

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE INFORMÁTICA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS

PRISCILA BAER GOMES VIEIRA

**DESENVOLVIMENTO DE UM *SOFTWARE* MULTIMÍDIA
EDUCACIONAL PARA CRIANÇAS**

TRABALHO DE DIPLOMAÇÃO

PONTA GROSSA

2011

PRISCILA BAER GOMES VIEIRA

**DESENVOLVIMENTO DE UM *SOFTWARE* MULTIMÍDIA
EDUCACIONAL PARA CRIANÇAS**

Trabalho de conclusão de curso de graduação, apresentado à disciplina Trabalho de Diplomação, do curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Coordenação de Informática – COINF – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Tecnólogo.

Orientador: Prof. Dr. André Koscianski

PONTA GROSSA
2011



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Ponta Grossa
Diretoria de Graduação
Coordenação de Informática
Tecnologia em Análise e Desenvolvimento de Sistemas



TERMO DE APROVAÇÃO

DESENVOLVIMENTO DE UM *SOFTWARE* MULTIMÍDIA EDUCACIONAL PARA CRIANÇAS

por

PRISCILA BAER GOMES VIEIRA

Este Trabalho de Conclusão de Curso foi apresentado em 08 de novembro de 2011 como requisito parcial para a obtenção do título de Tecnóloga em Análise e Desenvolvimento de Sistemas. A candidata foi arguida pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

André koscianski
Prof. Orientador

Mônica Hoeldtke Pietruchinski

Lourival Aparecido de Góis

AGRADECIMENTOS

Agradeço primeiramente a Deus pela saúde e sabedoria concedidas para a realização deste trabalho.

Agradeço aos meus pais Geanini e Victor, minha irmã Daniele, pela força, incentivo, apoio e ajuda que dispuseram nas horas mais difíceis.

À minha família que dá sentido a vida, meu marido Marcelo pelo apoio, incentivo, carinho, paciência e força que sempre me passou para poder chegar ao fim do curso. Ao meu doce Vitor por ter aguentado firme essa trajetória.

A Denise por ter me proporcionado essa experiência com o mundo da animação, com suas fadas, bruxas e angiospermas, por além de ter sido “cliente” ter se tornado uma amiga e ao Welington por seus incríveis desenhos me inspirando para as técnicas de *design*.

Especialmente ao Prof. André Koscianski, que orientou com extremo profissionalismo e dedicação em todas as etapas deste grande desafio.

A Universidade Tecnológica Federal do Paraná, Campus Ponta Grossa, representado por sua diretoria, por proporcionar este curso de tão alto nível.

A todos os demais familiares, amigos e professores, que de uma forma ou outra, me apoiaram no desenvolvimento deste trabalho e/ou ao longo do curso acreditando no meu sucesso pessoal e profissional.

RESUMO

VIEIRA, Priscila B. G. **Desenvolvimento de um *software* multimídia educacional para crianças**. 2011, 47f. Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2011.

Este trabalho descreve o desenvolvimento de um *software* multimídia destinado ao ensino de Ciências para crianças entre 11 a 13 anos do 7º ano do ensino fundamental. São apresentados estudos sobre os recursos de multimídia e acerca da definição de uma interface utilizando conceitos de Interação Humano-Computador (IHC), para que venham facilitar a navegabilidade, usabilidade e interatividade de maneira eficiente e dinâmica. Traz como exemplo de utilização prática do *software*, o desenvolvimento de uma multimídia educacional, cujo conteúdo e formato geral são objetos de um projeto de mestrado no PPGECT (Programa de Pós-Graduação em Ensino de Ciência e Tecnologia) da UTFPR, campus Ponta Grossa. Envolve uma equipe multidisciplinar explorando as potencialidades de cada um para o desenvolvimento do aplicativo, resultando num *software* com qualidade técnica, tecnológica e pedagógica.

Palavras-chave: Multimídia. Interface humano-computador. *Software*.

ABSTRACT

VIEIRA, Priscila B. G. **Development of an educacional multimedia software for children**. 2011, 47f. Final Paper for a Course Conclusion – Systems Analysis And Development Technology, Federal Technological University of Paraná. Ponta Grossa, 2011.

This work describes the development of a multimedia software, applied to children Science learning for children between 11 to 13 years from 7th year of elementary school. Here are presented studies about the resources of multimedia and about the definition of an interface, using subjects of Human-Computer Interaction (HCI), to make easy, surf the net, uses and interactivity in an efficient and dynamic way. Has as example, practical uses of software, the development of an educational multimedia, which subject and general shape are subjects of a master project at PPGET (Post Graduation Program in Technology and Science Learning) of UTFPR, Ponta Grossa campus. Join a multi subject team, exploring the potencialities of each one to the development of a software, resulting a high quality technique, technologic and pedagogical software.

Keywords: Multimedia. Human-computer interaction. Software.

LISTA DE ILUSTRAÇÕES

Figura 1 – Eventos	17
Figura 2 – Animação	18
Figura 3 – Disposição dos quadros na navegação sequencial	20
Figura 4 – Disposição dos quadros na navegação hierárquica.....	20
Figura 5 – Disposição dos quadros na navegação em rede	21
Figura 6 – Diagrama de Navegação.....	24
Figura 7 – Ícones de navegação	26
Figura 8 – Biblioteca.....	27
Figura 9 – Cartão de visita	27
Figura 10 – Livro Semente	28
Figura 11 – Livro Fruto	29
Figura 12 – Castelo α Zen fundo alaranjado	30
Figura 13 – Castelo α Zen.....	30
Figura 14 – Eventos do mouse.....	31
Figura 15 – Eventos do mouse.....	32
Figura 16 – Batalha raízes tuberosas.....	33
Figura 17 – Código raiz tuberosa	34
Figura 18 – Sombrios sendo atacados.....	35
Figura 19 – Primeira versão Atividade Noções Gerais de Angiospermas	36
Figura 20 – Atividade Noções Gerais de Angiospermas	37
Figura 21 – Esboço da Atividade Raízes	38
Figura 22 – Esboço Instruções.....	38
Figura 23 – Protótipo Atividade Raízes	39
Figura 24 – Protótipo Cena Instruções.....	39
Figura 25 – Atividade Caule opções desabilitadas.....	41
Figura 26 – Atividade Caule	42
Figura 27 – Atividade Flor	43
Figura 28 – Atividade Flor descartada.....	44

LISTA DE SIGLAS

AVM	<i>Action Script Virtual Machine</i>
IHC	Interface Humano-Computador
JVM	<i>Java Virtual Machine</i>
POO	Programação Orientada a Objetos
PPGECT	Programa de Pós-Graduação em Ensino de Ciência e Tecnologia
TCC	Trabalho de Conclusão de Curso
UTFPR	Universidade Tecnológica Federal do Paraná
XP	<i>Extreme Programming</i>

SUMÁRIO

1 INTRODUÇÃO	10
1.1 PROBLEMA	11
1.2 OBJETIVOS.....	11
1.2.1 Objetivo Geral.....	11
1.2.2 Objetivos Específicos.....	11
1.3 ORGANIZAÇÃO DO TRABALHO.....	12
2 SOFTWARE MULTIMÍDIA EDUCACIONAL: UMA VISÃO GERAL.....	13
2.1 METODOLOGIA	13
2.2 FLASH	15
2.2.1 Máquina Virtual.....	16
2.2.2 Execução	16
2.2.3 Suporte a Animações Gráficas	18
2.3 INTERAÇÃO HUMANO-COMPUTADOR	19
2.3.1 Usuários não Habitados a Computador	21
3 DESENVOLVIMENTO DE UM SOFTWARE MULTIMÍDIA PARA CRIANÇAS	23
3.1 ESTRUTURA DE NAVEGAÇÃO	23
3.2 <i>DESIGN</i> ATRATIVO	25
3.2.1 Animações Gráficas.....	32
3.3 PROJETO DE INTERAÇÃO	35
3.4 DIFICULDADES DE IMPLEMENTAÇÃO.....	41
4 CONCLUSÃO	45
4.1 TRABALHOS FUTUROS	46

1 INTRODUÇÃO

O desenvolvimento na tecnologia da informação possibilita o armazenamento, manipulação e transmissão da informação de diversas maneiras, podendo fazer uso não somente do componente textual como também da manipulação de imagens, sons e movimento.

Atualmente muitos educadores vêm utilizando recursos tecnológicos com o objetivo de atrair mais a atenção dos alunos e facilitar a aprendizagem. Entre esses recursos destaca-se a multimídia.

Entende-se por multimídia o conjunto de textos, sons, gráficos, animações, imagens estáticas e em movimento, podendo ser acessados simultaneamente e objetivando a interatividade com o usuário. O termo hipermídia é muitas vezes usado para indicar que um dado produto multimídia permite a exploração de forma não linear, da mesma maneira como um hipertexto permite que o leitor navegue dentro de um texto.

O desenvolvimento e utilização de multimídia tem se tornado comum, sendo muito utilizado em propaganda, *sites* internet e em educação (GOUVEIA, 2004).

Pode-se dizer que a multimídia oferece um novo meio de comunicação, repleto de possibilidades a serem exploradas e acesso a informações em diferentes formatos com componentes visualmente atrativos e dinâmicos, "[...] a multimídia é um recurso objetivo, versátil, fácil e simples, que ajuda na revisão dos conteúdos, podendo ser utilizado no próprio domicílio do aluno, [...] trata-se de um recurso complementar às outras formas" (SANTIAGO, 2003, p.10).

Existem diferentes maneiras de implementar um aplicativo multimídia. A utilização do Flash como ferramenta de autoria no desenvolvimento de *softwares* multimídia facilita a manipulação direta da animação, por possuir linguagem de programação capaz de controlar as ações que corresponderem às expectativas e solicitações do usuário sobre a interface (PAULA FILHO, 2000).

Muitos *softwares* multimídia, como jogos, são utilizados por crianças. Isso traz uma preocupação especial quanto à qualidade de interfaces, facilidade de uso e navegabilidade.

Este trabalho propõe a produção de um *software* multimídia educacional com conteúdo voltado para o ensino de Ciências do 7º ano do ensino fundamental. O projeto está vinculado a uma dissertação de mestrado no PPGECT.

O desenvolvimento desse *software* busca colocar em prática conceitos adquiridos na trajetória do curso, bem como propõe desafios ao envolver áreas que não são vistas dentro da grade curricular, como design gráfico, o desenvolvimento de jogos e a computação gráfica. Dentre os problemas envolvidos no projeto, fez-se necessário atender uma realidade específica, considerando o público a que se destina (crianças), as dificuldades encontradas pela cliente (professora), o meio em que será empregado (colégio) e os recursos disponíveis (computador, Internet, leitor de CD).

1.1 PROBLEMA

O problema central do trabalho pode ser sumarizado em projetar e implementar um *software* multimídia adequando a interface gráfica de acordo com critérios de ergonomia e com critérios impostos por professor e alunos.

O *software* proposto busca testar a hipótese de que a interface pode garantir o sucesso do aplicativo, ao atrair a atenção da criança, motivando-a a descobrir o que este oferece.

1.2 OBJETIVOS

A presente pesquisa apresenta os seguintes objetivos:

1.2.1 Objetivo Geral

Desenvolver um *software* multimídia com interface amigável e intuitiva para ser utilizado como ferramenta de apoio ao ensino de angiospermas para crianças com idade entre 11 e 13 anos.

1.2.2 Objetivos Específicos

- Projetar o *software*, junto à equipe multidisciplinar, buscando uma interface adequada aos usuários.
- Verificar a possibilidade de incluir jogos dentro do *software*.

- Implementar o *software*.

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho está dividido em quatro capítulos. O capítulo 1 apresenta a introdução. No capítulo 2 são apresentados os estudos efetuados durante o desenvolvimento do trabalho e conceitos relacionados ao *software* multimídia.

As etapas de desenvolvimento do *software* multimídia e a metodologia adotada fazem parte do capítulo 3.

O capítulo 4, por sua vez, apresenta as conclusões finais sobre o trabalho, apontando possibilidades para atividades futuras e sugestões que darão continuidade ao *software*.

2 SOFTWARE MULTIMÍDIA EDUCACIONAL: UMA VISÃO GERAL

Este capítulo apresenta uma visão geral dos tópicos envolvidos no desenvolvimento do aplicativo. Isto inclui as principais características que um *software* multimídia educacional deve apresentar, bem como a escolha de ferramentas de desenvolvimento.

2.1 METODOLOGIA

Para o desenvolvimento e gerenciamento de projetos de *software* existem metodologias ágeis tais como *Extreme Programming* (XP), Scrum, Crystal, DSDM, entre outras, sendo as duas primeiras as mais conhecidas (KOSCIANSKI e SOARES 2007).

A metodologia de desenvolvimento ágil não deve ser aplicada a sistemas críticos que necessitem de análise mais detalhada dos requisitos, com maior chance de retrabalho e implicações na questão de segurança (SOMMERVILLE, 2007).

É definida a escolha de uma metodologia pela adaptação de seus métodos com relação ao *software* que se deseja produzir.

Neste projeto foi utilizada a metodologia XP, por possuir um conjunto de princípios mais adequado ao desenvolvimento do *software*. Esta metodologia é utilizada em equipes pequenas e médias, proporcionando um relacionamento de comunicação contínua entre os integrantes.

Segundo SOMMERVILLE (2007), alguns princípios são:

- Envolvimento do cliente no processo;
- Entrega por incremento;
- Foco nas pessoas, não em processos;
- Mudanças são aceitas;
- Simplicidade do *software*.

Com o cliente envolvido durante o processo de desenvolvimento, a troca de informações sobre as funcionalidades que o aplicativo deve ter, as características que espera obter são mais facilmente identificadas e requisitos não-conformes serão verificados e corrigidos mais rapidamente. Com isto o cliente obterá frequentemente

uma parte funcional do *software* para estar avaliando (KOSCIANSKI e SOARES 2007).

Os desenvolvedores podem estar mais à vontade para trabalhar não precisando seguir processos prescritivos. A proximidade do cliente no desenvolvimento pode ocasionar mudanças nos requisitos para obter melhorias no *software*. Simplicidade na criação de código e funcionalidades, evitando implementações que só serão significativas no futuro (KOSCIANSKI e SOARES 2007).

A XP baseia-se em alguns princípios ou práticas que são aplicadas gradativamente. Segundo KOSCIANSKI e SOARES (2007):

O planejamento deve levar em conta os requisitos a serem atendidos observando o prazo de entrega.

As entregas frequentes possibilitam avaliações pelo cliente, sugerindo pequenas alterações, evitando grandes surpresas no final do desenvolvimento como diversas mudanças no *software*.

Utilização de metáfora para evitar uso de termos técnicos, descrevendo de maneira a conduzir o desenvolvimento do *software*.

O projeto deve ser simples, tendo em mente a funcionalidade imediata.

Devem ser realizados testes durante todo o processo para validar o aplicativo.

Programação em pares facilita na identificação de erros e a troca de informações traz soluções mais rápidas e simples.

Refatoração busca o aperfeiçoamento do projeto e assim que se consegue aprimorar o código, este se torna simples e fácil de manter.

Na propriedade coletiva todos buscam melhorar e isto possibilita a alteração do projeto por qualquer um dos integrantes da equipe.

Tão logo uma etapa do desenvolvimento termine, isto é, tenha passado pelos testes e seja aprovado, deve ser integrado ao sistema e este também deverá passar por testes.

Trabalhar mais de 40 horas semanais não deve ser visto como normal ou aceitável, deve rever o planejamento para não sobrecarregar os integrantes do desenvolvimento.

O cliente deve estar disponível durante todo o projeto, podendo até ser parte da equipe. Com isto evita-se atrasos e dúvidas durante o desenvolvimento.

Código padronizado melhora o entendimento e a visualização com regras para o estilo e formato de comentários.

2.2 FLASH

Flash é uma tecnologia criada pela empresa FutureWave e levava o nome de *Future Splash Animator*, que após ser comprada e alterada pela Macromedia passou a se chamar Flash¹. Tem grande popularidade no desenvolvimento dinâmico de *softwares* multimídia e utilização de conceitos da POO (Programação Orientada a Objetos) por meio da linguagem de programação *Action Script*. Pode-se dizer que Flash facilita a criação de um *software* ao proporcionar um design atraente e interativo com o usuário.

A tecnologia Flash também permite o desenvolvimento de animações multimídia, que utilizam texto, imagem estática e em movimento, vídeo, som, animações, de forma coerente, respeitando o modo de relacionamento do aluno com a informação apresentada.

A POO modela *softwares* de maneira similar a objetos do mundo real, onde cada objeto contém atributos e comportamentos (DEITEL, 2005). A linguagem *Action Script* é utilizada para criar aplicações Flash, permitindo controlar o comportamento de imagens na tela que correspondem a objetos no código (SANTOS, 2010).

Existem várias ferramentas de desenvolvimento que produzem aplicativos Flash. Pode-se citar Adobe Flash *Player*, *Swift*, *KoolMoves*, *MaxMedia Standard*. Para este projeto foi escolhida a ferramenta SWISH Max versão 3.0. Essa escolha é justificada em função do custo, da facilidade de uso e bom número de recursos.

A linguagem de programação utilizada é a *Swish Script* que, como o próprio nome sugere, é uma linguagem própria da ferramenta, porém, semelhante a *Action Script* pertencente à Adobe Flash *Player* (SANTOS, 2010). Trata-se de uma linguagem de fácil aprendizado devido à sua semelhança a *JavaScript* – linguagem utilizada para programação *web*.

¹ (WIKIPÉDIA, 2011)

2.2.1 Máquina Virtual

As aplicações Flash executadas dentro de navegadores internet usam a mesma solução criada anteriormente pela linguagem Java. Para compreender é interessante analisar a máquina virtual Java.

A JVM, no inglês *Java Virtual Machine*, é um interpretador que executa um código de máquina portátil, conhecido como “*bytecodes*”. Esse código é gerado pelo compilador Java (DEITEL, 2005). Sendo assim, os *softwares* criados a partir da JVM têm portabilidade, ganham independência de sistema operacional, podendo ser executados em qualquer computador que contenha uma JVM.

Pode-se dizer que funciona como um conjunto de instruções que seguem um roteiro. Essas instruções são executadas pela máquina com o propósito de resolver problemas, melhorar desempenho e com isso reduzir o uso de memória. “[...] Como o próprio nome diz, uma máquina virtual é como um ‘computador de mentira’: tem tudo que um computador tem. Em outras palavras, ela é responsável por gerenciar memória, *threads*, a pilha de execução, etc”².

O código executado pela JVM não possibilita acesso direto ao *hardware* do computador. Dessa forma, a JVM garante proteção contra códigos maliciosos, além de permitir executar um programa em qualquer plataforma.

Para os programas desenvolvidos em Flash a máquina virtual é *ActionScript Virtual Machine* (AVM) que, ao transformar o código em *bytecode*, este é então incorporado nos arquivos de formato Flash representada pela extensão *.swf*³.

2.2.2 Execução

Existem dois tipos de interfaces conhecidas, a textual e a gráfica. Na interface textual o usuário deve saber quais comandos o programa faz a leitura para que sua solicitação seja atendida. Já a interface gráfica disponibiliza ações através do pressionamento do teclado ou mouse em elementos gráficos. Estas ações são

²FJ-11: Java e Orientação a Objetos. Disponível em <<http://www.caelum.com.br/apostilas>> Acesso em: 20 de Set. 2011. p. 5.

³Introduction to ActionScript 3.0. Disponível em <http://help.adobe.com/en_US/as3/learn/WSf00ab63af761f170-43fa6dce12937d272e9-8000.html> Acesso em: 10 de Out. 2011.

chamadas eventos, o programa responde aos eventos de mouse e teclado que o usuário fizer, dentro de um laço (KOSCIANSKI, 2011).

No início do projeto, são definidas algumas propriedades como o tamanho em pixels que a aplicação terá na tela, a cor de fundo, e o número de *frames* por segundo que ficará padrão para todas as cenas (SOUSA, 2011).

Para o desenvolvimento do código *Swish Script* existe na própria ferramenta Swish Max associado à cena em que estiver trabalhando um editor de *scripts*. Neste editor pode-se estar trabalhando com a cena em si, individualmente com os elementos que a compõem ou estar associando uma ação a um *frame* específico.

A figura 1 ilustra um exemplo: o evento *onPress* será acionado pelo usuário quando este pressionar o botão do mouse num objeto clicável. No código o objeto é representado pelo evento *onPress*. Toda vez que o usuário pressionar este objeto é gerado um evento do mouse e a rotina *onPress* será disparada. Já o objeto que tiver o evento *onEnterFrame* a ele associado, será executado a cada novo *frame* de animação. Como já explicado isso acontece dentro de um *loop* infinito.

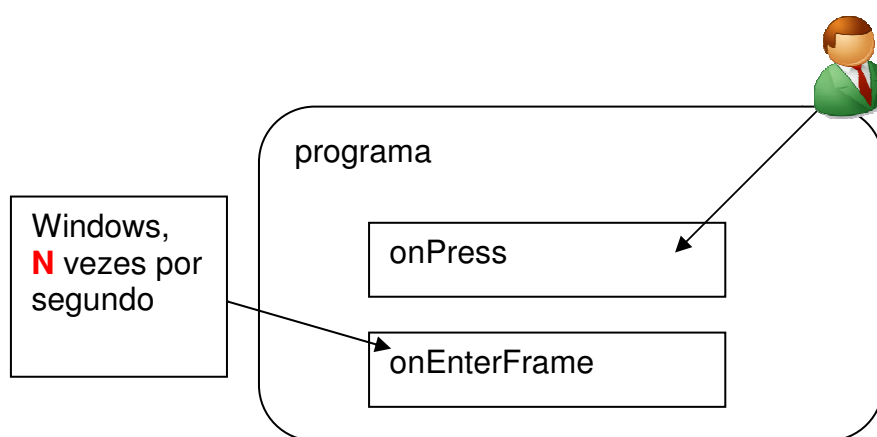


Figura 1 – Eventos
Fonte: Koscianski

Em relação aos eventos relacionados ao mouse fazem parte:

- *dragOver* - quando o cursor do mouse é pressionado e está em movimento sobre o objeto,
- *dragOut* - quando o cursor do mouse está pressionado e é movido para fora do objeto clicável,
- *releaseOutside* - quando o cursor do mouse pressionado sobre o objeto é movido para fora e solto.

Fazem parte dos eventos de *frames*:

- *onFrame*, em que a ação só será iniciada quando atingir o *frame* especificado;
- *onKeyPress* que trata o teclado.

Alguns botões do cenário ativam rotinas para controlar a sequência do aplicativo: *gotoAndPlay* que pode direcionar a outra cena e iniciar a animação, *gotoAndStop* direciona para outra cena e fica parada nela até obter outro evento acionado, que pode ser pelo usuário ou por um *frame* que contém um evento.

2.2.3 Suporte a Animações Gráficas

Um dos objetivos da tecnologia Flash é criar interfaces com elementos gráficos. Assim, no *software* Swish Max existem vários recursos que facilitam a manipulação de imagens na tela. Um desses recursos é a biblioteca de animações dentro do programa.

O Swish Max fornece animações como:

- mover um objeto entre dois pontos;
- rotacionar um objeto no mesmo lugar;
- piscar um objeto;
- fazer um objeto percorrer um caminho, determinado por pontos pelo usuário.

Para usar essas animações, basta selecionar um objeto no cenário e associar a ele a animação desejada. A figura 2 mostra um exemplo.

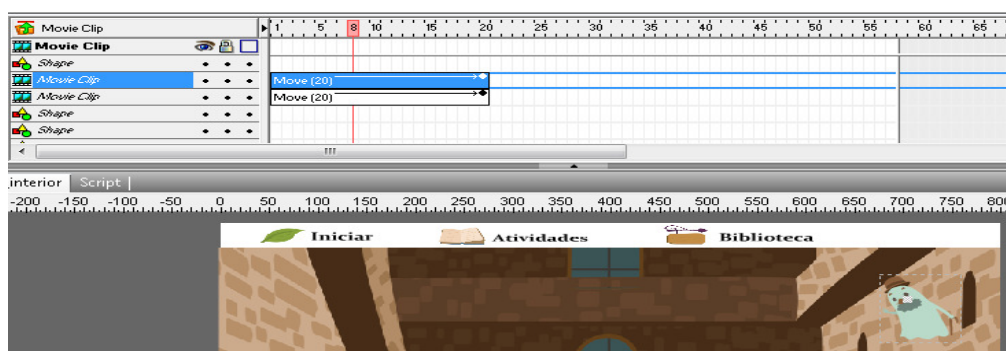


Figura 2 – Animação
Fonte: Autoria Própria

Na figura 2 há dois objetos com animações associadas. O objeto fantasma, que aparece no canto do vídeo, se move automaticamente pelo cenário. Isso é configurado na linha de tempo, que aparece na parte superior da janela. O efeito “move” está configurado para executar do *frame* 1 até o *frame* 20. Para mudar esse período basta arrastar o efeito com o mouse. Também é possível trocar o efeito por outro, como piscar. Isso é feito usando o botão direito do mouse e selecionando outro efeito da biblioteca.

2.3 INTERAÇÃO HUMANO-COMPUTADOR

A interação humano-computador vem sendo tratada dentro de IHC, cabendo a ergonomia de interfaces tratar de aspectos objetivos e subjetivos da interação entre pessoas e computadores.

Cabe a ergonomia de interfaces estabelecer formas de representação e apresentação da informação, facilitando essa interação humano-computador além de estruturar o *layout* da interface (PREECE; ROGERS; SHARP, 2005), de forma que a exposição da informação aos alunos torne o conhecimento viável e com uma comunicação atraente.

Para envolver e motivar os alunos se torna fundamental uma interface amigável, ou seja, tenha uma linguagem e um design de interação de acordo com a idade do público alvo, para que instigue o aluno na busca pelo aprendizado, ou seja, que ele se torne “... o centro do processo de aprendizagem, em que a autonomia da aprendizagem seja estimulada e que vise a desenvolver habilidades cognitivas associadas às diversas áreas do conhecimento” (FALKEMBACH, 2005, p. 1).

O uso de computadores no ensino pode proporcionar um ensino diferente do tradicional, potencializando o aprendizado do aluno, tornando esse processo dinâmico e atraente através da integração dessa tecnologia ao ensino. A utilização do computador permite apresentar o conteúdo de forma não linear, ajudando no processo de aprendizado do aluno, por meio de um visual dinâmico e dialogante, oportunizando uma interação entre ele e o computador.

Um dos pontos importantes do projeto, por ser um *software* educativo, é analisar qual estrutura de navegação atende melhor. A seguir são listadas algumas possibilidades.

Uma primeira ideia é a navegação sequencial, onde as telas seguem uma ordem linear, conforme ilustrado na figura 3. Neste caso a comunicação só fará sentido se não houver desvios.

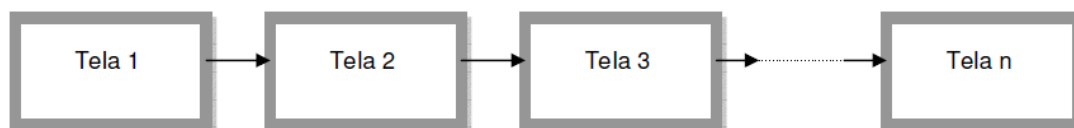


Figura 3 – Disposição dos quadros na navegação sequencial
Fonte: Falkembach (2005, p. 10)

Uma navegação que pode atender melhor ao tipo de comunicação é a da estrutura hierárquica também conhecida como *top-down*, onde a sequência da navegação segue de cima para baixo, conforme se observa na figura 4. Após o início tem-se a alternativa de escolher entre dois ou mais caminhos. Porém deve-se observar o sentido das flechas, que seguem um único sentido e após escolhido o caminho não há retorno.

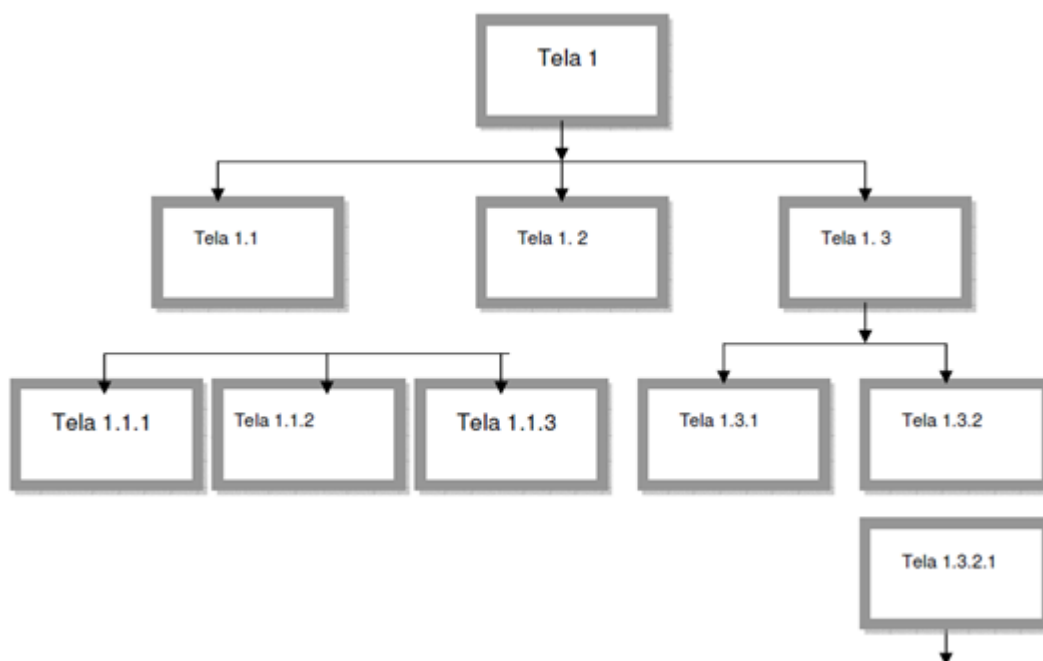


Figura 4 – Disposição dos quadros na navegação hierárquica
Fonte: Falkembach (2005, p. 11)

Neste outro tipo de estrutura, representada na figura 5, é apresentada a navegação *web* onde não existe uma ordem definida, esta ordem é estabelecida no momento da utilização. Se as telas não estivessem numeradas não teria como se

saber por qual delas a execução iniciou. Partindo da tela nomeada como Tela 1 pode-se decidir qual caminho fazer não precisando seguir um sentido linear nem uma hierarquia, tendo como opção pular para outra tela ou retornar.

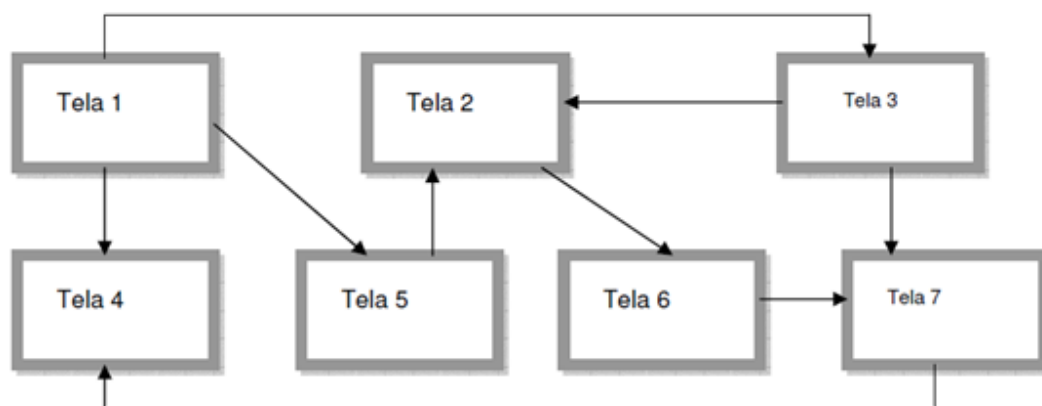


Figura 5 – Disposição dos quadros na navegação em rede
Fonte: Falkembach (2005, p. 11)

Como nesta estrutura há maior autonomia, podendo decidir qual caminho fazer, considera-se que, nesta comunicação visual, as informações devem estar dispostas de maneira clara e objetiva, de tal forma que, independente de qual seja o caminho escolhido, ele faça sentido, facilitando a interação (ANTUNES et al., 2008).

Mesmo o aluno tendo esta liberdade de escolher o caminho a seguir, existem certas restrições para que ele possa aprender o conteúdo necessário nesta interação com o aplicativo (FALKEMBACH, 2005). O objetivo dessas restrições é evitar confusão para o estudante pelo fato de haver muitas opções, ou guiá-lo tanto ao longo da leitura quanto do aprendizado de um conceito.

2.3.1 Usuários não Habitados a Computador

Ao iniciar o projeto, sabia-se que algumas crianças tiveram pouco ou nenhum contato com computador. A disciplina que faz mais relação com o usuário e a utilização do computador é a IHC. Ela vem ao auxílio proporcionando um campo multidisciplinar envolvendo disciplinas como ciência da computação, psicologia cognitiva, ergonomia e design, objetivando o desenvolvimento de aplicações

funcionais (PREECE; ROGERS; SHARP, 2005) que é importante quando se deseja produzir um *software* educacional.

Para o desenvolvimento do *software* existem vários itens a serem considerados, como por exemplo, critérios de ergonomia, pensando-se principalmente que os utilizadores do *software* são crianças. Uma vez que o programa visa ao aprendizado, é importante dosar a quantidade de informação exposta no espaço e no tempo, evitando uma sobrecarga cognitiva. Deve-se então, segmentar textos, narrativas e animações (ZANOTTO et al., 2011).

A qualidade das imagens, padrão de cores, tamanhos de textos, legibilidade e harmonia vídeo e áudio também são considerados aspectos ergonômicos que precisam ser analisados com critério no momento em que o professor escolher um *software* ou quiser projetá-lo e implementá-lo com fins educativos.

Para auxiliar a definição da interface pode-se empregar listas de critérios (SCAPIN; BASTIEN, 1997; CATAPAN; FIALHO, 2000). O projeto e avaliação de um produto multimídia pode ser organizado em três partes (ROMISZOWSKI, 2000): a) conteúdo, em que será verificado se o material tem qualidades como clareza, consistência e utilidade; b) roteirização, que envolve definir a navegação dentro do *software* a distribuição de conteúdos nele; e c) testagem, em que se busca coletar as opiniões e impressões de usuários para corrigir eventuais falhas do produto e torná-lo o mais eficiente possível.

Em relação ao conteúdo didático, algumas diretrizes básicas são (TAVARES, 2008): a) pertinência: o material incluso no produto deve ter relação direta com o conteúdo tratado; b) relevância: deve-se selecionar os textos, imagens e animações mais apropriados, ou criar conteúdo focado no objetivo da aula; c) unicidade: esse critério deverá possibilitar a obtenção de respostas para a pergunta “em que medida são aproveitadas as qualidades únicas do computador como meio de acrescentar valor aos meios tradicionais de aprendizagem?”

Outro item a ser considerado diz respeito à avaliação do *software*. Utiliza-se da engenharia de *software* para avaliação de desempenho no *software* enquanto código e da engenharia de usabilidade na avaliação em relação ao diálogo com o utilizador através da interface (DAMASCENO, 2004).

3 DESENVOLVIMENTO DE UM SOFTWARE MULTIMÍDIA PARA CRIANÇAS

Conforme apresentado na introdução, o programa foi desenvolvido por uma equipe. Fizeram parte dela a aluna de mestrado Denise Zanotto, o orientador da dissertação e também orientador deste TCC e um desenhista, Welington Guimarães.

Coincidindo com a abordagem apresentada em (ROMISZOWSKI, 2000), o projeto começou com a definição de conteúdos didáticos, seguida do projeto de roteiros e de testes. A definição de conteúdos e sua apresentação, o que inclui textos e imagens, coube principalmente à aluna de mestrado, responsável principal pelos critérios didáticos usados no *software*. A definição de roteiros foi realizada conjuntamente entre a programadora e a aluna de mestrado, em função das possibilidades de implementação e das necessidades que foram elencadas para o produto. Finalmente, houveram testes práticos com estudantes durante o desenvolvimento do *software*, para verificar sua qualidade técnica e didática.

O programa foi desenvolvido gradualmente, cena por cena. Cada parte pode ser trabalhada de maneira independente das demais.

A seguir são apresentados mais detalhes sobre o desenvolvimento do *software* proposto.

3.1 ESTRUTURA DE NAVEGAÇÃO

Com as estruturas de navegação já vistas, a que foi escolhida para o desenvolvimento do *software* foi a navegação *web*, onde o usuário tem uma página inicial definida mas, quem determina qual caminho seguir é ele, e seja qual caminho escolhido, ele tem a opção de retornar.

A seguir foi elaborado um diagrama de navegação do *software* ilustrado na figura 6.

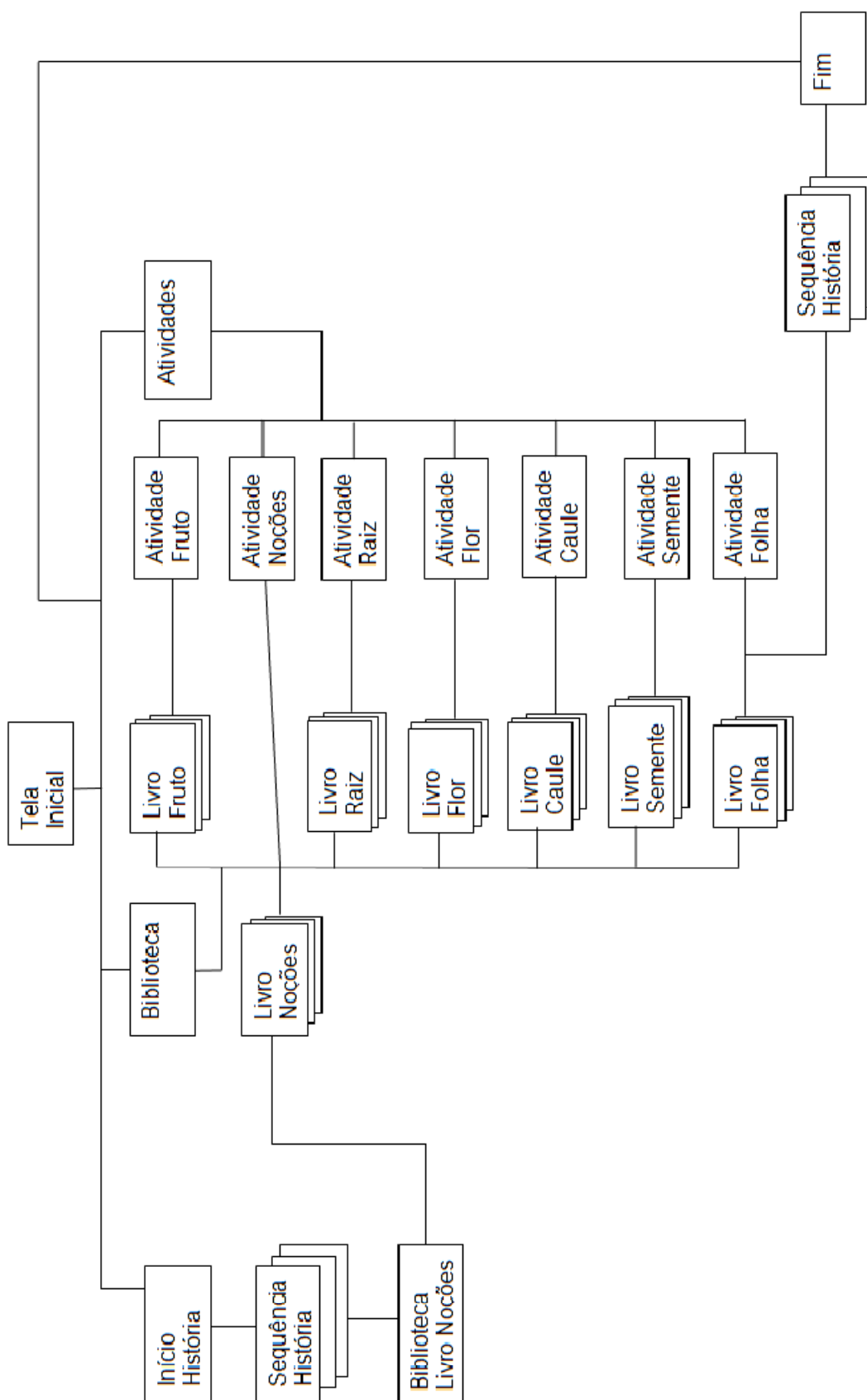


Figura 6 – Diagrama de Navegação
 Fonte: Autoria Própria

Uma possibilidade discutida para implementar a navegação, seria o uso de um menu em cascata. Esse menu seguiria a mesma estrutura mostrada na Figura 6. Entretanto, decidiu-se não usar, pois poderia ser um obstáculo ao invés de um facilitador agrupando as opções. O menu pode ser:

- dinâmico, ou seja, ao passar o mouse no ícone Biblioteca apareceria abaixo dele os títulos dos livros existentes;
- ou *pull down*, quando o usuário clica no ícone, aparece a lista de opções abaixo dele (NIELSEN; LORANGER, 2007).

Menus com este aspecto para usuários inexperientes poderiam trazer dificuldades em controlar o mouse e não entender seu funcionamento.

Com as opções visíveis o tempo todo ou que atraia sua atenção, as crianças entenderão e lembrarão mais facilmente e poderão decidir melhor qual caminho seguir. Sendo mais rápido chegar à informação desejada

No caso da biblioteca, após clicar no ícone que a representa, o usuário é direcionado para a cena de uma biblioteca. Na cena alguns livros estão dispostos na mesa e outros na estante, ao passar o cursor pelo livro que tem sua lombada diferenciada por conter listras, o livro “sai” da estante e seu tamanho é aumentado para melhor visualização de seu título.

Foi percebido nos testes que o fato das crianças procurarem o livro na estante, associar a cor do livro com o assunto, tornou atrativo e fez com que gostassem muito desta interatividade. Tanto que após terminarem de ver todas as cenas e realizar as atividades, retornavam à biblioteca. Gostaram de explorar esta cena e descobrir os livros na mesa e na estante.

3.2 DESIGN ATRATIVO

Uma das preocupações da interface é como deixá-la atrativa, com conteúdo bem distribuído, harmonioso sem poluir visivelmente o cenário. Para este *design* visual foram observadas técnicas de navegabilidade (NIELSEN; LORANGER, 2007), proximidade, alinhamento, repetição, contraste e tipografia (WILLIAMS, 1995).

A interface do programa é baseada no uso de mouse. Foram definidos diversos ícones, colocados na parte superior da janela. Cada ícone tem um rótulo de texto para ajudar na compreensão.

O botão *Iniciar* é representado por uma folha, o livro aberto representa as atividades, a tabuleta indica ir para a biblioteca, enquanto a porta representa a saída do aplicativo. No canto inferior esquerdo e direito encontram-se respectivamente, o retorno e avanço à outra cena, representados por uma flor, como mostrados na Figura 7.



Figura 7 – Ícones de navegação
Fonte: Guimarães (2011)

Os ícones foram dispostos de uma maneira linear propositalmente. Isso faz com que o aluno percorra visualmente a sequência de ícones, ao mesmo tempo em que pode ler os rótulos sob cada um, para saber quais as opções disponíveis. Seguindo essa ideia, o ícone iniciar é o primeiro à esquerda, seguindo a ordem de leitura (da esquerda para a direita), já o ícone de saída é posicionado ao final da lista, na extrema direita. Esta organização não define a sequência de ações, porém este alinhamento define uma conexão visual com as cenas. Apresentando a maneira que possibilita mover-se pelo aplicativo (WILLIAMS, 1995).

Na definição dos ícones, considerou-se a consistência dos mesmos, aparecendo sempre na mesma posição (KOSCIANSKI e SOARES 2007), mesmo em algumas cenas, como a própria biblioteca, ilustrada na figura 8, o ícone que a representa não estivesse presente, por estar no cenário biblioteca. Mesmo assim o seu espaço foi preservado, não aproximando os outros ícones por sua ausência. As figuras que os representam tem relação com a matéria de Ciências, flores, folhas e com objetos reais representados no *software*, ícone porta para “Sair”, caderno representando “Atividades”.



Figura 8 – Biblioteca
Fonte: Guimarães (2011)

O botão *Sair* presente em todas as cenas permite ao aluno finalizar o aplicativo no momento que desejar, sem que necessite ficar indo para outras cenas e efetuando uma série de “cliques” (FALKEMBACH, 2005).

A proximidade entre os elementos de interface forma a unidade visual, relaciona os assuntos de maneira organizada para não acarretar cansaço visual, nem a falta de percepção de itens importantes presentes na cena. Com início bem definido, consegue-se atrair a atenção e favorecer a memorização da leitura. Pode-se ilustrar isso com um contra-exemplo. Na figura 9 o olhar percorre por todos os lados do cartão para reunir as informações, não apresentando um início nem um fim bem definido (WILLIAMS, 1995).

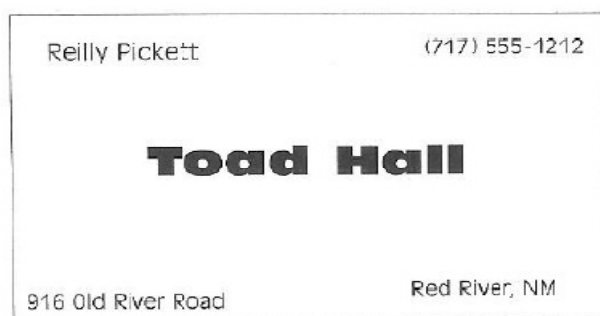


Figura 9 – Cartão de visita
Fonte: Williams (1995, p. 16)

Nas cenas onde havia somente conteúdo, como apresentado pela figura 10, este conceito foi bastante empregado para organizar os assuntos relacionados e colocar uma separação para os itens não deixando o texto num bloco único, facilitando a leitura para que não se tornasse cansativa. O próprio espaço em branco confere uma pausa na leitura, um descanso para os olhos (WILLIAMS, 1995). Por estar em formato de livro a criança já deduz a sequência da leitura, isto quer dizer que, o início é bem definido e por meio do sumário já pode saber quais assuntos serão abordados e quantas páginas o livro possui.

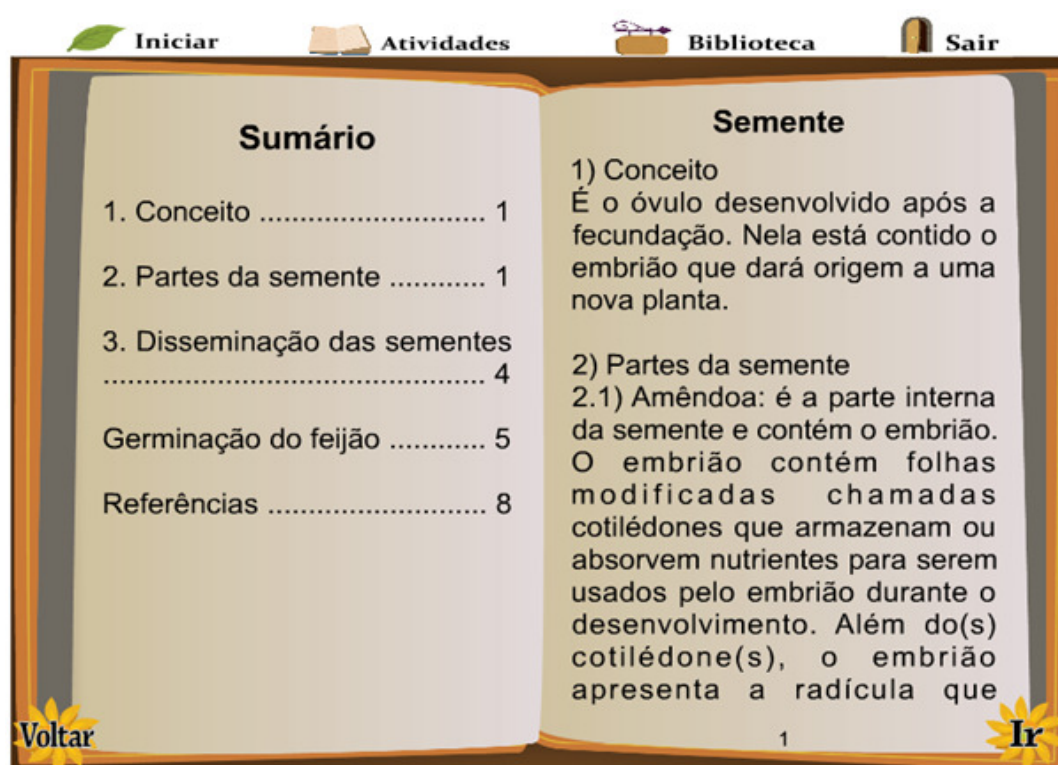


Figura 10 – Livro Semente
Fonte: Guimarães (2011)

Quando se fala 'alinhamento', quer dizer a disposição em que os elementos estão colocados na cena. De acordo com Williams (1995, p. 27, grifo do autor) **“...nada deve ser colocado arbitrariamente em uma página. Cada item deve ter uma conexão visual com algo na página”**.

A linha mesmo que invisível deixa o texto organizado, o título centralizado indica o assunto a ser tratado e o texto com o parágrafo justificado cria um alinhamento visual, demonstrando organização e clareza, transmitindo com isto uma

sensação de segurança. Na Figura 11 é evidenciado que até mesmo as imagens devem estar alinhadas com sua legenda (WILLIAMS, 1995).

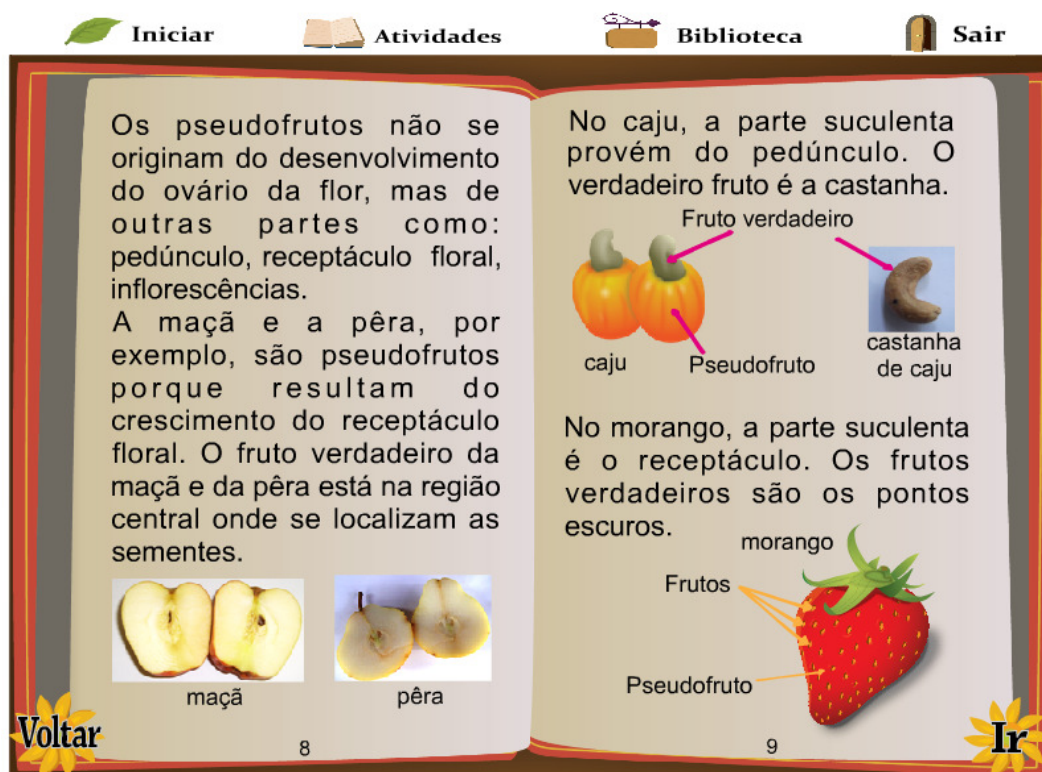


Figura 11 – Livro Fruto
Fonte: Guimarães (2011)

Assim como os ícones são importantes para a navegabilidade também o são para criar o aspecto consistente do aplicativo. Assim, os ícones são repetidos em todas as cenas pertencendo ao grupo de elementos-chave que transmitem uma aparência coesa.

Também fazem parte do grupo o sumário no início de cada livro, a numeração de páginas destes, o tipo de letra utilizada e as referências (WILLIAMS, 1995). A repetição dos ícones confere certa segurança, sobretudo para alunos pouco habituados com o uso de computador. Embora um aluno nem sempre saiba exatamente o que o espera na próxima cena, existem itens conhecidos que o deixarão mais à vontade para prosseguir, pois entende como as opções estão dispostas (KOSCIANSKI e SOARES 2007). Isso permite também que o aluno se concentre no conteúdo a ser exposto e não precise pensar muito cada vez que precisar acionar a interface do programa.

Para atrair a atenção não bastam somente cores, é preciso ter contraste. Conforme Williams (1995), quando dois elementos não se diferem muito, não se pode chamar de contraste, mas sim de conflito. Um exemplo disso aparece nas próximas imagens.



Figura 12 – Castelo α Zen fundo alaranjado
Fonte: Guimarães (2011)

Na Figura 12, o laranja predominou na cena e assim acabou não sendo tão atraente. O conjunto céu, sol e nuvens alaranjadas ficaram conflitantes na cena, pois a cor é a mesma, porém o tom é diferente.



Figura 13 – Castelo α Zen
Fonte: Guimarães (2011)

Na figura 13 o contraste harmonizou, conferindo ao sol o seu devido destaque diferenciando-o do céu e das nuvens, deixando a imagem mais

ergonômica do ponto de vista visual. Esse tipo de contraste também pode ser utilizado na tipologia que tem como definição ser o “conjunto de características gráficas ou estruturais de um texto” (BUENO, 2007, p. 756), ou seja, a aplicação dos tipos (fontes), incluindo o tamanho, característica como negrito, itálico ou sublinhado e posicionamento que as letras terão no aplicativo, diferindo de tipografia que se preocupa com as formas que serão impressas (BUENO, 2007).

No *software* foi criado este contraste conferindo três estilos: durante o desenrolar da história utilizou-se um tipo de letra que combinasse com o papiro inserido para contar a história; enquanto nas falas foi usada outra fonte com o objetivo de diferenciar, o narrador das personagens; nos livros e atividades, optou-se por um estilo formal, sem um contraste marcante entre a fonte, apenas conferindo certa diferença ao aplicar negrito nos títulos e redução da fonte em algumas legendas como as de fotos e diagramas.

Na figura 14 são exemplificados outros eventos utilizados na interação com usuário. O evento *onRollOver* é acionado assim que o cursor do mouse passar por cima do elemento que contém este evento. A ação de aumentar o elemento é disparada e quando o mouse sair de cima do elemento outro evento será acionado, o *onRollOut*, para que o elemento retorne ao seu tamanho.



Figura 14 – Eventos do mouse
Fonte: Guimarães (2011)

Na figura 15 visualiza-se o trecho de código com o manipulador de evento correspondente. A variável `_root.fruto`, corresponde ao caminho raiz do elemento gráfico em questão ao passo que, `_xscale` e `_yscale` correspondem respectivamente às coordenadas x e y do objeto. Elas recebem o valor em percentual em função do quanto se deseja que o objeto aumente ou diminua.

```
1 on (rollOver)
2 {
3     _root.fruto._xscale=150;
4     _root.fruto._yscale=150;
5 }
6 on (rollOut)
7 {
8     _root.fruto._xscale=100;
9     _root.fruto._yscale=100;
10 }
```

Figura 15 – Eventos do mouse
Fonte: Autoria Própria

3.2.1 Animações Gráficas

Uma parte importante do projeto, para garantir uma interface atrativa, foi incluir diversas animações gráficas. Como o *software* contava uma estória fictícia, em algumas cenas, personagens foram animados para atrair a atenção das crianças. Várias cenas usaram esse recurso, que consumiu grande parte do trabalho de implementação.

Para o desenvolvimento utilizou-se de programação usando a linguagem *SwishScript* e também recursos de animação prontos, dependendo do efeito que desejava obter na tela.

Com criatividade e à medida que conhece melhor as possibilidades da ferramenta Swish Max e seus recursos, unida a sua linguagem, acaba-se desenvolvendo cenas com animações envolventes e mais próximas da imaginação. Por exemplo: fantasma saindo de um túmulo no cemitério, morcego surgindo de um portal, dragões soltando fumaça pelas ventas, entre outras.

Durante o desenvolvimento da cena representada na figura 16, a catapulta deveria lançar raízes tuberosas para atingir as naves. Inicialmente pensou em parábola invertida para poder fazer o efeito da catapulta lançando a raiz, estudando as coordenadas x e y e a fórmula a ser aplicada, porém a própria ferramenta

proporcionou a solução através do evento do mouse associado ao efeito que o Swish Max disponibiliza através de sua biblioteca.



Figura 16 – Batalha raízes tuberosas
Fonte: Guimarães (2011)

Através de eventos do mouse é feito o controle de quando inicia o efeito que a raiz e catapulta terão. Aciona-se através de código no *script* as ações que acontecerão quando o objeto raiz for solto próximo a catapulta, iniciando seu “filme”, considera-se também uma ação apropriada caso o objeto for solto em outro local da cena, distante da catapulta, conforme figura 17.

```

1 onSelfEvent (press) {
2     x=178.05;
3     y=525.1;
4     x1 = this._x - x ;
5     y1 = this._y - y;
6     distance = Math.sqrt ((x1*x1)+(y1*y1));
7
8     // se objeto longe do alvo, usuario pode movimenta-lo; senao, objeto fica no lugar.
9     if (distance > 5) {
10        startDragUnlocked();
11    }
12}
13// evento soltar mouse
14onSelfEvent (release) {
15    // soltou o botao = pare de seguir o mouse.
16    stopDrag();
17    x1 = this._x - x;
18    y1 = this._y - y;
19    distance = Math.sqrt((x1*x1)+(y1*y1));
20
21    if ( distance < 150){
22        this._x = x;
23        this._y = y;
24        _root.beterraba_2.play();
25        _root.catapulta.play();
26    }
27    else{
28        this._x = 307;
29        this._y= 534.2;
30    }
31}
32onFrame (104) {
33    _root.espaconave_B.play();
34}

```

Figura 17 – Código raiz tuberosa
Fonte: Autoria Própria

Ao atingir o *frame* específico, inicia o efeito relacionado ao objeto nave. A colisão da raiz com a nave faz com que esta comece a pegar fogo e perder altura até cair resultando em uma explosão.

A solução proporcionada pela ferramenta ficou bem mais simples do que implementar o código através de equações com parábola.

Outro efeito pode ser verificado na figura 18, o vaivém das abelhas conferindo o ataque contra os personagens sombrios e a conseqüente queda deles.



Figura 18 – Sombrios sendo atacados
Fonte: Guimarães (2011)

3.3 PROJETO DE INTERAÇÃO

O projeto de interação trata do comportamento humano frente a utilização da tecnologia, facilitando a vida do usuário, para que ele não tenha que adivinhar o que deve fazer e como fazer. Segundo Preece, Rogers e Sharp (2005, p. 28), “especificamente, significa criar experiências que melhorem e estendam a maneira como as pessoas trabalham, se comunicam e interagem”.

O desenvolvimento do *software* mereceu uma atenção especial por estar voltado a um público específico - crianças – e, considerando-se que o aplicativo foi utilizado durante as aulas. Ao definir usuário e local, o projeto de *design* possibilita a especialização identificada através das necessidades do usuário (PREECE; ROGERS; SHARP, 2005).

Uma das escolhas feitas durante o desenvolvimento foi utilizar a técnica de testar e aprovar (PREECE; ROGERS; SHARP, 2005) o aplicativo com um grupo de crianças, para verificar se a experiência proporcionada atingia o objetivo esperado.

Após realizado teste com um grupo de crianças para verificar quais seriam suas dificuldades e reações frente ao *software*, foi constatado um problema inicial de navegabilidade. Em uma das atividades apresentadas na Figura 19 (cena de Atividade de Noções Gerais), o cursor representado por uma fada atrapalhava ficando sobreposto aos botões Ir e Voltar. Isso confundia as crianças ao tentarem pressionar esses botões. Curiosamente, o fato relatado foi observado por crianças que possuem contato com computador. As crianças que tinham pouca familiaridade não relataram esse desconforto.



Figura 19 – Primeira versão Atividade Noções Gerais de Angiospermas
Fonte: Guimarães (2011)

Através dessa identificação, decidiu-se fazer com que a fada não representasse o cursor em si. Em lugar disso, a imagem deveria seguir a posição do cursor, que passou a ser representado novamente pela flecha branca com que muitos já estão acostumados. Também seguindo o padrão que se vê em diversos aplicativos, quando o mouse fica em cima de um objeto pressionável, a imagem é transformada no ícone representado por uma mão, conforme mostra a Figura 20. Por fim, quando o mouse está sobre os botões Ir e Voltar a fada passou a ficar atrás do botão a ser acionado, facilitando assim a visualização.

Esta observação só foi relatada no pressionar dos botões Ir e Voltar. Nessa mesma tela há potes relacionados com a atividade que também devem receber cliques de mouse. Após análise, acredita-se que isto ocorreu, pelo fato de que, quando a fada fica sobreposta a um pote, o mesmo obtém um aumento de tamanho, através de um pequeno zoom, demonstrando ser um objeto pressionável enquanto que os botões Ir e Voltar representados pela flor, não tem este efeito, e a fada ficando na frente acabava gerando a dúvida se era mesmo um objeto pressionável.



Figura 20 – Atividade Noções Gerais de Angiospermas
Fonte: Guimarães (2011)

Durante a fase de desenvolvimento, uma das atividades criadas pela cliente (a aluna de mestrado) conforme Figura 21, consistia em ter um botão de Instruções. Ao ser pressionado esse botão, o *software* abriria uma caixa de texto, ou outra cena como mostra a Figura 22, orientando a criança na realização da atividade.

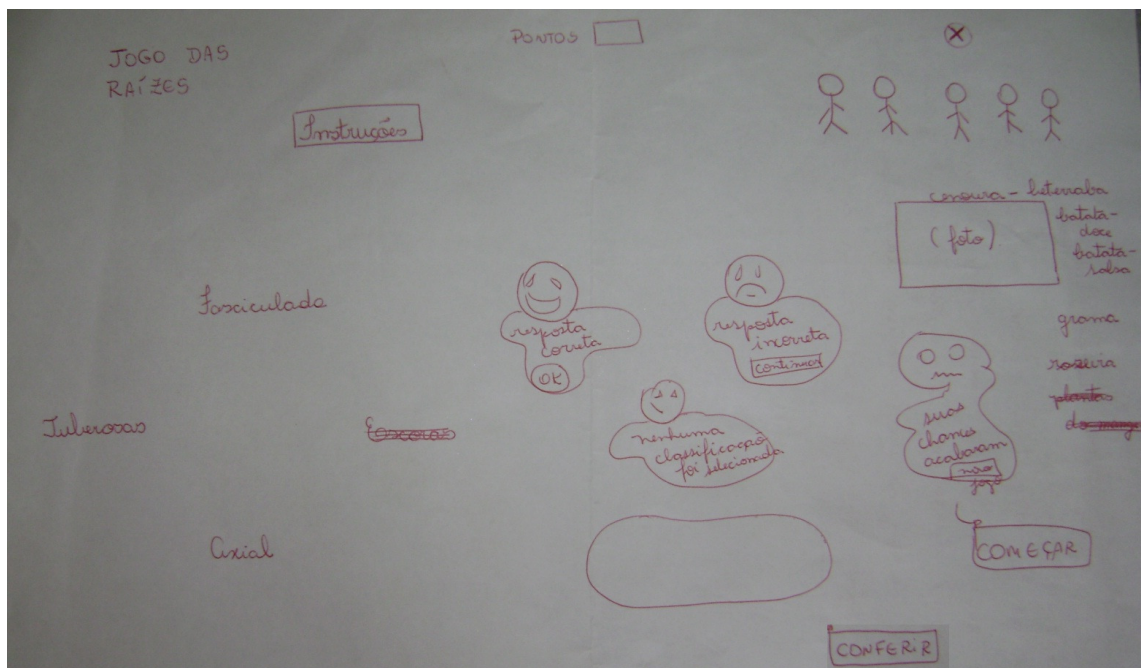


Figura 21 – Esboço da Atividade Raízes
Fonte: Zanotto

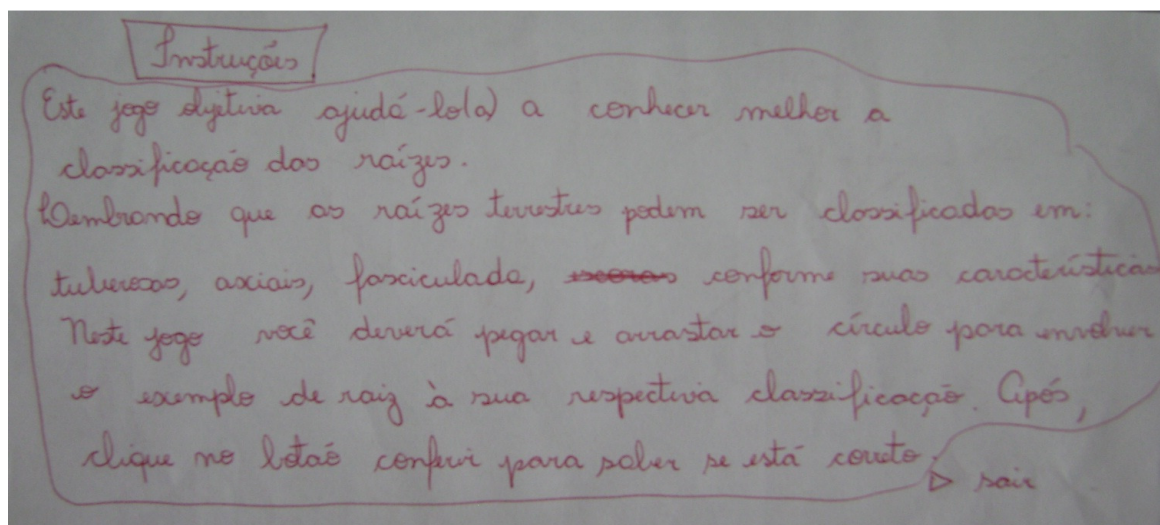


Figura 22 – Esboço Instruções
Fonte: Zanotto

Antes da implementação, foi realizado um protótipo, representado na Figura 23, para constatar se era viável o desenvolvimento e se ao fazer o teste a criança leria as instruções antes de começar a atividade conforme Figura 24. A atividade consistia de um banco de palavras espalhadas, a foto de uma raiz e uma forma elíptica. O que se esperava dessa atividade é que a foto da raiz fosse associada a uma das palavras, para isso a criança deveria pegar a forma elíptica e cercar a palavra que julgasse correta.



Figura 23 – Protótipo Atividade Raízes
Fonte: Autoria Própria

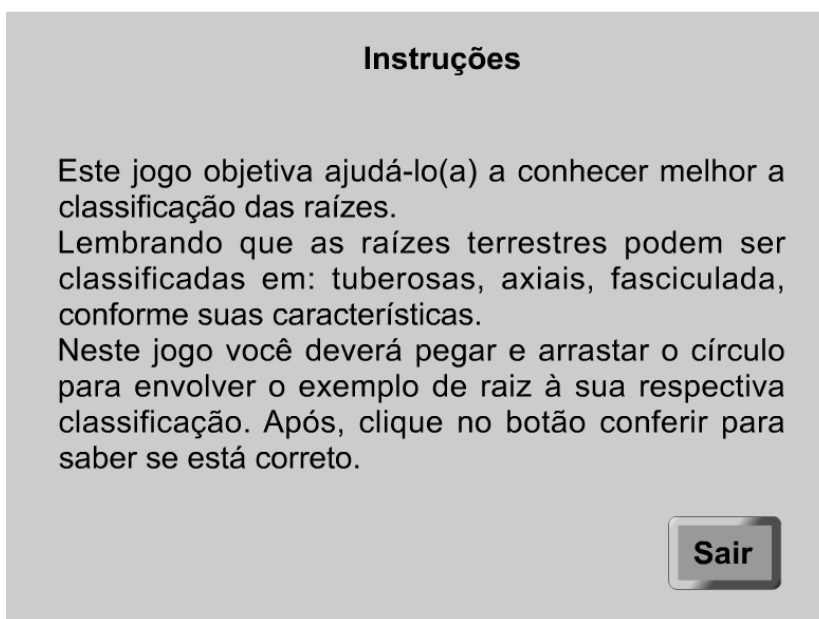


Figura 24 – Protótipo Cena Instruções
Fonte: Autoria Própria

Um aspecto perceptual conhecido é a de que são raros os usuários que lêem instruções, o mais comum de se observar é já iniciarem o aplicativo. Isto é ainda mais comum se tratando de crianças, que pela ânsia em jogar, não se preocupam em ler como realizar o objetivo. Por causa disso muitas vezes podem perder o interesse no *software* por não conseguir descobrir logo como funciona, avançando para a próxima cena ou podem até se sentirem frustrados e oprimidos

pela tecnologia por não conseguir atingir o objetivo rapidamente (NIELSEN; LORANGER, 2007). Ao ver a cena sem ler as instruções a primeira reação é a de pressionar a palavra que se relaciona com a foto, que são os primeiros objetos a chamarem atenção. Ao observar melhor a cena, percebe-se a forma elíptica e então descobre-se que há uma razão dela existir, a próxima provável tentativa é a de arrastar a palavra relacionada a foto para dentro dela. Estas foram as primeiras impressões da implementadora da atividade, porém como o público alvo são crianças o certo é fazer o teste com elas para verificar se esta impressão se confirma.

Ao realizar o teste, fica evidenciado que a atividade está confusa e por mais que melhore a disposição dos objetos na cena, ainda poderá gerar desconforto, principalmente quando se sabe que o aplicativo também será utilizado por crianças que possuem pouco contato com computador.

Conforme Nielsen e Loranger (2007), os usuários costumam olhar primeiro para o centro da tela para depois olharem para o lado esquerdo e então para a parte superior, e assim já vão separando o que julgam importante e relacionando os elementos encontrados, ficando a parte inferior ignorada por pressuporem ser de menor importância.

Outro ponto em que se teve cuidado no projeto foi com o uso de cores. Por exemplo, o programa apresenta balões coloridos com as respostas “certo” e “errado”. Esses balões tem cores vermelho e verde, mas para garantir que não houvesse dificuldades de uso com crianças daltônicas, as palavras foram escritas com letras claras, garantindo a facilidade de leitura. Posteriormente, por um critério didático solicitado pela cliente, as palavras dentro dos balões foram trocadas para “verdadeiro” e “falso”, pois as palavras usadas anteriormente poderiam desmotivar as crianças.

Mais um critério de acessibilidade é poder usar uma interface por meio do teclado, sem o mouse. Embora o *software* tenha sido desenvolvido pensando principalmente no mouse, a implementação usando o Swish Max automaticamente tratou também o teclado. Assim, usando a tecla *Tab*, um aluno pode navegar pelos elementos da interface e usar a tecla *Enter* ou a *barra de espaço* para acionar o objeto desejado.

3.4 DIFICULDADES DE IMPLEMENTAÇÃO

Apesar de *SwishScript* ser uma linguagem relativamente fácil não há muita documentação específica. Isso gerou algumas dificuldades.

A ferramenta Swish Max utiliza técnicas da POO, mas de uma maneira particular, diferente, por exemplo, de Java, em que a implementação de um programa começa com uma classe. Assim houve uma certa confusão com variáveis globais no início do desenvolvimento do *software*: acreditava-se que as variáveis criadas em cada cena seriam locais, ou seja, que ao partir para a próxima cena tais variáveis não mais existiriam. Porém, ao repetir uma variável se a mesma possuísse um valor armazenado este aparecia. Como é o caso de uma cena de atividade que continha a variável booleana erro: se o aluno pressionasse a resposta errada a variável armazenaria o valor falso, ao partir para outra atividade, a resposta errada já continha esse valor demonstrando na atividade a resposta errada.






Isto é evidenciado em algumas cenas, como exemplificado na atividade Caule da Figura 25. Quando o aluno erra a resposta, desaparecem as opções de continuar tentando, assim como os botões de “Ir” ou “Voltar” a uma cena anterior. Isto deixou como única opção o botão para ele ir até a Biblioteca para que reveja o conteúdo do livro “Caule”. Após o término do conteúdo do livro o usuário retorna à atividade “Caule” para nova tentativa conforme Figura 26.



Figura 25 – Atividade Caule opções desabilitadas
Fonte: Guimarães (2011)

Iniciar Atividades Biblioteca

Preciso resolver essa atividade para subir de cargo no exército. Ajude-me a relacionar corretamente o tipo de caule com a foto correspondente.

(a) Tubérculo	()	
(b) Estipe	()	
() Colmo	()	
(d) Tronco	(c)	
(e) Bulbo	()	

Voltar Ir

Figura 26 – Atividade Caule
Fonte: Guimarães (2011)

No desenvolvimento de cada cena o programa Swish Max tem como opção executar todas as cenas do projeto, ou apenas a cena que está desenvolvendo. Para agilizar o desenvolvimento, verificando como as animações e implementações ficavam, era executada apenas a cena que estava sendo implementada, até porque o projeto final conta com o total de 106 cenas. Para que não ficasse passando novamente por cenas já avaliadas, executava-se apenas a cena em que se estava trabalhando.

Após cerca de 20 cenas o projeto foi salvo em formato executável para realização de testes, constatou-se que muitas das cenas haviam perdido seus efeitos de animação. Teve que se fazer nova análise das cenas prontas, após comparar com as que haviam ficado com a animação, descobriu-se que os objetos que continham efeito direto sem estar com característica de filme acabavam voltando a ficar estáticas. Esta falha contribuiu para atrasos no desenvolvimento, pois toda cena teve que ser reavaliada e executada por completo para garantir sua perfeita execução.

Mesmo o desenvolvimento do *software* contando apenas com uma implementadora, optou-se pelo controle de versões, para cada grande alteração realizada no aplicativo o projeto era salvo com versão diferente. Isto possibilita o retorno a trechos de códigos e retorno a mudanças (KOSCIANSKI e SOARES, 2007) que não foram bem aceitas após testes com aluno, como o caso do cursor

representado pela fada da figura 19 e da atividade flor exemplificado na figura 27. Onde na primeira implementação a atividade consistia em arrastar os nomes de cada parte da flor para o local correspondente. Em outra versão seguindo os critérios estabelecidos pela cliente, a atividade consistia em duas partes, sendo a primeira montar a flor e a segunda identificar as partes que a compõem, conforme figura 28. Após análise, verificou-se que a atividade ficou confusa e resolveu retornar à aplicação antiga.



Figura 27 – Atividade Flor
Fonte: Guimarães (2011)



Figura 28 – Atividade Flor descartada
Fonte: Guimarães (2011)

4 CONCLUSÃO

O desenvolvimento do *software* acrescentou novos conhecimentos aos adquiridos no curso. No início, o desafio assustou um pouco, pela responsabilidade no desenvolvimento de algo que a cliente tinha em mente e julgava que tudo era passível de concretização, num espaço de tempo relativamente curto, quando na realidade não é bem assim que a programação funciona.

Saber a quem se destinava o *software* ajudou no desenvolvimento prevendo sua utilização a partir das limitações e interesses dos usuários. A realização de testes, ao longo de todas as fases, contribuiu para correção de eventuais falhas que só seriam identificadas no final do desenvolvimento. Isso possibilitou economia de tempo e recursos.

Num primeiro momento, o fato de se trabalhar com uma equipe multidisciplinar parece que não vai funcionar. Ao longo do processo, porém, percebe-se que, quando realizado em equipe, atende objetivos talvez não alcançáveis caso fosse desenvolvido por apenas uma das partes envolvidas. Exemplificando, a bióloga não compreendia técnicas de desenho nem de programação; o desenhista não entendia programação nem Biologia; a desenvolvedora não compreendia o ensino de Biologia nem técnicas de desenho.

Entretanto, explorando as potencialidades de cada um para o desenvolvimento do aplicativo, o resultado final foi surpreendente, gratificante. Um acabou aprendendo um pouco do que o outro sabia, ou pelo menos, sabendo opinar sobre o produto, existindo um intercâmbio muito grande de informações. Esse intercâmbio propiciou um resultado final satisfatório que foi atender uma determinada clientela que precisava aprender um conteúdo considerado essencialmente memorístico e de difícil compreensão.

Pode-se dizer que outro ponto favorável foi que, mesmo com papéis e responsabilidades diferenciadas houve um entrosamento harmonioso entre a equipe multidisciplinar, com muito respeito humano e do conhecimento do outro, conseguindo-se assim atender aos prazos pré-estabelecidos.

Concluiu-se, portanto que, um *software* educativo tem melhores resultados quando projetado e implementado por uma equipe multidisciplinar, propiciando qualidade tecnológica, técnica e pedagógica ao produto.

4.1 TRABALHOS FUTUROS

O *software* foi desenvolvido para ser utilizado como complemento do estudo em sala de aula, pensando nas limitações dos computadores da escola, como trabalho futuro poderia ser transformado em um site em que os alunos pudessem acessar via *web* também de suas casas, ou transformado em um cd não necessitando assim de conexão com a Internet.

Desenvolver *software* dos demais conteúdos que as crianças tem dificuldade na apreensão estimulando o aprendizado de forma mais prazerosa, através da interação com o aplicativo.

Adicionar sons para maior imersão no uso do *software*, complementando efeitos quando acertar e errar as atividades. Em cenas que aparecem bichos, colocar sons que estes animais produzem, ou mesmo nas falas dos personagens envolvendo ainda mais o aluno com o *software*, porém, não deixando de lado a parte textual ampliando a acessibilidade ao aplicativo.

REFERÊNCIAS

- ANTUNES, Diego Roberto; PEREIRA, Diogo Augusto Barros. **Usabilidade em sistemas estatísticos**: estudo e implementação de uma interface para usuários leigos. Ponta Grossa, PR. 2008. 69 f. Trabalho de Conclusão de Curso (Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas) – Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2008.
- BUENO, Francisco da Silveira. **Minidicionário da língua portuguesa**. 2. ed. São Paulo: FTD, 2007.
- CATAPAN, Araci Hack; FIALHO, Francisco. Antonio Pereira. Search for pedagogical indicators for evaluating educational material. **Human Factors And Ergonomics Society Annual Meeting Proceedings**, v.54, p.109-112, nov. 2009.
- DAMASCENO, Anielle. **Flash MX 2004**: design e animação para web e multimídia. Florianópolis: Visual Books, 2004.
- DEITEL, Harvey M.; DEITEL, Paul J. **Java Como Programar**. 6. ed. São Paulo: Pearson Prentice Hall, 2005.
- FALKEMBACH, Gilse A. Morgental. Concepção e desenvolvimento de material educativo digital. **Revista Novas Tecnologias na Educação**, Porto Alegre, v. 3 n. 1, p. 1-15, mai. 2005.
- GOUVEIA, Luis Borges. **Uma introdução ao multimédia**: conceitos básicos. 2004. Disponível em: <http://www2.ufp.pt/~lmbg/reserva/intro_mm.pdf>. Acesso em: 15 fev. 2011.
- GUIMARÃES, Welington. **Figuras**. São Mateus do Sul. 2011.
- KOSCIANSKI, André. **Game Programming with Irrlicht**. Seattle (USA): CreateSpace, 2011.
- KOSCIANSKI, André; SOARES, Michel dos Santos. **Qualidade de software**: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2. ed. São Paulo: Novatec, 2007.
- NIELSEN, Jacob; LORANGER, Hoa. **Usabilidade na Web**. Rio de Janeiro: Elsevier, 2007.
- PAULA FILHO, Wilson de Pádua. **Multimídia**: conceitos e aplicações. Rio de Janeiro: LTC, 2000.

PREECE, Jennifer; ROGERS, Yvonne; SHARP, Helen. **Design de interação: além da interação homem-computador**. Porto Alegre: Bookman, 2005.

ROMISZOWSKI, Hermelina Pastor. Avaliação no design e desenvolvimento de multimídia educativa: estratégia de apoio ou parte do processo? In: CONGRESSO INTERNACIONAL DE EDUCAÇÃO A DISTÂNCIA DA ASSOCIAÇÃO BRASILEIRA DE EDUCAÇÃO A DISTÂNCIA (ABED). 7., 2000. **Anais...** São Paulo, 2000.

SANTIAGO, Luiz Carlos. **A multimídia interativa como recurso didático-pedagógico para o ensino teórico-prático da semiologia em enfermagem**. 2003. 118 f. Tese (Doutorado em Enfermagem) - Universidade de São Paulo. Ribeirão Preto, 2003.

SANTOS, Mário. **Porque o AS3 pode ser entendido**. Disponível em: <<http://www.msdevstudio.com>>. Acesso em: 9 set. 2010.

SCAPIN, Dominique L.; BASTIEN, J. M. Christian. Ergonomic criteria for evaluating the ergonomic quality of interactive systems. **Behaviour & Information Technology**, v.6, n.4-5, p.220-231, jul. 1997.

SOMMERVILLE, Ian. **Engenharia de software**. 8. ed. São Paulo: Pearson Addison-Wesley, 2007.

SOUSA, Alexandre S. **Desenvolvendo jogos em ActionScript: estrutura básica da aplicação**. Disponível em: <<http://alexess.com/2011/03/28/desenvolvendo-jogos-em-actionscript-estrutura-basica-da-aplicacao>>. Acesso em: 19 out. 2011.

SOUZA, Clarice. S.; LEITE, Jair. C.; PRATES, Raquel. O.; BARBOSA, Simone. D. J. **Projeto de Interfaces de Usuário: perspectivas cognitivas e semióticas**. Disponível em: <http://www.dimap.ufrn.br/~jair/piu/JAI_Apostila.pdf>. Acesso em: 18 nov. 2010.

TAVARES, R. Animações interativas e mapas conceituais: uma proposta para facilitar a aprendizagem significativa em ciências. **Ciência & Cognição**, v.13, n.2, p.99-108, jul. 2008.

WILLIAMS, Robin. **Design para quem não é designer: noções básicas de planejamento visual**. 8. ed. São Paulo: Callis, 1995.

ZANOTTO, Denise do Carmo Farago; VIEIRA, Priscila Baer Gomes; FRANCISCO, Antônio Carlos de; KOSCIANSKI, André. Construção de softwares multimídia no ensino de ciências: uma contribuição para o aprendizado de angiospermas. In: Encontro Regional de Ensino de Biologia (EREBIO), 5., 2011. **Anais...** Londrina, 2011.