



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

---

Rodrigo dos Santos Carricondo

**DESENVOLVIMENTO DE UM *SERIOUS GAME* VOLTADO AO  
ENSINO DE LINGUAGEM DE PROGRAMAÇÃO EM C++**

ASSIS  
2013

**Rodrigo dos Santos Carricondo**

**DESENVOLVIMENTO DE UM *SERIOUS GAME* VOLTADO AO  
ENSINO DE LINGUAGEM DE PROGRAMAÇÃO EM C++**

Trabalho de Conclusão de Curso apresentado  
ao Instituto Municipal de Ensino Superior de  
Assis, como requisito parcial e obtenção do  
Certificado de Conclusão.

**Orientado:** Rodrigo dos Santos Carricondo

**Orientador:** Dr. Luiz Carlos Begosso

ASSIS  
2013

## FICHA CATALOGRÁFICA

DOS SANTOS CARRICONDO, Rodrigo.

DESENVOLVIMENTO DE UM SERIOUS GAMES VOLTADO AO ENSINO DE LINGUAGEM DE PROGRAMAÇÃO C++ / Rodrigo dos Santos Carricondo. Fundação Educacional do Município de Assis –FEMA– Assis, 2013.**45p.**

Orientador: Dr. Luiz Carlos Begosso

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

1. Serious Game 2. Aprendizado.

CDD: 001.61

Biblioteca da FEMA

## DEDICATÓRIA

Dedico este trabalho a  
minha mãe Maria Antonia  
que sempre esteve ao meu  
lado em todas as  
dificuldades e sempre foi  
meu pé de apoio em todas  
as minhas escolhas.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, por todo seu amor para comigo tornando assim possível que mais uma etapa da minha vida fosse concluída. Agradeço a minha mãe Maria Antonia dos Santos que além de mãe sempre foi minha melhor amiga, me ajudando e dando o maior suporte quando mais precisei. Agradeço ao meu pai Manoel Carricondo Júnior que sempre foi o maior exemplo de homem para mim, muito do que sou hoje é graças a todo esse amor que ele tem transmitido ao longo de toda minha vida. A minha namorada Jéssica Laura Chacon Belotti que mesmo com pouco tempo, tem sido uma das maiores motivações para a conclusão de mais esta etapa em minha vida, sem todo o seu carinho, amor e paciência eu não teria conseguido. Agradeço a toda minha família, meus tios, primos e minha vó, meus amigos, todos aqui contribuíram para minha formação e sabedoria.

## RESUMO

O conceito de utilizar jogos com propósitos educativos tem a sua origem ainda antes da revolução tecnológica e do uso comum de computadores. Um *Serious Game* é um software ou hardware desenvolvido através dos princípios do jogo interativo, com o objetivo de transmitir um conteúdo de caráter educativo ao usuário e facilitar o seu aprendizado de uma forma lúdica. Neste trabalho será apresentado o desenvolvimento de um Serious Game voltado ao ensino de linguagem C++ destinado a alunos iniciantes de cursos da área de informática.

**Palavras chave:** Serious Games, Linguagem C++

## **ABSTRACT**

The concept of using games for educational purposes has its origin even before the technological revolution and the common use of computers. A Serious Game software or hardware is developed through the principles of interactive game, in order to transmit a character education content to the user and facilitate your learning in a playful way. In this work will be presented the development of a Serious Game aimed at teaching language C + + for students beginning courses in the area of information technology.

**Keywords:** Serious Games, Language C + +

## LISTA DE FIGURAS

Figura 1: Diagrama de Caso de Uso .....	17
Figura 2: Caso de Uso: Iniciar Jogo .....	17
Figura 3: Caso de Uso: Jogar Game.....	18
Figura 4: Diagrama de Atividades .....	19
Figura 5: Tela Inicial Mundo C++ .....	24
Figura 6: Tela Explicativa Jogo.....	25
Figura 7: Jogo Associação de Imagens.....	26



## LISTA DE TABELAS

Tabela 1: Iniciar Jogo.....	17
Tabela 2: Jogar Game.....	18
Tabela 3: Cronograma.....	21

## SUMÁRIO

1. INTRODUÇÃO.....	12
1.1. OBJETIVO DO TRABALHO.....	13
1.2. PÚBLICO ALVO.....	13
1.3. JUSTIFICATIVA.....	14
2. DESENVOLVIMENTO DE JOGOS.....	15
2.1. MERCADO DE JOGOS.....	15
2.2. CONCEITOS BÁSICOS DE JOGOS.....	15
2.3. METODOLOGIA DE ANÁLISE.....	16
2.4. LEVANTAMENTO DOS REQUISITOS.....	17
2.5. DIAGRAMAS UML.....	17
2.5.1. DIAGRAMA DE CASO DE USO.....	17
2.5.2. USE CASE 1 – INICIAR JOGO.....	18
2.5.3. USE CASE 2 – JOGAR GAME.....	19
2.6. DIAGRAMA DE ATIVIDADES.....	20
2.7. DIAGRAMA DE WBS (WORK BREAKDOWN STRUCTURE).....	21
3. CRONOGRAMA.....	22
4. DESENVOLVIMENTO.....	23
4.1. Adobe Flash CS6.....	23
4.2. Action Script 3.0.....	23
4.2.1. Trabalho com os objetos.....	25
4.2.2. Criando Objetos.....	25
4.2.3. Condicionais.....	26
4.2.3.1. If..else.....	26
4.2.3.2. If..else if.....	26
4.2.3.3. Switch.....	27
4.2.4. Repetição.....	27
4.2.4.1. For.....	27

4.2.4.2.	For...in.....	27
4.2.4.3.	For each..in.....	28
4.2.4.4.	While.....	28
4.2.4.5.	Do..While .....	28
4.2.5.	Funções .....	28
4.2.5.1.	Chamada de funções.....	29
4.2.6.	Classes .....	29
5.	MUNDO C++.....	30
5.1.	Cenários.....	31
6.	CONCLUSÃO .....	33
7.	REFERÊNCIAS BIBLIOGRÁFICAS.....	34

## 1. INTRODUÇÃO

Os computadores estão cada vez mais presentes na vida cotidiana da sociedade. Com o grande crescimento tecnológico a chegada dos computadores na área da educação tem como objetivo tornar o ensino-aprendizagem mais dinâmico e prático. Nem sempre o uso dessa tecnologia é feita de maneira correta, é necessário refletir sobre o que se espera dessa tecnologia como recurso pedagógico para ser utilizado neste processo.

Justifica-se a introdução de computadores nas instituições de ensino e empresas através dos argumentos de que os mesmos se tornaram uma ferramenta eficaz, que possibilita ao seu usuário uma maior motivação e desperta no mesmo a capacidade de aprender e a resolver problemas.

Uma forma interessante de utilizar o ferramental computacional à serviço do aprendizado e do treinamento de usuários são os *serious games*.

De acordo com Derryberry (2008), o que difere o *serious games* dos outros jogos é o seu foco principal. Normalmente o conteúdo a ser ensinado deve ser interativo, deve conter telas atrativas, animação possui uma fácil interação com o usuário podendo variar seus ambientes de acordo com a dificuldade que é escolhida. Complementando este raciocínio, Gee (2013), afirma que os *serious games* devem seguir as seguintes diretrizes: interatividade de conteúdo, produção, customização, consolidação, informação sob demanda, sentidos contextualizados, incentivo a lateralidade, ferramentas inteligentes e performance antes da competência. O desenvolvimento de aplicações no contexto dos *serious games* tende a proporcionar um ambiente muito crítico, o aluno deve ter consciência que tudo é usado para a construção de seu conhecimento de uma forma mais prazerosa a fim de desenvolver seu pensar.

(...) por muitos anos os jogos têm sido usados apenas para diversão, mas só recentemente têm sido aplicados os elementos estratégicos de jogos em computadores com propósitos instrutivos (Lerner, 1991. p.31.).

A combinação feita por meio de jogos e computadores se torna muito eficiente, pois associa a riqueza que os *serious games* possuem com o grande poder de atração que existem nos computadores. Com a junção dessas duas ferramentas obtemos os jogos educacionais computadorizados, onde o jogo não será apenas para diversão e sim para explorar determinados ramos do conhecimento e até trabalhar habilidades no usuário, como, por exemplo, raciocínio lógico, destreza, entre outros.

É nesse contexto que o *serious game* ganha um espaço como uma importante ferramenta para a aprendizagem.

Para a elaboração desse trabalho, parte-se do princípio de que um *serious games* podem auxiliar o usuário a construir um saber de forma mais prazerosa, enriquecendo o seu vocabulário e sua personalidade profissional. Como material pedagógico, o *serious game* pode auxiliar o professor a conduzir a aula de uma forma mais atrativa. Como ferramenta de treinamento em empresas, pode estimular seus usuários, dando combustível para o seu próprio progresso com as novas descobertas que o jogo propõe.

## 1.1. OBJETIVO DO TRABALHO

Este trabalho tem por objetivo criar um *serious game* para o aprendizado de Linguagem de Programação em C++, dando ênfase também no seu uso de forma adequada para tornar o ensino aprendizagem mais dinâmico e prazeroso para o usuário.

## 1.2. PÚBLICO ALVO

O *serious game* que será desenvolvido tem como o seu público alvo o aluno que está iniciando cursos da área de informática. Estes cursos, geralmente, abordam o conteúdo de algoritmos e os implementam em Linguagem de Programação C++.

### 1.3. JUSTIFICATIVA

A ideia da criação deste *serious game* surgiu da vontade de trazer para o aluno de Linguagem de Programação em C++ um aprendizado mais visual. Como no início todos os exercícios e conceitos são vistos apenas em uma tela pequena e preta do MS-DOS, com esse jogo o usuário poderá visualizar seus exercícios de uma forma mais agradável e atual, gerando assim mais interesse pela programação.

## 2. DESENVOLVIMENTO DE JOGOS

### 2.1. MERCADO DE JOGOS

O Brasil tem o quarto maior mercado do mundo no segmento de jogos digitais, são 35 milhões de usuários. O mercado nacional de games movimentou R\$ 5,3 bilhões em 2012, teve um crescimento de 32% em relação ao ano anterior. As perspectivas de crescimento são positivas, embora seja um setor ainda recente e que carece de estrutura e incentivos. O crescimento do mercado de games é uma oportunidade lucrativa do segmento de Economia Criativa. Os dados referentes ao crescimento do setor no Brasil e as perspectivas de expansão do segmento de quadruplicar o faturamento até 2016, revelam a existência de oportunidades de negócios. Caracterizada como uma área bastante versátil e abrangente que atinge os mais variados públicos, a expansão do mercado de games está além do entretenimento. Jogos publicitários, educativos e empresariais representam importantes oportunidades de negócio para essa indústria.

Os grandes eventos esportivos que o Brasil sediará nos próximos anos abrem um enorme mercado para jogos temáticos das variadas modalidades de esportes existentes. Além disso, as demandas geradas pela Copa do Mundo da FIFA em 2014, trazem a oportunidade de desenvolvimento de jogos educativos e empresariais voltadas para os mais diversos segmentos da economia.

### 2.2. CONCEITOS BÁSICOS DE JOGOS

O filósofo HUIZINGA (1938), definiu jogo como:

Uma atividade voluntária exercida dentro de certos e determinados limites de tempo e espaço, segundo regras livremente consentidas, mas absolutamente obrigatórias, dotado de um fim em si mesmo, acompanhado de um sentimento de tensão e alegria e de uma consciência de ser diferente de vida

cotidiana.

Segundo IMPA (2007), existem dois tipos de jogos, os jogos cooperativos e os não cooperativos, onde suas diferenças estão no enfoque sobre o jogador.

Antes de iniciar o desenvolvimento de um jogo é preciso pensar em qual jogo criar, se estará criando algo novo. Para um jogo ser desenvolvido é necessário uma equipe qualificada. PERUCIA et. al (2005) ressaltam que para a maior parte das empresas que tem como meta o desenvolvimento de bons jogos e de qualidade, são requeridos cargos específicos para cada área, a lista abaixo mostra quais são:

- Programadores: Encarregados de desenvolver o software do jogo.
- Artistas: Responsáveis pelo *layout* do jogo. Fazem os objetos, texturas, ilustrações, personagens, animações etc.
- Projetista de níveis: Inventam as fases dos jogos estruturando seus desafios.
- Projetista de jogos: Considerados os administradores dos projetos também são conhecidos como *Game Designers*. São responsáveis desde a criação das ideias, preparação do *design document*.
- Planejador de Software: dividem o projeto do jogo em um conjunto de requisitos e avaliam o grau de dificuldade para desenvolver os requisitos e o tempo necessário.
- Arquiteto-chefe: Trabalha em conjunto com o planejador de software, elaboram um conjunto de especificações e módulos e toda a arquitetura geral do projeto é de sua responsabilidade.
- Gerente de projeto: Gerencia a carga de tarefas geradas, e planeja um melhor jeito de realizar as tarefas.
- Sonoplastas: Compõe as trilhas sonoras, efeitos especiais e vozes para os jogos.
- Testadores: Responsáveis por procurarem falhas e erros.

### 2.3. METODOLOGIA DE ANÁLISE



Toda a parte de documentação de análise será utilizada a metodologia de Análise Orientada a Objetos, utilizando a linguagem de modelagem UML (Unified Modeling Language).

## 2.4. LEVANTAMENTO DOS REQUISITOS

Os levantamentos de requisitos foram feitos em três meses, no qual foi possível acompanhar o desenvolvimento de um jogo feito pela equipe de desenvolvimento de jogos independentes do site Programadores de Jogos ([www.programadoresdejogos.com](http://www.programadoresdejogos.com)). Analisando e estudando os requisitos levantados, foram escolhidos três deles que servirão de base para o desenvolvimento do jogo. São eles:

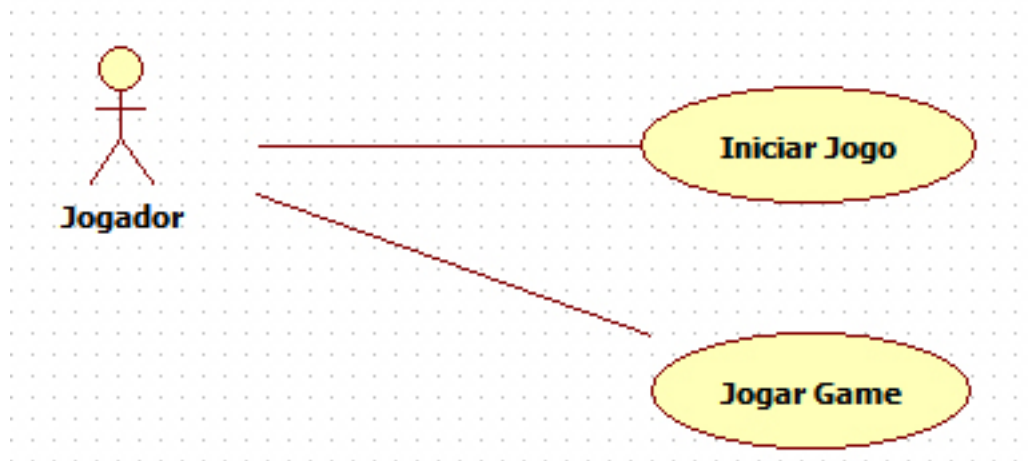
- **Autonomia:** o estímulo necessário para que o jogador inicie a interação com a ferramenta e a capacidade de interagir com o computador manuseando mouse e teclado.

- **Motivação:** análise do interesse do jogador pelo jogo, estímulo necessário para que o jogador permaneça jogando durante determinado período de tempo e o tempo máximo para que cada jogador jogue ativamente.

- **Habilidades:** análise das habilidades dos jogadores em relação às ações específicas de cada jogo como, por exemplo, a capacidade de associar imagens aos seus respectivos conceitos.

## 2.5. DIAGRAMAS UML

### 2.5.1. DIAGRAMA DE CASO DE USO



**Figura 1 – Diagrama de Casos de Uso**

### 2.5.2. USE CASE 1 – INICIAR JOGO



**Figura 2 – Caso de Uso: Iniciar Jogo**

<b>Funcionalidade/Objetivo</b>	Iniciar o Jogo
<b>Ator</b>	Jogador
<b>Pré Condição</b>	Nenhuma
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1- O jogo exibe um menu inicial. [A1]</li> <li>2- O Jogador clica em iniciar.</li> <li>3- O jogo inicia.</li> </ol>
<b>Cenário Alternativo</b>	A1 – Se o jogador não clicar em iniciar o jogo não começa.

**Tabela 1 – Iniciar Jogo**

### 2.5.3. USE CASE 2 – JOGAR GAME



Figura 3 – Caso de Uso: Jogar Game

<b>Funcionalidade/Objetivo</b>	Iniciar o jogo
<b>Ator</b>	Jogador
<b>Pré Condição</b>	Ter iniciado o Jogo
<b>Cenário Principal</b>	<ol style="list-style-type: none"><li>1- O jogo mostra uma tela com uma breve introdução sobre o Mundo C++.</li><li>2- O jogador clica no botão 'Jogar'.</li><li>3- O jogo é iniciado.</li></ol>
<b>Cenário Alternativo</b>	

Tabela 2 – Jogar Game

## 2.6. DIAGRAMA DE ATIVIDADES

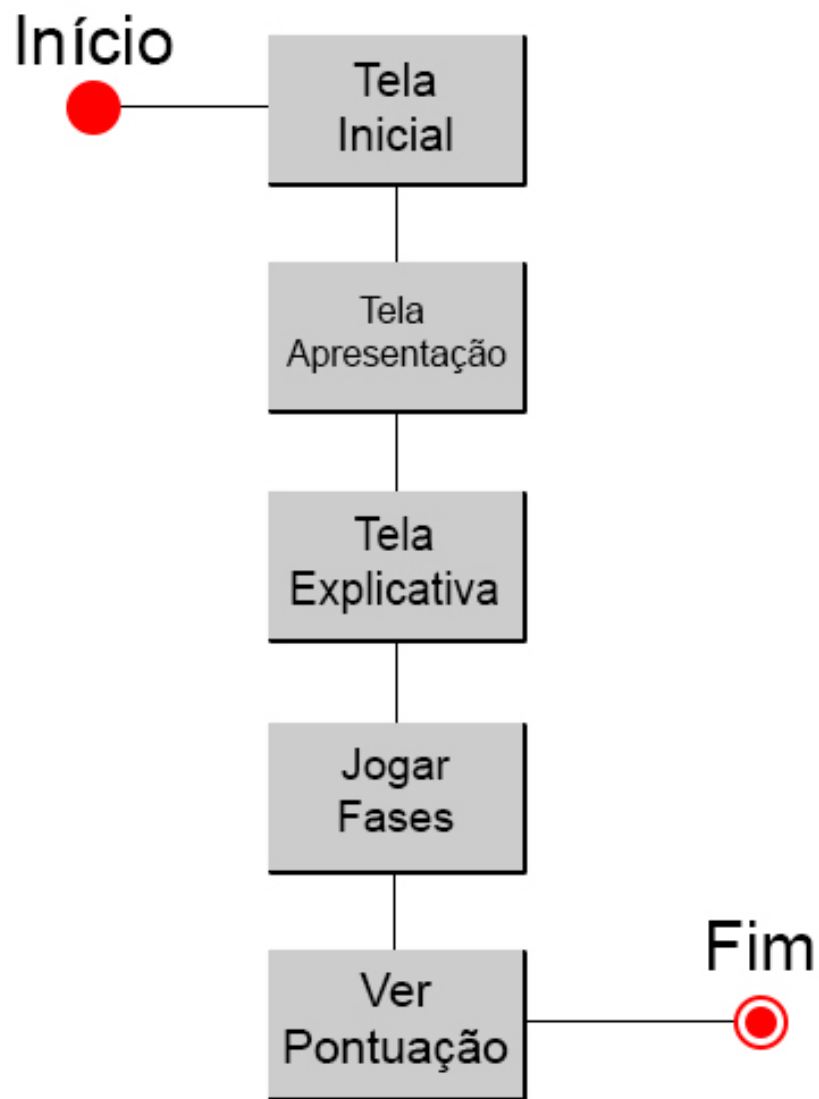
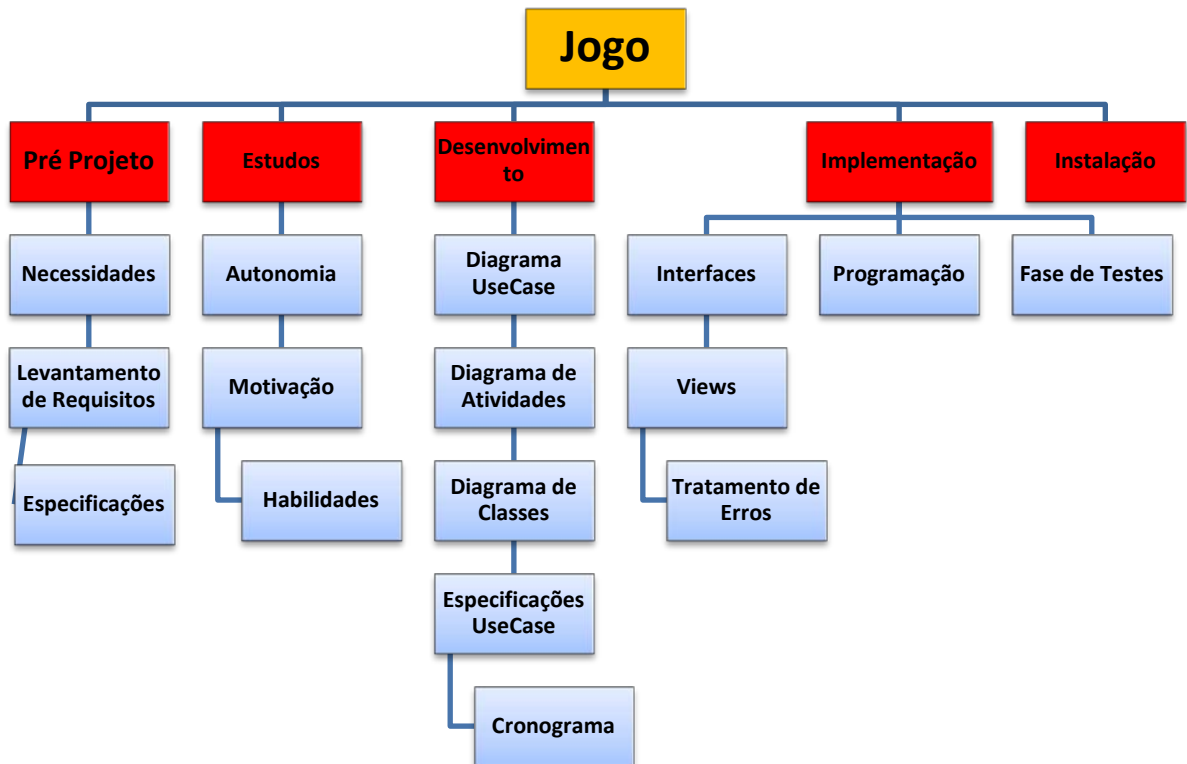


Figura 4 – Diagrama de Atividades

## 2.7. DIAGRAMA DE WBS (WORK BREAKDOWN STRUCTURE)



### 3. CRONOGRAMA

ATIVIDADES	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV
PRÉ-PROJETO	■	■								
REQUISITOS		■	■	■						
MODELAGEM				■						
DOCUMENTAÇÃO				■	■					
QUALIFICAÇÃO						■				
PROGRAMAÇÃO							■	■	■	
TESTE									■	
CONCLUSÃO										■

Tabela 3 – Cronograma

## 4. DESENVOLVIMENTO

### 4.1. Adobe Flash CS6

Para o desenvolvimento do jogo foi utilizado a ferramenta Adobe Flash CS6, que possui um ambiente de criação perfeito para produzir diversos tipos de conteúdos animados e interativo para jogos, sites, software, aplicativos portáteis, etc. Ele oferece a opção de se programar na linguagem Action Script 2.0 e Action Script 3.0, além disso, o Adobe Flash é um aplicativo que vem preparado para fornecer ferramentas que possibilitem criar textos, desenhos, botões e muito mais. Para realizar tais tarefas, o aplicativo necessita de recursos eficientes e fáceis de usar, características que são perceptíveis no programa.

Para desenhar, o Adobe Flash traz ferramentas avançadas de design, formas inteligentes e uma enormidade de opções para que o desenvolvedor não tenha que dispende muito esforço no momento de criar suas ilustrações. É possível utilizar recursos automatizados para rotação de objetos, ampliação e movimentação tridimensional.

### 4.2. Action Script 3.0

Adobe Actionscript é uma linguagem de programação da plataforma Adobe Flash. Originalmente desenvolvida como um meio para os desenvolvedores programarem dinamicamente, melhorando a eficiência do desenvolvimento de aplicações na plataforma Flash, desde uma imagem simples à uma complexa animação.

Nesta versão foram introduzidos novos recursos:

- Verificação de tipos em tempo de compilação e de execução: os tipos são agora preservados em tempo de execução. O Flash Player faz verificação de tipos no momento da execução

- Exceções em tempo de execução: Erros em tempo de execução disponibilizam estado da pilha ("stack traces") com indicação de arquivo fonte e número de linhas.
- Separação do sistema de herança de classes do sistema de herança de prototipação utilizado na versão anterior.
- Classes seladas: o comportamento padrão é que as classes não podem ser estendidas em tempo de execução, melhorando a alocação de memória e mantendo a aplicação mais segura. Classes dinâmicas podem ser utilizadas se definidas com o modificador *dynamic*.
- Suporte a pacotes (*package*), como em Java, *namespaces*.
- Métodos com clausura: as clausuras nos métodos guardam o estado dos objetos da qual originaram.
- A API do flash player foi reorganizada em *packages*.
- Sistema unificado de tratamento de eventos baseado no Document Object Model.
- Acesso direto à lista de componentes do Flash em tempo de execução, permitindo controle completo em ActionScript.

O código-fonte a seguir ilustra um exemplo de codificação ActionScript para imprimir "Olá, mundo!".

#### Exemplo de ActionScript 3.0

```
botao.addEventListener(MouseEvent.CLICK, escreve)
function escreve(e:MouseEvent){
    texto.text = "Olá, mundo!"
}
```

O ActionScript começou como uma linguagem de script para a ferramenta Macromedia Flash. As primeiras três versões das ferramentas Flash possuíam recursos de interatividade limitados. Os programadores podiam incluir comandos simples, chamados de "action", à um botão ou frame. O conjunto de ações era limitado a controles de navegação básicos, com comandos como "play", "stop", "getURL", e "gotoAndPlay". Com a introdução do Flash 4 em 1999, novos recursos como variáveis, expressões, operadores diversos, blocos de condição e loops foram adicionados.



#### 4.2.1. Trabalho com os objetos

O ActionScript é conhecido como uma linguagem de programação orientada a objetos. Programação orientada a objetos é simplesmente uma abordagem da programação, nada mais do que uma forma de organizar o código em um programa, usando objetos. Na programação orientada a objetos, as instruções do programa são divididas em objetos diferentes, ou seja, o código é agrupado em blocos de funcionalidade, de forma que tipos relacionados de funcionalidade ou partes relacionadas de informação são agrupados em um único contêiner. Imagine que você definiu um símbolo de clipe de filme, esse símbolo de clipe de filme também é um objeto no ActionScript. Na programação orientada a objetos do ActionScript, há três tipos de características que qualquer classe pode incluir:

- Propriedades
- Métodos
- Eventos

Juntos, esses elementos ajudam a gerenciar as partes dos dados usados pelo programa e a decidir quais ações são executadas em uma determinada ordem.

#### 4.2.2. Criando Objetos

Para que você possa usar um objeto no ActionScript, primeiro, ele deve existir. Uma parte da criação de um objeto é declarar uma variável; porém, a declaração de uma variável só cria um lugar vazio na memória do computador. Você deve atribuir um valor real à variável, o processo de criar um objeto é conhecido como instanciar o objeto. Eis alguns exemplos:

- Valor numérico literal (insere o número diretamente):

```
var somaNumero:Number = 17.239;
```

```
var somaNegativoInteiro:int = -53;
```

```
var somaUInteiro:uint = 22;
```

- Valor da seqüência de caracteres literal (envolve o texto com aspas duplas):

```
var Nome:String = "Rodrigo";
```

```
var Frase:String = "To be or not to be, that is the question...";
```

- Valor booleano literal (usa os valores literais true ou false):

```
var InicioEvento:Boolean = true;
```

```
var TocarMusica:Boolean = false;
```

- Valor Array literal (envolve uma lista de valores separados por vírgula entre colchetes):

```
var vetorEstacoes:Array = ["Primevera", "Verão", "Outono", "Inverno"];
```

#### 4.2.3. Condicionais

O ActionScript 3.0 fornece três instruções condicionais para que você possa controlar o fluxo de programa.

##### 4.2.3.1. If..else

Permite testar uma condição e executar um bloco de código se essa condição existir, ou executar um bloco de código alternativo se ela não existir:

```
If (x > 20) {  
    trace("x e >20");  
}  
else{  
    trace("x e < 20);  
}
```

##### 4.2.3.2. If..else if

É possível testar mais de uma condição usando esta condicional:

```
if (x > 20)  
{  
    trace("x e > 20");  
}  
else if (x < 0)  
{  
    trace("x e negativo");  
}
```

#### 4.2.3.3. Switch

A instrução será útil se houver vários caminhos de execução que dependam da mesma expressão de condição:

```
switch(dayNum)
{
    case 0:
        trace("Sunday");
        break;
    case 1:
        trace("Monday");
        break;
    case 2:
        trace("Tuesday");
        break;
}
```

#### 4.2.4. Repetição

Instruções de repetição permitem executar um bloco específico de código repetidamente usando uma série de valores ou variáveis.

##### 4.2.4.1. For

Permite fazer a iteração por meio de uma variável para um intervalo específico de valores:

```
var i:int;
for (i = 0; i < 5; i++){
    trace(i);
}
```

##### 4.2.4.2. For...in

A repetição itera por meio das propriedades de um objeto ou dos elementos de uma matriz:

```
var myObj:Object = {x:20, y:30};
for (var i:String in myObj)
{
    trace(i + ": " + myObj[i]);
}
// output:
// x: 20
// y: 30
```

#### 4.2.4.3. For each..in

A repetição itera por meio dos itens de um conjunto, que podem ser tags em um objeto XML, a variável do iterador contém o valor mantido pela propriedade em vez do nome da propriedade:

```
var myObj:Object = {x:20, y:30};
for each (var num in myObj)
    {
        trace(num);
    }
// output:
// 20
// 30
```

#### 4.2.4.4. While

A repetição é como uma instrução *if* que é repetida desde que a condição seja *true*:

```
var i:int = 0;
while (i < 5)
    {
        trace(i);
        i++;
    }
```

#### 4.2.4.5. Do..While

A repetição garante que o bloco de código seja executado pelo menos uma vez, porque a condição é verificada depois que o bloco é executado:

```
var i:int = 5;
do
    {
        trace(i);
        i++;
    } while (i < 5);
// output: 5
```

#### 4.2.5. Funções

As funções são blocos de códigos que executam tarefas e podem ser reutilizadas no seu programa.

#### 4.2.5.1. Chamada de funções

Para uma função ser usada é necessário que seu identificador seja usado. O operador parênteses delimita qualquer parâmetro de função que você deseja enviar para a função:

```
var NumeroAleatorio:Number = Math.random();
```

#### 4.2.6. Classes

Uma classe é uma representação abstrata de um objeto. Uma classe armazena informações sobre os tipos de dados que um objeto pode manter e os comportamentos que um objeto pode exibir.

As definições de classes no ActionScript 3.0 usam uma sintaxe semelhante a usada no ActionScript 2.0:

```
Public class teste  
{  
    Var ligado:Boolean = true;  
}
```

## 5. MUNDO C++

O Mundo C++ é um jogo de associação de imagens com conceitos relacionados com o cotidiano do aluno, o qual pode construir, aos poucos, a concepção correta dos conceitos de programação. A figura 5 ilustra a página inicial do jogo.

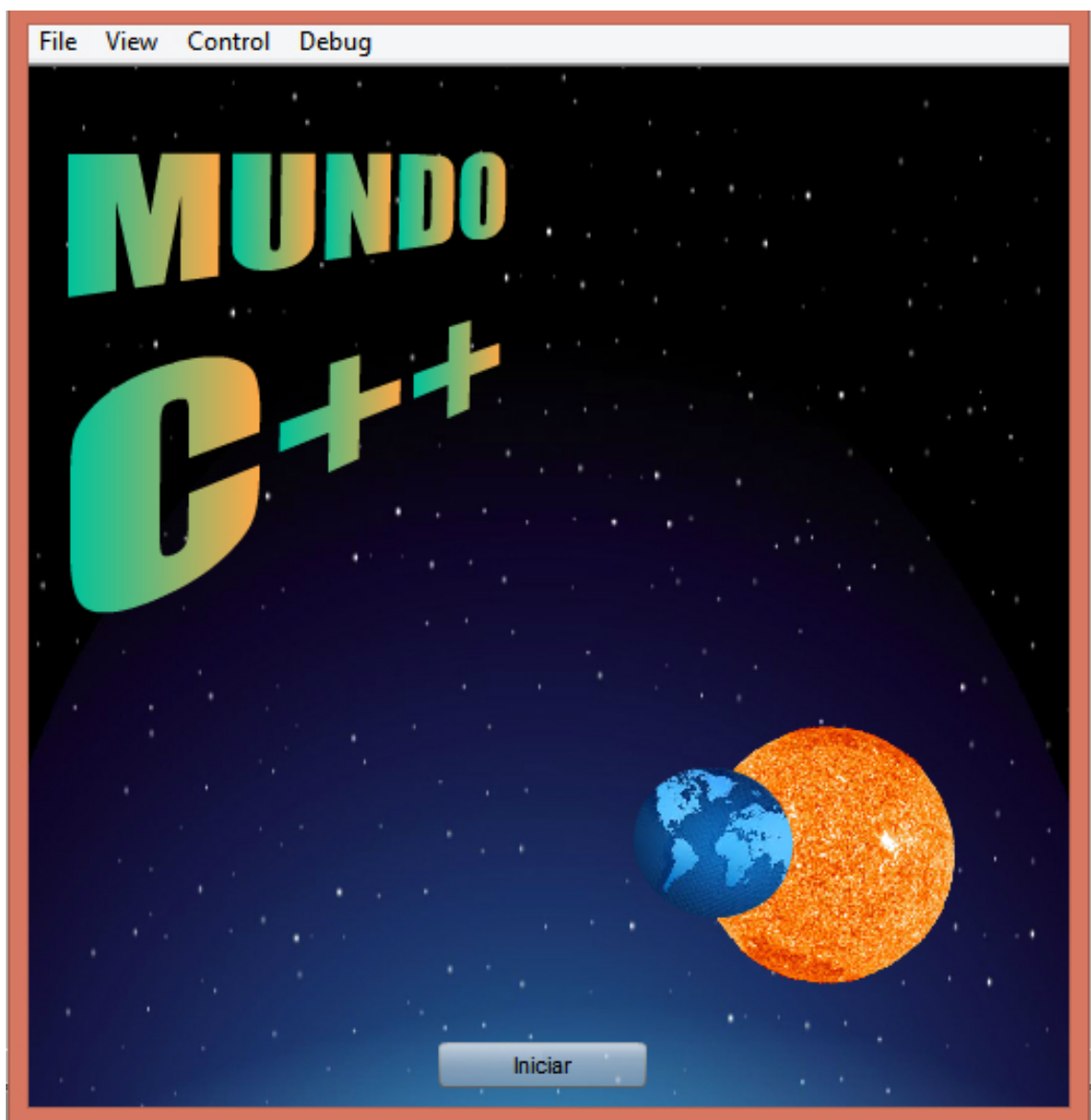


Figura 5 – Tela Inicial Mundo C++

A figura 6 apresenta a tela de ajuda do jogo. Ela objetiva mostrar ao jogador as características gerais do jogo.

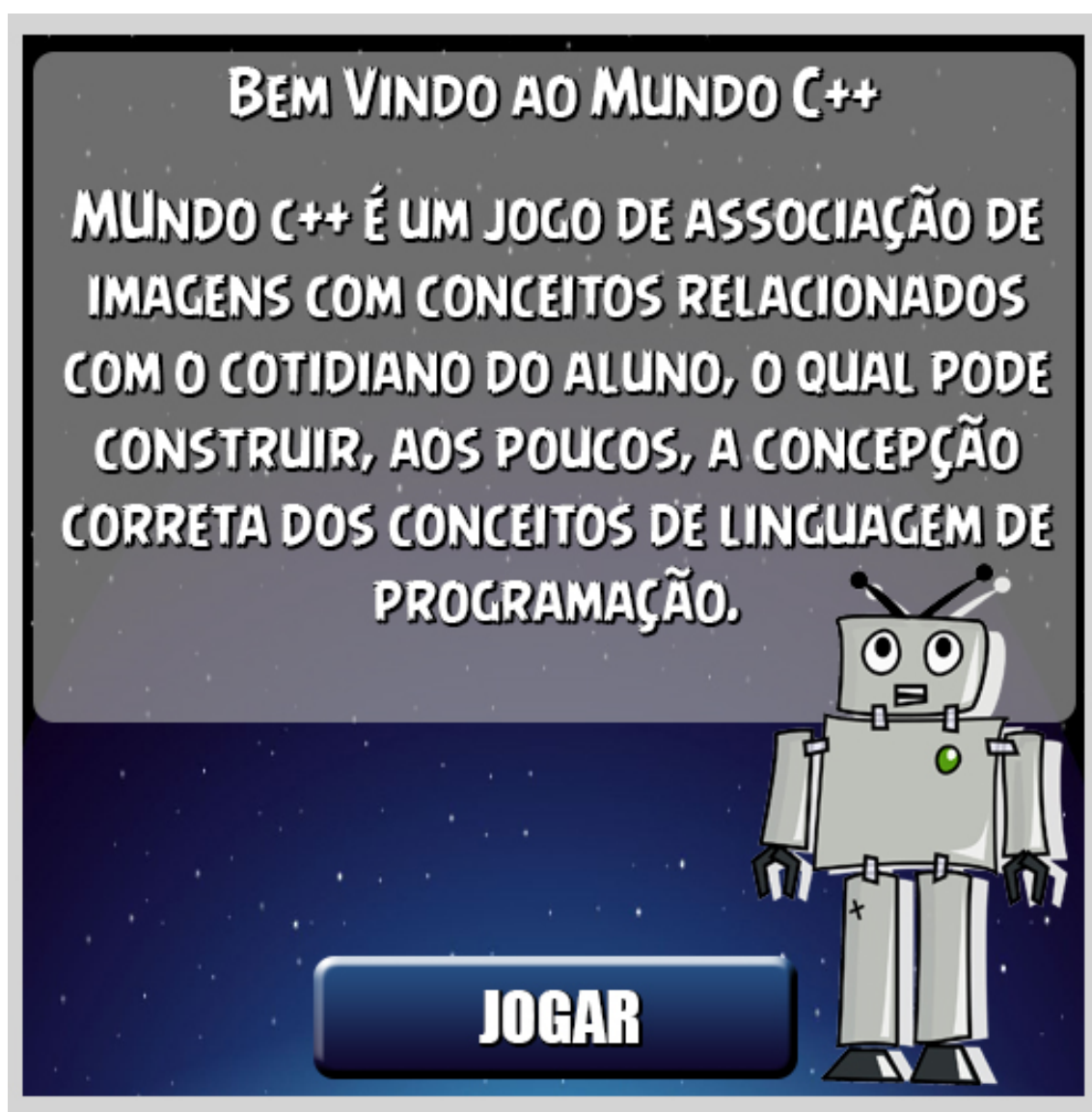


Figura 6 – Tela Explicativa Jogo

#### 5.1. Cenários

Os cenários do *Mundo C++* foram construídos com conceitos da linguagem de programação C++, o jogador terá que ler o conceito e escolher qual imagem

corresponde de uma melhor forma ao conceito. A cada novo cenário o jogo fica mais desafiador, com conceitos mais avançados e imagens que irão exigir do jogador melhores associações. A figura 7 requer, por parte do jogador, a associação entre o conceito de variável e a informação inserida num programa pelo usuário.



Figura 7 – Jogo de Associação de Imagens

Com uma jogabilidade simples mas desafiadora, Mundo C++ vai trazer para o jogador os conceitos de uma forma mais visual e agradável.



## **6. CONCLUSÃO**

Conclui-se que a definição proposta para jogo é de caráter sólido, ou seja, ensinar de uma forma mais visual e lúdica aos alunos que estão começando a aprender linguagem C++.

Por meio das pesquisas e análises feitas junto com os desenvolvedores do site Programadores de Jogos, foi possível notar que é de grande preocupação que o jogo não tenha o seu objetivo principal perdido, para isso este jogo possuirá uma mecânica simples, mais desafiadora. Todos os requisitos levantados para a construção dos cenários têm base no conteúdo programático que o professor da matéria de Linguagem C++ está desenvolvendo em sala.

As tecnologias que utilizadas para o desenvolvimento do Mundo C++ vêm sendo muito difundidas no mercado atual de jogos, isso fez com que o autor desse trabalho tivesse um grande crescimento e desenvolvimento profissional nas áreas de análise e programação de sistemas.

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

DERRYBERRY, Anne. "Serious games: online games for learning". I'm Serious.net. Disponível em <[http://www.adobe.com/products/director/pdfs/serious\\_games\\_wp\\_1107.pdf](http://www.adobe.com/products/director/pdfs/serious_games_wp_1107.pdf)> Acesso em: 24 de mar. de 2013.

GEE, James Paul. "Good vídeo games and good learning". Academic ADL Co-Lab. Disponível em <[http://www.academiccolab.org/resources/documents/Good\\_Learning.pdf](http://www.academiccolab.org/resources/documents/Good_Learning.pdf)> Acesso em: 24 de mar. de 2013.

HUIZINGA, Johan H. (1938). Homo Ludens - A Study of the Play-Element in Culture. Routledge.

IMPA, Instituto Nacional de Matemática Pura e Aplicada. (2007). Teoria dos Jogos Não Cooperativos. Disponível em <http://webold.impa.br/Disciplinas/Ementas/9.3.30.html>. Visitado em 29/10/2013.

LERNER, M. **Uma Avaliação da Utilização de Jogos em Educação**. Rio de Janeiro: COPPE/UFRJ,1991. (Oficinas de Informática na Educação).

CARVALHO, Flávia Pereira. Introdução ao ActionScript 3.0. Disponível em [https://fit.faccat.br/~fpereira/aula\\_flash/slides\\_aula\\_flash\\_actionscript3\\_1porfolha.pdf](https://fit.faccat.br/~fpereira/aula_flash/slides_aula_flash_actionscript3_1porfolha.pdf). Acesso em 04/12/2013.

FURLAN, Marcos Paulo. Flash AS3 básico. Disponível em <http://www.apostilando.com/download.php?cod=3323>. Acesso em 04/12/2013.