.

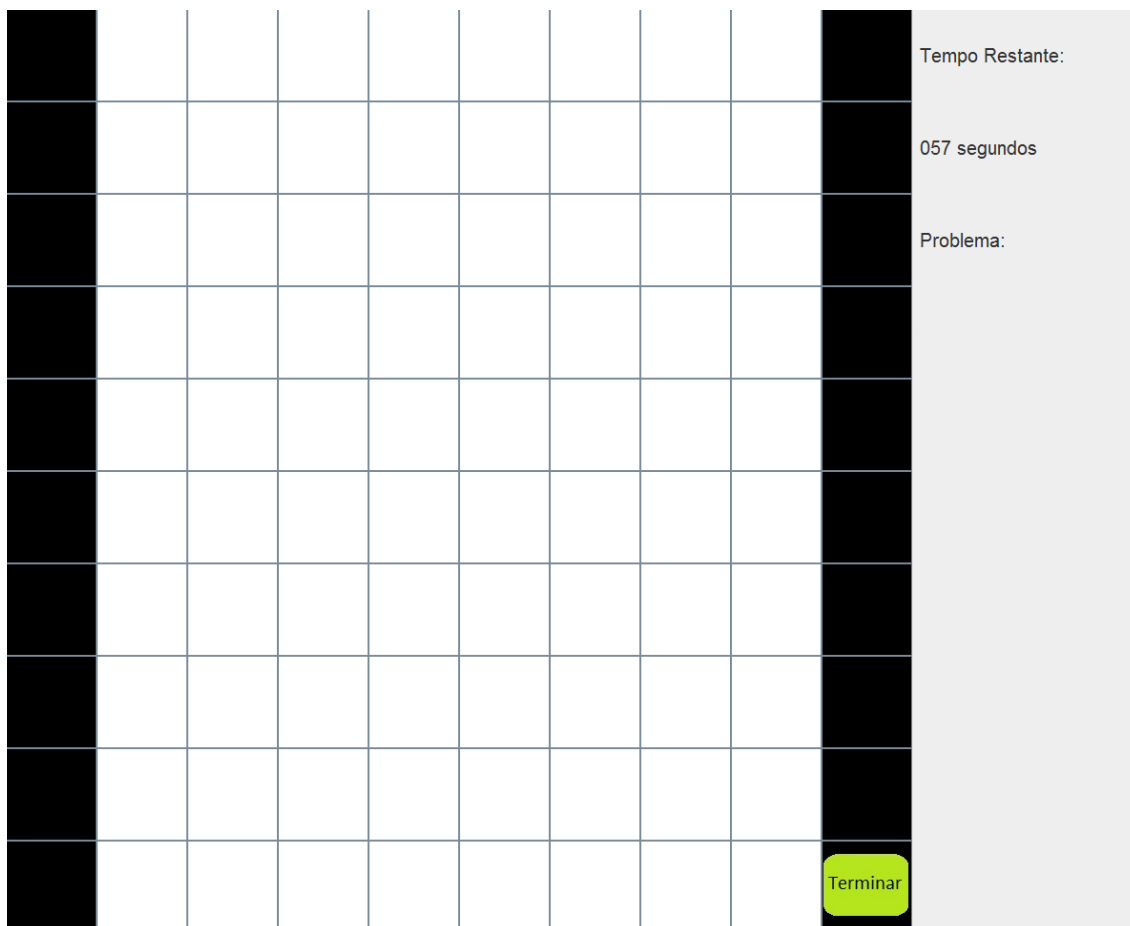


Figura 18: Tabuleiro do jogo.

6.2 MÉTODO ACTIONPERFORMED

O código abaixo representa o método sobre-escrito **actionPerformed** que é usado para mudar as figuras de cada casa do tabuleiro e para quando se escolher o campo Terminar, chamar o método de correção do problema.

```
public void actionPerformed(ActionEvent e) {
    JButton botao = (JButton) e.getSource();
    Dimension tamanho = botao.getSize();
    int botaoX = botao.getX();
    int botaoY = botao.getY();
    int posBotaoX = botaoX / tamanho.width;
    int posBotaoY = botaoY / tamanho.height;
```

```

int indice = posicoes[posBotaoY][posBotaoX];

switch(indice){
    case 1:

        if(posicoes[posBotaoY][posBotaoX-1] == 2 || posicoes[posBotaoY][posBotaoX-1] == 3 ||
posicoes[posBotaoY][posBotaoX-1] == 4){

            posicoes[posBotaoY][posBotaoX] = 8;
            botao.setIcon(new ImageIcon(imagens[8]));

        }else if(posicoes[posBotaoY][posBotaoX+1] == 2 || posicoes[posBotaoY][posBotaoX+1] == 3
|| posicoes[posBotaoY][posBotaoX+1] == 4){
            posicoes[posBotaoY][posBotaoX] = 8;
            botao.setIcon(new ImageIcon(imagens[8]));

        }else if(posicoes[posBotaoY-1][posBotaoX] == 2 || posicoes[posBotaoY-1][posBotaoX] == 3
|| posicoes[posBotaoY-1][posBotaoX] == 4){
            posicoes[posBotaoY][posBotaoX] = 6;
            botao.setIcon(new ImageIcon(imagens[6]));

        }else if(posicoes[posBotaoY+1][posBotaoX] == 2 || posicoes[posBotaoY+1][posBotaoX] == 3
|| posicoes[posBotaoY+1][posBotaoX] == 4){
            posicoes[posBotaoY][posBotaoX] = 6;
            botao.setIcon(new ImageIcon(imagens[6]));

        } else {

            posicoes[posBotaoY][posBotaoX] = 3;
            botao.setIcon(new ImageIcon(imagens[3]));}
            break;
            .
            .
            .
        }
    }
}

```

As variáveis definidas no começo deste código são para identificar a posição da casa selecionada e também seu valor na matriz.

6.3 MÉTODO DE TEMPO

Abaixo o método que controla o tempo da fase, o mostrando na tela e quando este termina a fase é reiniciada.

```
public void tempoPassado(){  
  
    String resposta = null;  
  
    Long conta;  
  
    horaTermino = System.currentTimeMillis();  
  
    conta = (horaTermino - horaInicio)/1000;  
  
    conta = TEMPO_LIMITE - conta;  
  
    if(conta >= 100){  
        resposta = conta.toString();  
    } else if(conta < 100 && conta >= 10){  
        resposta = "0"+conta.toString();  
    } else resposta = "00"+conta.toString();  
  
    if(conta == 0) {  
  
        JOptionPane.showMessageDialog(null, "Tempo Esgotado...Recomeçando Problema");  
        if(NumeroFase == 1){  
            faseUm();  
            atualizarGrid();  
        }else  
            if(NumeroFase == 2){  
                faseDois();  
                atualizarGrid();  
            }else  
                if(NumeroFase == 3){  
                    faseTres();  
                    atualizarGrid();  
                }  
  
    }  
  
    label2.setText(resposta + " segundos");  
  
}
```


6.4 A CLASSE DE CORREÇÃO

A classe corretora contém o método de **Resposta** que pega a matriz de **resposta dada** pelo jogado e confere com a **matriz resposta** que ele possui, salva. Se a matriz for compatível com a resposta desejada, o atributo de passagem de fase receberá true, ou false em caso contrário. O método **setMatrizResposta** apenas guarda a matriz resposta correspondente a fase na variável de mesmo nome.

```
public class Corretor {  
  
    private boolean resposta = false;  
    private int matrizResposta[][];  
  
    public boolean Resposta(int[][] matriz, int num){  
  
        setMatrizResposta(num);  
  
        for(int i=0;i<10;i++){  
            for(int j=0;j<10;j++){  
  
                if(matriz[i][j] == matrizResposta[i][j]){  
  
                    resposta = true;  
  
                } else {  
  
                    resposta = false;  
                    return resposta;  
  
                }  
  
            }  
        }  
  
        return resposta;  
    }  
}
```

7. CONCLUSÃO

Como desenvolvimento pessoal, este trabalho apresentou um desafio na forma de ver e programar o Java, usando a API 2D.

Trabalhar com este Serious Game, visando ensinar linguagem de programação, foi desafiador, pois exigiu do programador criatividade em criar o sistema, empenho em aprender sobre uma nova biblioteca de funções e objetos, e juntar todo o conhecimento para a realização deste projeto.

Fica como sugestão desenvolver o sistema, implementando novas fases, e novas funcionalidades.

8. REFERÊNCIAS BIBLIOGRÁFICAS

[1] CLUA, Esteban Walter Gonzalez; BITTENCOURT João Ricardo. Desenvolvimento de Jogos 3D: Concepção, Design e Programação. In: Congresso da Sociedade Brasileira de Computação, XXV, São Leopoldo.

[2] Tennis for Two. Em: <
<http://pceconsolas.blogspot.com.br/2009/12/tennis-for-two.html>>. Acesso em: 10/10/2013.

[3] MASSARANI, Sandro. Spacewar! Espaço - A Fronteira Inicial. Em: <
http://www.massarani.com.br/FGHQ_Spacewar.html> Acesso em: 10/10/2013.

[4] Atari. Em: <
<http://pt.wikipedia.org/wiki/Atari>> Acesso em: 10/10/2013.

[5] PEREIRA, Cilene. Videogame do bem. Em:
<http://www.istoe.com.br/reportagens/5303_VIDEOGAME+DO+BEM>. Acesso em: 16/07/2013.

[6] GENESTRETI, Guilherme. Jogos eletrônicos estimulam recuperação de pacientes. Em:
<<http://www1.folha.uol.com.br/equilibrioesaude/878564-jogos-eletronicos-estimulam-recuperacao-de-pacientes.shtml>>. Acesso em: 15/06/2013

[7] Xilo-Game brasileiro que é cabra da peste. Em: <
<http://www.girlsofwar.com.br/xilo-game-brasileiro-que-e-cabra-da-pestes/>>.
Acesso em: 20/10/2013

[8] SANTOS, Núbia dos; RAPKIEWICZ, Clevi Elena; XEXÉO, José Antônio Moreira; CORDEIRO, Rogério Avellar. Jogos Educacionais – ferramentas para o ensino de programação.

[9] GIANNOTTI, José Arthur. Jogo ProGame. Em: <
<http://jogoprogame.webnode.com.br/>>. Acesso em: 20/10/13

[10] MENDONÇA, Vinicius Godoy de. Uma visão rápida sobre Java 2d. Em: <<http://pontov.com.br/site/java/48-java2d/93-uma-visao-rapida-sobre-o-java-2d>>. Acesso em: 02/07/2013

[11] STEIL, Rafael. Introdução a programação gráfica em Java com Swing. Em< <http://www.guj.com.br/articles/38>>. Acesso em: 20/10/2013

[12]CAMPOS, Pedro F; NUNES, Nuno J. Introdução ao Java Swing e AWT. Em: < <http://cee.uma.pt/people/faculty/pedro.campos/docs/guia-ihm.pdf>>. Acesso em: 20/10/2013.

[13]MACHADO, Liliane S; MORAES, Ronei M; NUNES, Fátima L.S. Serious Game para Saúde e Treinamento Imersivo.

[14]FARREL, Joyce. Lógica e Design de Programação: Introdução. 5. ed. Tradução de André Schifnagel Avrichir, Editora Cengage Learning, 2010.

ANEXO

```
package testejogo3;

import com.sun.java.swing.plaf.windows.resources.windows;

import java.awt.BorderLayout;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.GridLayout;

import java.awt.Image;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JDialog;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.plaf.basic.BasicSplitPaneUI;

import sun.awt.VerticalBagLayout;

/**
 *
 * @author Jean
 */

public class Jogo extends JFrame implements ActionListener{

    private JLabel label , label2, label3, label4;
```

```
private JPanel painelDireita;

private JPanel painelEsquerda;

private JButton botao;

private Long cronometro;

private Image imagens[] = new Image[25];

private int posicoes[][];

private boolean checarFimFase = false;

private long horainicio, horaTermino;

private final long TEMPO_LIMITE = 60;

int NumeroFase = 0;

public Jogo(){

    label = new JLabel("Tempo Restante:");
    label.setFont(new Font(null, 5, 20));
    label2 = new JLabel();
    label2.setFont(new Font(null, 5, 20));
    label3 = new JLabel("Problema:");
    label3.setFont(new Font(null, 5, 20));
    label4 = new JLabel("");
    label4.setFont(new Font(null, 5, 20));

    painelEsquerda = new JPanel();
    painelEsquerda.setLayout(new GridLayout(10, 10, 0, 0));
    painelDireita = new JPanel();
    painelDireita.setLayout(new GridLayout(10, 1));
    add(painelEsquerda, BorderLayout.CENTER);
    add(painelDireita, BorderLayout.EAST);
```



```
ImagemCon ii22 = new ImagemCon(this.getClass().getResource("imagens/reta4.jpg"));
ImagemCon ii23 = new ImagemCon(this.getClass().getResource("imagens/reta5.jpg"));
ImagemCon ii24 = new ImagemCon(this.getClass().getResource("imagens/reta6.jpg"));

imagens[0] = ii5.getImage();//preto
imagens[1] = ii4.getImage();//branco
imagens[2] = ii1.getImage();//inicio
imagens[3] = ii3.getImage();//losango
imagens[4] = ii2.getImage();//retangulo
imagens[5] = ii25.getImage();//fim
imagens[6] = ii7.getImage();
imagens[7] = ii8.getImage();
imagens[8] = ii9.getImage();
imagens[9] = ii10.getImage();
imagens[10] = ii11.getImage();
imagens[11] = ii12.getImage();
imagens[12] = ii13.getImage();
imagens[13] = ii14.getImage();
imagens[14] = ii15.getImage();
imagens[15] = ii16.getImage();
imagens[16] = ii17.getImage();
imagens[17] = ii18.getImage();
imagens[18] = ii19.getImage();
imagens[19] = ii20.getImage();
imagens[20] = ii21.getImage();
imagens[21] = ii22.getImage();
imagens[22] = ii23.getImage();
imagens[23] = ii24.getImage();
imagens[24] = ii6.getImage();//terminar
```

```
faseUm();

for(int i=0;i<10;i++){
    for(int j=0;j<10;j++){

        botao = new JButton();
        botao.addActionListener(this);
        painelEsquerda.add(botao);
        botao.setIcon(new ImageIcon(imagens[posicoes[i][j]]));

    }
}

setSize(800, 600);
setTitle("TCC - Jogo Atualmente Sem Nome");
setResizable(false);
setLocationRelativeTo(null);
setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
setVisible(true);

} // final do construtor

public void actionPerformed(ActionEvent e) {
    JButton botao = (JButton) e.getSource();
    Dimension tamanho = botao.getSize();
    int botaoX = botao.getX();
    int botaoY = botao.getY();
```

```
int posBotaoX = botaoX / tamanho.width;

int posBotaoY = botaoY / tamanho.height;

int indice = posicoes[posBotaoY][posBotaoX];

System.out.println("Botão clicado foi: " + posBotaoX + posBotaoY + " com valor de: " +
indice);

switch(indice){
    case 1:

        if(posicoes[posBotaoY][posBotaoX-1] == 2 || posicoes[posBotaoY][posBotaoX-1] == 3
|| posicoes[posBotaoY][posBotaoX-1] == 4){

            posicoes[posBotaoY][posBotaoX] = 8;
            botao.setIcon(new ImageIcon(imagens[8]));

        }else if(posicoes[posBotaoY][posBotaoX+1] == 2 ||
posicoes[posBotaoY][posBotaoX+1] == 3 || posicoes[posBotaoY][posBotaoX+1] == 4){

            posicoes[posBotaoY][posBotaoX] = 8;
            botao.setIcon(new ImageIcon(imagens[8]));

        }else if(posicoes[posBotaoY-1][posBotaoX] == 2 || posicoes[posBotaoY-
1][posBotaoX] == 3 || posicoes[posBotaoY-1][posBotaoX] == 4){

            posicoes[posBotaoY][posBotaoX] = 6;
            botao.setIcon(new ImageIcon(imagens[6]));

        }else if(posicoes[posBotaoY+1][posBotaoX] == 2 ||
posicoes[posBotaoY+1][posBotaoX] == 3 || posicoes[posBotaoY+1][posBotaoX] == 4){

            posicoes[posBotaoY][posBotaoX] = 6;
            botao.setIcon(new ImageIcon(imagens[6]));
```

```
    } else {

        posicoes[posBotaoY][posBotaoX] = 3;
        botao.setIcon(new ImageIcon(imagens[3]));
        break;
    case 3:
        posicoes[posBotaoY][posBotaoX] = 4;
        botao.setIcon(new ImageIcon(imagens[4]));
        break;
    case 4:
        posicoes[posBotaoY][posBotaoX] = 1;
        botao.setIcon(new ImageIcon(imagens[1]));
        break;
    case 6:
        posicoes[posBotaoY][posBotaoX] = 7;
        botao.setIcon(new ImageIcon(imagens[7]));
        break;
    case 7:
        if(posicoes[posBotaoY-1][posBotaoX] == 2 || posicoes[posBotaoY-1][posBotaoX] == 3
|| posicoes[posBotaoY-1][posBotaoX] == 4){
            posicoes[posBotaoY][posBotaoX] = 11;
            botao.setIcon(new ImageIcon(imagens[11]));
        }else if(posicoes[posBotaoY+1][posBotaoX] == 2 ||
posicoes[posBotaoY+1][posBotaoX] == 3 || posicoes[posBotaoY+1][posBotaoX] == 4){
            posicoes[posBotaoY][posBotaoX] = 13;
            botao.setIcon(new ImageIcon(imagens[13]));
        }
        break;
    case 8:
        posicoes[posBotaoY][posBotaoX] = 9;
```



```
        botao.setIcon(new ImageIcon(imagens[9]));

        break;

    case 9:

        if(posicoes[posBotaoY][posBotaoX-1] == 2 || posicoes[posBotaoY][posBotaoX-1] == 3
|| posicoes[posBotaoY][posBotaoX-1] == 4){

            posicoes[posBotaoY][posBotaoX] = 10;

            botao.setIcon(new ImageIcon(imagens[10]));

        }

        }else if(posicoes[posBotaoY][posBotaoX+1] == 2 ||
posicoes[posBotaoY][posBotaoX+1] == 3 || posicoes[posBotaoY][posBotaoX+1] == 4){

            posicoes[posBotaoY][posBotaoX] = 12;

            botao.setIcon(new ImageIcon(imagens[12]));

        }

        break;

    case 10:

        if(posicoes[posBotaoY][posBotaoX-1] == 2 || posicoes[posBotaoY][posBotaoX-1] == 3
|| posicoes[posBotaoY][posBotaoX-1] == 4){

            posicoes[posBotaoY][posBotaoX] = 11;

            botao.setIcon(new ImageIcon(imagens[11]));

        }

        }else if(posicoes[posBotaoY+1][posBotaoX] == 2 ||
posicoes[posBotaoY+1][posBotaoX] == 3 || posicoes[posBotaoY+1][posBotaoX] == 4){

            posicoes[posBotaoY][posBotaoX] = 14;

            botao.setIcon(new ImageIcon(imagens[14]));

        }

        break;

    case 11:

        if(posicoes[posBotaoY][posBotaoX-1] == 2 || posicoes[posBotaoY][posBotaoX-1] == 3
|| posicoes[posBotaoY][posBotaoX-1] == 4){
```

```
posicoes[posBotaoY][posBotaoX] = 15;

botao.setIcon(new ImagemIcon(imagens[15]));

    }else if(posicoes[posBotaoY-1][posBotaoX] == 2 || posicoes[posBotaoY-
1][posBotaoX] == 3 || posicoes[posBotaoY-1][posBotaoX] == 4){

        posicoes[posBotaoY][posBotaoX] = 12;

        botao.setIcon(new ImagemIcon(imagens[12]));

    }

    break;

case 12:

    if(posicoes[posBotaoY][posBotaoX+1] == 2 || posicoes[posBotaoY][posBotaoX+1] ==
3 || posicoes[posBotaoY][posBotaoX+1] == 4){

        posicoes[posBotaoY][posBotaoX] = 13;

        botao.setIcon(new ImagemIcon(imagens[13]));

    }else if(posicoes[posBotaoY-1][posBotaoX] == 2 || posicoes[posBotaoY-
1][posBotaoX] == 3 || posicoes[posBotaoY-1][posBotaoX] == 4){

        posicoes[posBotaoY][posBotaoX] = 16;

        botao.setIcon(new ImagemIcon(imagens[16]));

    }

    break;

case 13:

    if(posicoes[posBotaoY][posBotaoX+1] == 2 || posicoes[posBotaoY][posBotaoX+1] ==
3 || posicoes[posBotaoY][posBotaoX+1] == 4){

        posicoes[posBotaoY][posBotaoX] = 14;

        botao.setIcon(new ImagemIcon(imagens[14]));
```

```

        }else if(posicoes[posBotaoY+1][posBotaoX] == 2 ||
posicoes[posBotaoY+1][posBotaoX] == 3 || posicoes[posBotaoY+1][posBotaoX] == 4){
            posicoes[posBotaoY][posBotaoX] = 15;
            botao.setIcon(new ImageIcon(imagens[15]));
        }
        break;
        case 14:
            if(posicoes[posBotaoY][posBotaoX+1] == 2 || posicoes[posBotaoY][posBotaoX+1] ==
3 || posicoes[posBotaoY][posBotaoX+1] == 4){
                posicoes[posBotaoY][posBotaoX] = 17;
                botao.setIcon(new ImageIcon(imagens[17]));
            }
            }else if(posicoes[posBotaoY+1][posBotaoX] == 2 ||
posicoes[posBotaoY+1][posBotaoX] == 3 || posicoes[posBotaoY+1][posBotaoX] == 4){
                posicoes[posBotaoY][posBotaoX] = 18;
                botao.setIcon(new ImageIcon(imagens[18]));
            }
        }
        break;
        case 15:
            if(posicoes[posBotaoY][posBotaoX-1] == 2 || posicoes[posBotaoY][posBotaoX-1] == 3
|| posicoes[posBotaoY][posBotaoX-1] == 4){
                posicoes[posBotaoY][posBotaoX] = 16;
                botao.setIcon(new ImageIcon(imagens[16]));
            }
            }else if(posicoes[posBotaoY+1][posBotaoX] == 2 ||
posicoes[posBotaoY+1][posBotaoX] == 3 || posicoes[posBotaoY+1][posBotaoX] == 4){
                posicoes[posBotaoY][posBotaoX] = 10;
                botao.setIcon(new ImageIcon(imagens[10]));
            }
        }

```

```
        break;

    case 16:

        if(posicoes[posBotaoY][posBotaoX-1] == 2 || posicoes[posBotaoY][posBotaoX-1] == 3
|| posicoes[posBotaoY][posBotaoX-1] == 4){

            posicoes[posBotaoY][posBotaoX] = 19;

            botao.setlcon(new ImagemIcon(imagens[19]));

        }else if(posicoes[posBotaoY-1][posBotaoX] == 2 || posicoes[posBotaoY-
1][posBotaoX] == 3 || posicoes[posBotaoY-1][posBotaoX] == 4){

            posicoes[posBotaoY][posBotaoX] = 17;

            botao.setlcon(new ImagemIcon(imagens[17]));

        }

        break;

    case 17:

        if(posicoes[posBotaoY][posBotaoX+1] == 2 || posicoes[posBotaoY][posBotaoX+1] ==
3 || posicoes[posBotaoY][posBotaoX+1] == 4){

            posicoes[posBotaoY][posBotaoX] = 19;

            botao.setlcon(new ImagemIcon(imagens[19]));

        }else if(posicoes[posBotaoY-1][posBotaoX] == 2 || posicoes[posBotaoY-
1][posBotaoX] == 3 || posicoes[posBotaoY-1][posBotaoX] == 4){

            posicoes[posBotaoY][posBotaoX] = 18;

            botao.setlcon(new ImagemIcon(imagens[18]));

        }

        break;

    case 18:

        if(posicoes[posBotaoY-1][posBotaoX] == 2 || posicoes[posBotaoY-1][posBotaoX] == 3
|| posicoes[posBotaoY-1][posBotaoX] == 4){
```

```

posicoes[posBotaoY][posBotaoX] = 21;

botao.setIcon(new ImagemIcon(imagens[21]));

    }else if(posicoes[posBotaoY+1][posBotaoX] == 2 ||
posicoes[posBotaoY+1][posBotaoX] == 3 || posicoes[posBotaoY+1][posBotaoX] == 4){
        posicoes[posBotaoY][posBotaoX] = 20;
        botao.setIcon(new ImagemIcon(imagens[20]));
    }
    break;
case 19:
    if(posicoes[posBotaoY][posBotaoX-1] == 2 || posicoes[posBotaoY][posBotaoX-1] == 3
|| posicoes[posBotaoY][posBotaoX-1] == 4){

        posicoes[posBotaoY][posBotaoX] = 20;
        botao.setIcon(new ImagemIcon(imagens[20]));

    }else if(posicoes[posBotaoY][posBotaoX+1] == 2 ||
posicoes[posBotaoY][posBotaoX+1] == 3 || posicoes[posBotaoY][posBotaoX+1] == 4){
        posicoes[posBotaoY][posBotaoX] = 22;
        botao.setIcon(new ImagemIcon(imagens[22]));
    }
    break;
case 20:
    if(posicoes[posBotaoY][posBotaoX-1] == 2 || posicoes[posBotaoY][posBotaoX-1] == 3
|| posicoes[posBotaoY][posBotaoX-1] == 4){

        posicoes[posBotaoY][posBotaoX] = 21;
        botao.setIcon(new ImagemIcon(imagens[21]));

    }else if(posicoes[posBotaoY+1][posBotaoX] == 2 ||
posicoes[posBotaoY+1][posBotaoX] == 3 || posicoes[posBotaoY+1][posBotaoX] == 4){

```

```
        posicoes[posBotaoY][posBotaoX] = 23;
        botao.setIcon(new ImagemIcon(imagens[23]));
    }
    break;
    case 21:
        if(posicoes[posBotaoY][posBotaoX-1] == 2 || posicoes[posBotaoY][posBotaoX-1] == 3
|| posicoes[posBotaoY][posBotaoX-1] == 4){

            posicoes[posBotaoY][posBotaoX] = 1;
            botao.setIcon(new ImagemIcon(imagens[1]));

        }else if(posicoes[posBotaoY-1][posBotaoX] == 2 || posicoes[posBotaoY-
1][posBotaoX] == 3 || posicoes[posBotaoY-1][posBotaoX] == 4){
            posicoes[posBotaoY][posBotaoX] = 22;
            botao.setIcon(new ImagemIcon(imagens[22]));
        }
        break;
    case 22:
        if(posicoes[posBotaoY][posBotaoX+1] == 2 || posicoes[posBotaoY][posBotaoX+1] ==
3 || posicoes[posBotaoY][posBotaoX+1] == 4){

            posicoes[posBotaoY][posBotaoX] = 23;
            botao.setIcon(new ImagemIcon(imagens[23]));

        }else if(posicoes[posBotaoY-1][posBotaoX] == 2 || posicoes[posBotaoY-
1][posBotaoX] == 3 || posicoes[posBotaoY-1][posBotaoX] == 4){
            posicoes[posBotaoY][posBotaoX] = 1;
            botao.setIcon(new ImagemIcon(imagens[1]));
        }
        break;
    case 23:
```

```
        if(posicoes[posBotaoY][posBotaoX+1] == 2 || posicoes[posBotaoY][posBotaoX+1] ==
3 || posicoes[posBotaoY][posBotaoX+1] == 4){

            posicoes[posBotaoY][posBotaoX] = 1;
            botao.setIcon(new ImageIcon(imagens[1]));

        }else if(posicoes[posBotaoY+1][posBotaoX] == 2 ||
posicoes[posBotaoY+1][posBotaoX] == 3 || posicoes[posBotaoY+1][posBotaoX] == 4){

            posicoes[posBotaoY][posBotaoX] = 1;
            botao.setIcon(new ImageIcon(imagens[1]));

        }

        break;
    case 24:
        terminado();
        break;

    }

}

public void terminado(){

    Corretor corretor = new Corretor();
    checarFimFase = corretor.Resposta(posicoes, NumeroFase);
    if(checarFimFase) JOptionPane.showMessageDialog(null, "Correto");
    else JOptionPane.showMessageDialog(null, "Tente Denovo");

}
```

```
public boolean isChecarFimFase() {  
    return checarFimFase;  
}  
  
public Long getCronometro() {  
    return cronometro;  
}  
  
public void tempoPassado(){  
  
    String resposta = null;  
  
    Long conta;  
  
    horaTermino = System.currentTimeMillis();  
  
    conta = (horaTermino - horaInicio)/1000;  
  
    conta = TEMPO_LIMITE - conta;  
  
    if(conta >= 100){  
        resposta = conta.toString();  
    } else if(conta < 100 && conta >= 10){
```



```
        resposta = "0"+conta.toString();
    } else resposta = "00"+conta.toString();

    if(conta == 0) {

        JOptionPane.showMessageDialog(null, "Tempo Esgotado...Recomeçando Problema");
        if(NumeroFase == 1){
            faseUm();
            atualizarGrid();
        }else
            if(NumeroFase == 2){
                faseDois();
                atualizarGrid();
            }else
                if(NumeroFase == 3){
                    faseTres();
                    atualizarGrid();
                }
        }

        label2.setText(resposta + " segundos");

    }

    public void atualizarGrid(){
```

```
painelEsquerda.removeAll();

for(int i=0;i<10;i++){
    for(int j=0;j<10;j++){

        botao = new JButton();
        botao.addActionListener(this);
        painelEsquerda.add(botao);
        botao.setIcon(new ImageIcon(imagens[posicoes[i][j]]));

    }
}

}

public void faseUm(){

    NumeroFase = 1;

    checarFimFase = false;

    horaInicio = System.currentTimeMillis();

    label4.setText("Complete a instrução if ");

    posicoes = new int[][] {{0,0,0,0,0,0,0,0,0,0},
                            {0,1,1,1,2,1,1,1,1,0},
                            {0,1,1,1,6,1,1,1,1,0},
```

```
        {0,1,1,1,1,10,1,1,1,0},
        {0,1,1,1,6,4,1,1,1,0},
        {0,1,1,1,4,6,1,1,1,0},
        {0,1,1,1,6,5,1,1,1,0},
        {0,1,1,1,5,1,1,1,1,0},
        {0,1,1,1,1,1,1,1,1,0},
        {0,0,0,0,0,0,0,0,0,24}};

}

public void faseDois(){

    NumeroFase = 2;

    checarFimFase = false;

    horainicio = System.currentTimeMillis();

    label4.setText("Complete a instrução while");

    posicoes = new int[][] {{0,0,0,0,0,0,0,0,0,0},
        {0,1,1,1,2,1,1,1,1,0},
        {0,1,1,1,6,1,1,1,1,0},
        {0,1,1,1,4,1,1,1,1,0},
        {0,1,1,1,6,18,1,1,1,0},
        {0,1,1,1,1,21,1,1,1,0},
        {0,1,1,1,1,1,1,1,1,0},
        {0,1,1,1,5,1,1,1,1,0},
```

```
        {0,1,1,1,1,1,1,1,1,0},
        {0,0,0,0,0,0,0,0,0,24}};

    }

    public void faseTres(){

        NumeroFase = 3;

        checarFimFase = false;

        horalInicio = System.currentTimeMillis();

        label4.setText("Completo while(){if}");

        posicoes = new int[][] {
            {0,0,0,0,0,0,0,0,0,0},
            {0,2,1,1,1,1,1,1,1,0},
            {0,6,1,1,1,1,1,1,1,0},
            {0,4,8,19,19,20,15,1,1,0},
            {0,6,1,1,1,1,18,1,1,0},
            {0,1,9,1,9,4,18,1,1,0},
            {0,6,1,6,1,1,18,1,1,0},
            {0,5,1,4,1,19,21,1,1,0},
            {0,1,1,1,1,1,1,1,1,0},
            {0,0,0,0,0,0,0,0,0,24}};

    }
```

```
public void fechar(){
```

```
setVisible(false);  
  
}  
  
}
```

```
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package testejogo3;  
  
/**  
 *  
 * @author Jean  
 */  
public class Corretor {  
  
    private boolean resposta = false;  
    private int matrizResposta[][];  
  
    public boolean Resposta(int[][] matriz, int num){  
  
        setMatrizResposta(num);  
  
        for(int i=0;i<10;i++){  
            for(int j=0;j<10;j++){  
  
                if(matriz[i][j] == matrizResposta[i][j]){
```

```
        resposta = true;

    } else {

        resposta = false;
        return resposta;

    }

}

}

}

return resposta;
}

public void setMatrizResposta(int num){

    if(num == 1){

        matrizResposta = new int[][] { {0,0,0,0,0,0,0,0,0},
            {0,1,1,1,2,1,1,1,0},
            {0,1,1,1,6,1,1,1,0},
            {0,1,1,1,3,10,1,1,0},
            {0,1,1,1,6,4,1,1,0},
            {0,1,1,1,4,6,1,1,0},
            {0,1,1,1,6,5,1,1,0},
            {0,1,1,1,5,1,1,1,0},
            {0,1,1,1,1,1,1,1,0},
            {0,0,0,0,0,0,0,0,24}};
```



```
}  
  
}
```