

Proposta de metodologia de aprendizado da programação de computadores através da recontextualização de jogos sérios no estilo *Game & Watch*

Rafael Faria de Azevedo

Bruno Campagnolo de Paula

Instituto de Tecnologia do Paraná, Centro de Engenharia de Sistemas Inteligentes, Brasil.

Pontifícia Universidade Católica do Paraná, Departamento de Informática, Brasil.

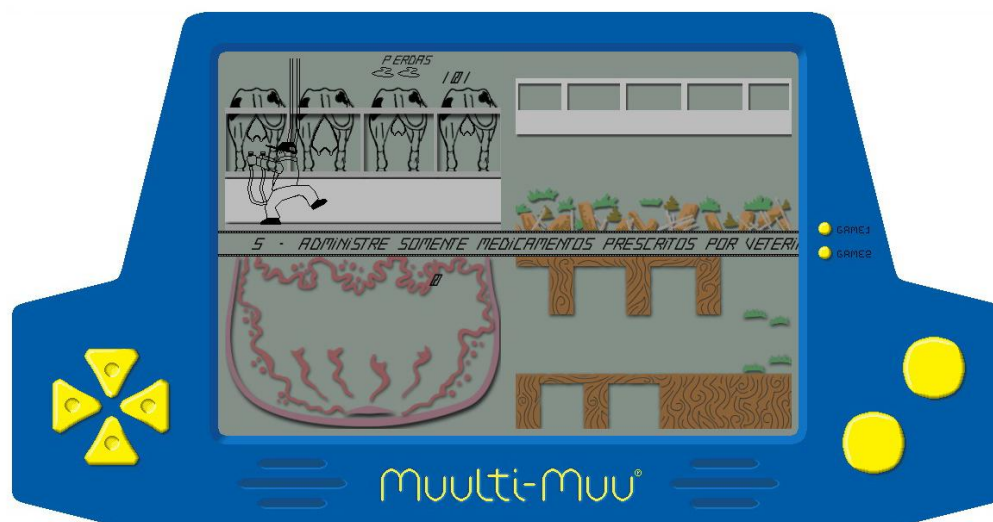


Figura 1: Ilustração de Muulti-Muu com o primeiro jogo carregado.

Abstract

Nowadays, the presence of the computer is substantial in almost every human environment. It became essential at work and highly appreciated for pleasure. Contradicting this popularity, the courses of graduation in computing – Computer Science, Computer Engineering, Information Systems, Digital Games, and others – have had a low number of freshmen yearly. As an aggravating of this situation, these courses have had high dropout rates due to, among other factors, the difficulty that students meet in subjects related to computer programming. As an alternative to decrease this difficulty, we propose the use of an engine called FlashPunk in order to recontextualize games. It is an open source library for the Flash Platform. In this proposal, its use aims the development of electronic games as a path of learning the basic concepts of programming. The article also presents a serious game that was developed on this platform. This game called Muulti-Muu teaches basics of the management of dairy cattle and was built through the recontextualization of the 80's games called Game & Watch. Finally, we also present the initial discussion of a methodology for teaching computer programming based on this process of recontextualization.

Keywords: computer programming, education, FlashPunk, serious games, Game & Watch.

Resumo

A presença do computador é massiva em praticamente todos os ambientes humanos atuais, pois se tornou essencial para o trabalho e muito apreciado no lazer. Contradizendo esta popularidade, os cursos de graduação em Informática – Ciência da Computação, Engenharia da Computação, Sistemas de Informação, Jogos Digitais, entre outros – tem tido anualmente um número baixo de calouros. Como agravante desta situação, estes cursos apresentam altas taxas de desistência devido, entre outros fatores, à dificuldade que os alunos encontram nas matérias relacionadas à programação de computadores. Como alternativa à diminuição desta dificuldade, propomos a aplicação de uma *engine* chamada FlashPunk para a recontextualização de jogos. Ela é uma biblioteca open source para a plataforma Flash. Nesta proposta, sua utilização objetiva o desenvolvimento de jogos eletrônicos como caminho rumo ao aprendizado dos conceitos básicos de programação. Neste artigo também apresentamos um jogo sério que foi desenvolvido nesta plataforma. Este jogo chamado Muulti-Muu ensina noções básicas do manejo do gado leiteiro e foi construído através da

recontextualização dos jogos dos anos 80 chamados Game & Watch. Por fim, apresentamos, também, a discussão inicial de uma metodologia para a aprendizagem da programação de computadores baseada neste processo de recontextualização.

Palavras chave: programação de computadores, educação, FlashPunk, jogos sérios, Game & Watch.

Contato dos autores:

strafa@hotmail.com
 brunodepaula@gmail.com
 {rfazevedo,campagnolo}@tecpar.br

1. Introdução

Durante a década de noventa houve um grande interesse nos cursos de graduação ligados à computação, isso devido à novidade tecnológica que o computador representou para a população da época. Todavia, desde então, a procura por estes cursos vem diminuindo ano após ano, contrariando o aumento de postos de trabalho e a necessidade de empreendimentos neste setor [Digiampietri e Kropiwiec 2008].

A dificuldade no aprendizado da programação de computadores é apontada, pelos calouros dos cursos de Computação, como um dos principais motivos de desistência ou reprovação destas matérias na primeira vez em que elas são cursadas. [Kinnunen e Malmi 2006].

Almejando minimizar esta dificuldade, propomos o desenvolvimento de jogos eletrônicos como alternativa extracurricular à aprendizagem da programação, visto que muitos estudantes gostam de tais *games*. Segundo [Gomes et al. 2008], a apresentação de conteúdos para o conhecimento das estruturas de programação através da criação de jogos pode facilitar a aquisição do arsenal teórico e prático necessário ao desenvolvimento básico de *software*.

Do ponto de vista da metodologia para o aprendizado da programação, nos inspiramos no trabalho de [Radtke e Binder 2010] que propõe como forma de ensino um método baseado em tutorias. No que tange ao desenvolvimento do jogo Muulti-Muu, até que ele começasse a tomar forma fizemos de forma metódica algumas cópias de jogos *Game & Watch* já existentes. Porém, nossa metodologia será mais explicada em uma seção posterior dedicada a este propósito.

Apresentados os motivos de nosso trabalho, podemos enfim justificar a escolha do FlashPunk [About 2011] como *engine* útil ao aprendizado individual de conceitos básicos na programação de computadores.

Embora lançado recentemente (2009) pelo desenvolvedor independente de jogos Chevy Ray Johnston, o FlashPunk oferece às pessoas que nunca

tiveram contato com a programação de computadores a oportunidade de criar jogos usando conceitos básicos de maneira simples e rápida. Ele é uma *engine* para o desenvolvimento de jogos em duas dimensões escrito na linguagem ActionScript 3 para a plataforma Flash [About 2011]. Entre suas características estão:

- código aberto;
- gratuito;
- suporta o paradigma da orientação a objetos;
- sua plataforma – o Flash – é muito utilizada na internet pela facilidade em colocar os jogos na web [Introducing the Adobe Flash platform 2011];
- O FlashPunk é fácil de ser utilizado, pois sua comunidade de desenvolvedores é ampla;
- Seu site é atualizado diariamente com novidades, perguntas e respostas de usuários dos mais diversos níveis de conhecimento [Learn Flash Punk 2011];
- Para desenvolver os jogos não é necessário usar a plataforma oficial paga de desenvolvimento da empresa Adobe, proprietária da plataforma Flash.

Uma vez elucidadas as principais características da ferramenta escopo deste artigo, comentaremos brevemente o conceito de jogo escolhido para ser criado utilizando-se o FlashPunk. Este conceito chama-se jogos sérios (do inglês *serious games*), termo utilizado pela primeira vez em 1970 no livro *Serious Games*, do autor estadunidense Clark Abt.

Inicialmente jogos sérios eram entendidos como a maneira de aprender ou ensinar um assunto sério de maneira lúdica. Como diz Clark C. Abt “*Serious Games unites the seriousness of thought and problems that require it with the experimental and emotional freedom of activity play*” [Abt 1987].

Atualmente um jogo sério é entendido como “softwares desenvolvidos com qualquer tecnologia de jogos digitais, para qualquer plataforma, de qualquer gênero. Os *serious games* têm um propósito educacional explícito e cuidadosamente elaborado, e sua função principal não é o entretenimento. Os *serious games* podem ser usados para a saúde, educação, treinamento profissional, ciência e pesquisa, produção, anúncios e para o ambiente de trabalho. Os temas trabalhados incluem interesses das seguintes áreas: empresas e negócios, propaganda e comunicações, indústria, educação, forças armadas, governo e organizações não-governamentais.” [Animaeco 2011]. Sua importância é crescente e promissora, como relata o jornal O Globo, “Mais de 60% das grandes empresas americanas já usaram algum tipo de simulação nos treinamentos” [O Globo 2011]. Tendo explicado brevemente o que é um jogo

sério, discutiremos sobre o motivo de sua escolha em nosso projeto.

Decidimos desenvolver um jogo sério ao invés de um jogo clássico ao aprendizado da programação como um Pacman ou Tetris pelos seguintes motivos:

- O desenvolvimento de um jogo sério pode ser mais estimulante para quem está aprendendo a programar, visto que o jogo terá uma aplicação lúdica e útil em um contexto real;
- Escolher um projeto nesta linha auxilia na quebra do tabu de que jogos são apenas para entretenimento;
- Para a criação deste tipo de *game*, é necessário que o programador tenha um contato interdisciplinar com profissionais de outras áreas do saber, enriquecendo sua visão de mundo e de um ambiente profissional real.

Certamente os jogos sérios não são a principal tendência no mundo dos jogos digitais, assim como nosso jogo também não é o primeiro desta tendência. Da mesma forma o FlashPunk não foi a primeira *engine* de jogos a ser criada. Por isso, na sequência apresentamos alguns trabalhos que antecederam, inspiraram ou se relacionam com nosso trabalho.

2. Trabalhos relacionados

Como aludido acima, o FlashPunk não foi a primeira *engine* de jogos a ser criada. Nem mesmo foi a primeira *engine* de jogos para Flash. Dentre outras opções de *engines* voltadas à aprendizagem da programação de computadores pesquisadas podemos citar a Chien2D para a linguagem C, que foi inicialmente criada para o uso pessoal de seu criador, mas que pelas características pedagógicas está sendo utilizada em aulas de programação básica em universidades [Radtke e Binder 2010]. Uma segunda *engine* que caminha neste sentido e que pesquisamos em nosso trabalho é a JGame [JGame - a Java/Flash Game Engine for 2D Games 2011]. Esta coleção de bibliotecas para jogos em Java difere das demais pelo fato de seu escopo ser a criação de jogos para tecnologia embarcada J2ME, todavia, também existe uma versão para ActionScript 3. Por fim é importante mencionarmos o Flixel, que é uma *engine* para a plataforma Flash que influenciou na criação do FlashPunk, sendo bastante semelhante ao segundo, diferindo apenas em alguns aspectos técnicos.

Nossa escolha pelo FlashPunk, dentre as várias opções ocorreu pela sua simplicidade de utilização comparado com as outras *engines*. Além disso, a forma que tutoriais e fóruns estão organizados em seu site facilitam o aprendizado de iniciantes da informática e mesmo para autodidatas.

Assim como o Flashpunk não representa uma novidade no que diz respeito à *engine* de jogos, o jogo sério que desenvolvemos – Muulti-Muu – se integra a muitos outros já desenvolvidos. Desta forma, apresentamos alguns exemplos variados.

America's Army [American's Army 2011] é um jogo desenvolvido pelo exercito americano para simulação de treinamento e combate. Outro exemplo é McDonald's Video Game [McDonald's Video Game 2011], um jogo desenvolvido na Itália como sátira com o intuito de conscientizar os consumidores da rede de *fast food* McDonald's sobre suas atitudes questionáveis. Com o objetivo de citar diferentes estilos citamos também o jogo Yourself!Fitness [Yourself!Fitness 2011], que é um jogo sério classificado como *exergaming*, ou seja, jogo de exercícios. Nele o jogador é guiado por uma instrutora de aeróbica virtual que orienta a pessoa a realizar atividade física benéfica à sua saúde.

Para não termos a impressão de que existem jogos sérios apenas fora do Brasil, citaremos algumas criações nacionais. Búzios – Ecos da Liberdade [Búzios: Ecos da Liberdade, 2011] é um jogo sério sobre a Revolta dos Alfaiates que aconteceu na Bahia no fim do século XVIII. Esse game foi desenvolvido na Universidade do Estado da Bahia. Seu objetivo é colocar o jogador em contato com o contexto histórico local da época com fins pedagógicos. Outro exemplo de jogo sério nacional é Atlas Test Brazil [Atlas Test Brazil 2011], que funciona dentro da rede social Facebook. Seu objetivo é testar o conhecimento do jogador sobre o mapa nacional fazendo com que este aponte com o mouse a posição do ponto geográfico pedido pelo jogo. Um último exemplo de jogo sério nacional é Tartarugas [Tartarugas 2011], um jogo de educação ambiental para crianças do ensino fundamental.

Após citarmos estes exemplos de jogos sérios é importante dizer que, como talvez o leitor tenha percebido, não existe uma fronteira clara entre jogos sérios e jogos educativos.

Na próxima seção apresentaremos mais detalhes sobre a metodologia utilizada em nosso trabalho.

3. Metodologia

Nesta jornada, como resumidamente mencionado anteriormente, sobre a programação do jogo, buscamos inspiração no trabalho de [Radtke e Binder 2010] que propõe como forma de ensino um método baseado em tutorias. Nesta metodologia, a aprendizagem do conteúdo de programação é realizada através da apresentação conceitual da matéria associada à realização de exemplos práticos pelo tutor junto ao aprendiz. Esta atividade acontece passo a passo, podendo ser tutorada pessoalmente, por tutorial escrito ou por meio eletrônico como

áudio ou vídeo. Um exemplo semelhante dessa metodologia mais voltada para a aprendizagem de ciências exatas ocorre no portal web americano chamado Khan Academy [Khanacademy 2011], no qual se aprende Matemática, Astronomia, Ciência da Computação, entre outros, através de tutorias de vídeo chamados *vodcasts*. No Khan Academy, o estudante aprende o conteúdo passo a passo ensinado por um tutor, podendo rever os vídeos várias vezes até aprender o assunto. Além disso, conta com exercícios para desenvolver e consolidar seu conhecimento sobre o assunto estudado, inclusive conta com uma área de bate-papo onde pode interagir com o tutor ou com outros participantes.

Este tipo de aprendizagem individual torna-se cada dia mais importante, promovido pela importância das pessoas envolvidas na área da informática buscar uma formação continuada, movidos pela evolução diária deste campo do conhecimento [Souza 1998]. Mais do que nunca, o conteúdo aprendido na graduação não é suficiente para toda a vida, exigindo que a pessoa busque novos e diferentes meios de se manter atualizada [Quintão et al. 2000].

Juntamente com o aprendizado da linguagem de programação ocorreu a construção do jogo sério onde se aplicaram os conceitos aprendidos na prática, entretanto para que ele nascesse e para treinar a programação, foi realizada a cópia exata do jogo Egg e do jogo Turtle Bridge, ambos Game & Watch da Nintendo. Somente quando as cópias destes estavam terminadas passamos para o passo da recontextualização dos mesmos, que se tornaram respectivamente os mini-games Ordenha Maluca e Defensor do Úbere. Estes são os nomes dos dois primeiros jogos que compõe o jogo maior chamado Muulti-Muu. Por esta metodologia se mostrar eficiente será mantida até o final do projeto.

Na sequência nos encaminharemos para tratar especificamente dos dois eixos do trabalho o FlashPunk e o Game & Watch, materializado no jogo sério Muulti-Muu. Todavia, faremos um relato sintetizado da plataforma Flash.

4. Apresentando o Flash

O software Flash pertence à companhia estadunidense Adobe. Ele é um conjunto de ferramentas que permite a criação de animações, vídeos, apresentações, jogos, banners, *websites* e outras formas de interação multimídia. Ele funciona em diferentes sistemas operacionais, sejam eles para computadores ou para tecnologia embarcada como celulares, *tablets* e outros [Plataforma Adobe Flash 2011].

O motivo de termos escolhido esta plataforma, mesmo considerando certa relutância a ela no que

tange ao aprendizado da programação de computadores no meio acadêmico, foi a facilidade da programação de jogos 2D utilizando sua *engine* FlashPunk, aliado à vantagem do Flash possuir código aberto e permitir que os jogos feitos com ele possam ser colocados facilmente na internet, pois rodam em praticamente todos os navegadores. Também é importante salientar que o desenvolvimento de jogos independentes (*indie games*) utilizando esta plataforma, é um movimento mundial importante e crescente. Segundo a própria Adobe, noventa e oito por cento dos usuários de *desktop* conectados à internet são atingidos pela plataforma Flash [Introducing the Adobe Flash Platform 2011].

Após uma breve explicação sobre a plataforma do FlashPunk feita nas seções anteriores, vamos a ele de fato.

5. Apresentando o FlashPunk

Genericamente podemos dizer que para programar jogos existem dois caminhos. Programar partindo do zero ou usar uma biblioteca/*engine* pronta para isso. Uma biblioteca do ponto de vista da programação é constituída por vários comandos prontos que foram escritos por um programador e que outro pode utilizar para tornar mais fácil e rápido o desenvolvimento de um jogo. Seguindo a mesma lógica, uma *engine* é uma coleção de bibliotecas para a programação de jogos. Assim, o FlashPunk é uma *engine* para a criação de jogos em Flash através da linguagem ActionScript 3 [About 2011]. Justamente por possuir muitos comandos prontos é que o uso do FlashPunk facilita o aprendizado da programação para pessoas que tiveram pouco ou nenhum contato com este tipo de atividade. Neste capítulo apresentamos algumas características do *FlashPunk*.

Em linhas gerais, uma das características mais interessantes do FlashPunk é a maneira como é possível e recomendável organizar o jogo. Nele tudo é inicializado em uma classe Main, que é responsável por configurar informações básicas como o tamanho da tela e o modo de depuração do jogo. A classe Main estende a classe interna do FlashPunk chamada *Engine*. Dentro da classe Main deve ser criado um objeto da classe Mundo (*world*), para que este mundo exista no jogo.

Seguindo o raciocínio lógico, devemos criar também uma classe Mundo (*world*) para nela colocarmos todos os elementos do jogo ou de uma fase específica, pois um mundo pode ser apenas uma fase do jogo ou o jogo inteiro. Observe que a classe Mundo estende a classe interna do FlashPunk chamada *World*.

Para que os elementos do jogo existam, devemos criar uma classe que estende a classe interna do

FlashPunk chamada *Entity*. Sendo assim, cada elemento do jogo nesta *engine* é uma extensão da classe *Entity*. Um elemento do jogo pode ser o *background*, os inimigos, o jogador, uma pedra, etc. Na prática, o FlashPunk obriga o programador a desenvolver o hábito de criar cada um dos elementos de seu jogo de maneira independente, para depois relacioná-los conforme sua necessidade, o que evidencia boas práticas de programação dentro do paradigma da orientação à objetos.

Uma peculiaridade do FlashPunk é sua forma de carregar imagens, seja uma imagem simples ou um vetor de imagens (*sprite*). O que queremos dizer é que para carregar uma imagem, é necessário criar uma classe para carregar a imagem e criar um objeto para tocar esta imagem. A mesma lógica funciona para a utilização de som. Entretanto, é possível carregar uma imagem e um som numa mesma classe, porém, não é possível carregar duas imagens ou dois sons numa mesma classe para serem executados ao mesmo tempo. Outras características que tornam o FlashPunk interessante, prático e simples são seus métodos para colisão, física, emissão de particular, desenho, etc.

Neste ponto acreditamos ter explicado a *engine* de uma maneira um tanto técnica. Todavia, um jogo não se resume apenas à sua programação. Como referido inicialmente, quando planejamos a construção de Muulti-Muu, utilizamos o estilo Game & Watch como referencia e inspiração da criação artística. Pois é chegada a hora de comentarmos melhor sobre este estilo.

6. Game & Watch

Criar um jogo envolve muitos aspectos relacionados a diversas áreas do conhecimento. Após definirmos os caminhos da programação do jogo, o passo seguinte foi escolher o estilo gráfico e de jogabilidade do mesmo. Após pesquisar estilos que fossem adequados aos objetivos do projeto escolhemos os jogos Game & Watch para tematizar nossa criação. No próximo parágrafo justificaremos esta escolha.

Game & Watch foi uma série de pequenos equipamentos portáteis que possuíam um único jogo desenvolvido pela empresa japonesa Nintendo, que se tornaram famosos a partir da década de 80. O primeiro jogo produzido foi Ball em 1980 e o último foi Mario the Juggler em 1991. Alguns dos jogos mais famosos no mundo dos games eletrônicos surgiram neste estilo. Entre eles podemos citar: Donkey Kong, Super Mario Bros., Mickey Mouse, Snoopy Tennis, Zelda e outros. Ao todo, foram lançados 60 títulos [80-89 Eight-Eightynine 2011]. No Brasil eles ficaram popularmente conhecidos como mini-games.



Figura 2: Game & Watch Egg da Nintendo.

Os jogos Game & Watch foram escolhidos em nosso projeto pela facilidade em programá-los e por permitirem que criássemos novos jogos recontextualizando um *Game & Watch* já existente. Ressaltamos que esta recontextualização é vista como uma parametrização, da forma que é apresentada por Teixeira et al [2011], à partir de jogos Game & Watch (em nosso caso). Sobre a relação entre a contextualização e o aprendizado da programação de computadores, enfatizamos sua importância pela facilidade no entendimento do funcionamento dos jogos originais em vista do desenvolvimento de jogos parametrizados (recontextualizados) conforme o propósito de nosso projeto. Desta forma eliminamos a preocupação com game design, com balanceamento (que é a atividade de tornar o jogo desafiador e justo, para que não seja fácil ou difícil em demasia) e tivemos a possibilidade de jogar e perceber o funcionamento do jogo em implementação. Todo este processo permite aos bolsistas envolvidos no projeto ter um jogo completo em seu portfólio, atividade importante na carreira dos profissionais da área de Tecnologia da Informação. Acrescido a tais detalhes incluímos a facilidade da criação da arte sentida por parte da equipe que esteve responsável por esta atividade.

Acreditando ter apresentados os dois eixos fundamentais da criação de nosso jogo sério, a programação e a arte, descreveremos o processo de criação de Muulti-Muu para, finalmente podemos falar do jogo em si.

7. Processo de desenvolvimento

De uma maneira resumida comentamos o processo de desenvolvimento de Muulti-Muu na seção que trata da metodologia do projeto. Todavia, neste ponto faremos uma explanação mais detalhada deste tópico. Vale insistir que o aprendizado da programação utilizando o FlashPunk e o desenvolvimento de jogos no estilo Game & Watch aconteceu ao mesmo tempo.

Inicialmente, para que os conceitos de programação fossem compreendidos e praticados gradativamente nossa tarefa foi escolher jogos Game & Watch que fossem condizentes com o nível de conhecimento do aprendiz de programação. Pois,

mesmo sendo semelhantes, existe certa variação na dificuldade de desenvolvimento deste estilo de jogo.

Após jogar vários destes games, decidimos que os jogos Game & Watch que inspirariam os mini-games dentro de Muulti-Muu seriam: Egg, Turtle Bridge, Escape e Snoopy. Efetivamente para entender o funcionamento do estilo foram feitas cópias para fins didáticos dos jogos Egg e Turtle Bridge. Terminada esta fase, partimos para a recontextualização dos jogos para que eles se tornassem o primeiro e o segundo jogos de Muulti-Muu sendo eles, Ordenha Maluca e Defensor do Úbere (tabela 1). Após o termino destes dois jogos, trabalhamos na integração destes na interface de Muulti-Muu, para somente então continuar desenvolvendo os outros jogos.

O terceiro jogo, que atualmente encontra-se em desenvolvimento, está sendo recontextualizado com base no Game & Watch chamado Escape. Este se chamará Escape da Mosca do Chifre – imagem e explicação na tabela 1. O último jogo a ser implementado se chamará Mate a praga, uma recontextualização do jogo Snoopy da Nintendo, outro Game & Watch (tabela 1). Todo o processo de desenvolvimento do jogo está representado na figura 3 no final deste trabalho.

Neste ponto acreditamos ter mostrado os passos necessários para a compreensão do projeto. Sendo assim, após a tabela demonstrativa do projeto, apresentaremos o fruto palpável desta atividade, nosso jogo.


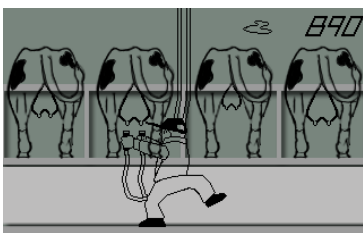

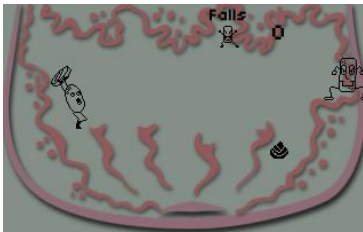
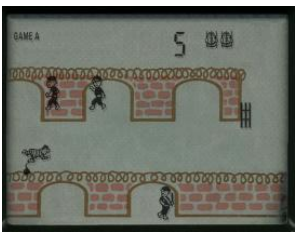


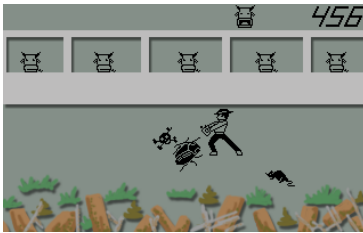
| Jogo original | Jogo contextualizado | Objetivo original | Objetivo contextualizado |
|---|---|---|--|
|  |  | Em Egg, o jogador tem que pegar os ovos postos pelas galinhas, sem deixar que caiam no chão. | Em Ordenha Maluca o produtor tem que tirar o leite das vacas no tempo certo sem derramá-lo, seguindo assim as boas práticas de manejo do gado leiteiro. |
|  |  | O jogador de Turtle Bridge deve passar a bagagem de um lado para outro pisando nas tartarugas sem cair na água. | O jogador de Defensor do Úbere deve controlar o anticorpo, ajudando-o a carregar o medicamento sem pisar na sujeira, para que acerte o microorganismo que causa mastite. |
|  |  | O objetivo deste Game & Watch chamado Escape é que o jogador consiga escapar da prisão sem ser percebido pelos guardas. | Neste jogo sério chamado Escape da Mosca do Chifre o jogador tem que guiar a vaca para fora do curral sem que as moscas do chifre a percebam. |
|  |  | No Game & Watch Snoopy, o jogador tem que acertar as notas musicais para que elas não acordem os passarinhos. | No jogo Mate a Praga, o produtor rural tem que matar as pragas (ratos e baratas) para que não importunem e contaminem as vacas do curral. |

Tabela 1: Tabela comparativa dos jogos utilizados no jogo.

8. Muulti-Muu: o jogo

Todas as informações teóricas e técnicas citadas acima foram e estão sendo utilizadas na criação de um jogo sério em estilo Game & Watch composto por quatro mini-games. Este jogo chama-se Muulti-Muu. Seu público alvo planejado é geral. Mesmo que seu conteúdo seja mais adequado aos filhos de agricultores, escolas rurais e outros. O jogo está disponível no site [Galaxis Farm 2010]. Ele é o fruto do trabalho integrado entre bolsistas e pesquisadores das áreas de sistemas inteligentes e de veterinária do Instituto de Tecnologia do Paraná – TECPAR, patrocinados pela Fundação Araucária usando recursos do CNPq.

Neste projeto está sendo integrada a pesquisa científica na área de informática, assim como o estímulo ao conhecimento de práticas simples e fundamentais no manejo do gado leiteiro de forma lúdica.

Para veterinários e estudantes do bem estar animal o que é tratado em Muulti-Muu é corriqueiro. Todavia, este mesmo conhecimento muitas vezes não é claro e utilizado nas propriedades de produção leiteira. Por isso é importante apresentar as noções mais elementares de higiene animal às crianças e aos produtores rurais.

Do ponto de vista prático, o aprendizado destas noções se dá através da jogabilidade dos mini-games e também através de frases baseadas na Instrução Normativa 51 do Ministério da Agricultura [Instrução Normativa 51 2011] que continuamente passam dentro de um letreiro que divide a tela dos jogos.

Estas frases apresentam alternância de velocidade, pois quando o jogador está jogando elas passam mais rapidamente na tela. Porém, quando há a transição entre um mini-game e outro, que ocorre pela perda do jogo ou quando se atinge cem pontos, o jogador é estimulado a ler uma frase inteira que passa numa velocidade menor. Vale lembrar que estas frases, num total de quatorze, são sorteadas aleatoriamente pelo próprio jogo, fazendo com que o jogador leia todas depois de um certo tempo de jogo.

De um ângulo metodológico, o jogo apresenta características de aprendizado tangencial [Video Games and Learning 2011; Aprendizado Tangencial como Solução para Jogos Educacionais 2011], pois o jogador pode obter ou aprimorar seu conhecimento sobre a produção leiteira de maneira não explícita. Quanto a isso nos referimos às ações tomadas no jogo como evitar o derramamento de leite e a existência de insetos que infectam o ambiente de ordenha.

Do ponto de vista de projeto, o jogo trata de boas práticas do manejo do gado leiteiro como: ordenhar as vacas sempre no mesmo horário, manutenção da limpeza do curral, manter as vacas em tratamento em bom estado de higiene.

Sobre a programação do jogo, vale lembrar que seu desenvolvimento não é apenas um exercício pedagógico, pois, mesmo estando sendo desenvolvido por um estudante da graduação, é um projeto completo que envolve profissionais de informática, design e medicina veterinária. Abaixo é apresentado um fluxograma que sumariza a criação de Muulti-Muu.

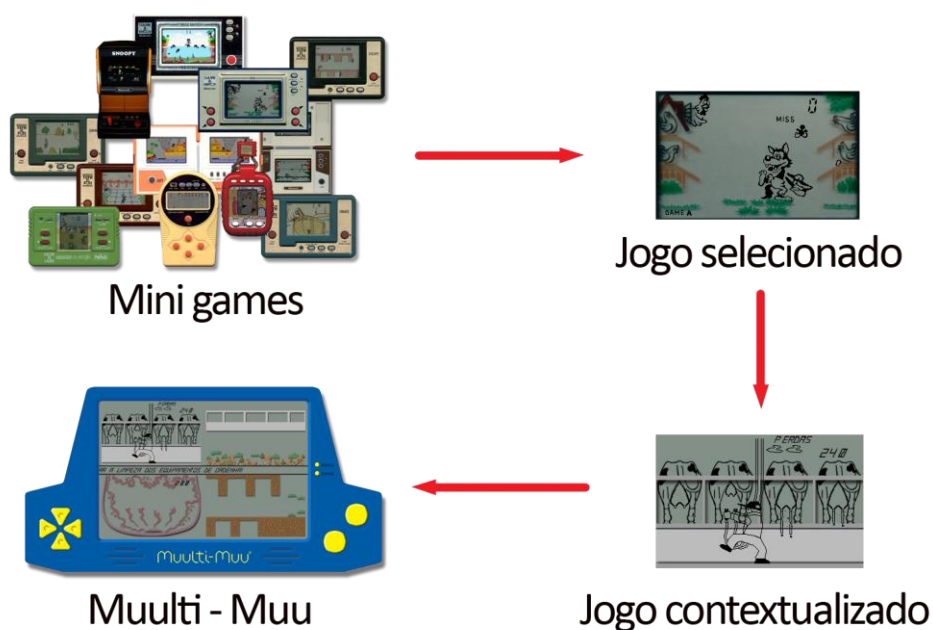


Figura 3: Imagem representando o processo de criação do jogo Muulti-Muu.

9. Conclusão

Com o surgimento de novas tecnologias o conhecimento tem descoberto novas e inusitadas formas de se propagar. Neste caso em especial, constatamos um benefício duplo do conhecimento. Do ponto de vista do desenvolvimento de software, temos experimentado os benefícios da utilização do FlashPunk e dos jogos Game & Watch no aprendizado dos conceitos elementares da programação de *software*. Por outro lado, em um viés rural, como fruto da utilização da informática de maneira lúdica, as novas tecnologias oferecem aos novos ou experientes produtores de leite a oportunidade de lembrar ou aprender conceitos fundamentais à sua atividade profissional em forma de um jogo sério em formato eletrônico.

Este trabalho relacionou áreas vistas comumente como antagônicas ou desconexas, provando que de agora em diante, no que diz respeito à informática, praticamente nenhum campo de atuação humano estará livre de suas influências e soluções. Por isso, outros trabalhos como este são incentivados.

Para isso apontamos algumas possibilidades de trabalhos futuros. Conforme salientamos acima, praticamente não existem barreiras para o uso da informática e mesmo dos games como ferramenta auxiliar ao aprendizado de qualquer campo do conhecimento. Por isso, propomos a criação de novos mini-games, e mesmo games, utilizando a engine usada neste projeto, assim como o estilo de jogos aproveitados neste trabalho. Devido ao amplo aumento da utilização da tecnologia embarcada (celulares, tablets, entre outros) sugerimos e planejamos a implementação destes e de outros jogos para estas plataformas, à fim de atingir um número maior de usuários dos mesmos.

Um ponto ainda a ser explorado em nosso trabalho é o teste de Muulti-Muu pelo público aberto, visto que ele será disponibilizado abertamente na internet buscando proporcionar um aprendizado de maneira tangencial, corroborando assim, com as novas formas de aprendizagem.

Agradecimentos

Chegado o fim deste trabalho queremos agradecer as pessoas e instituições que possibilitaram a realização desta bolsa de iniciação científica. Iniciamos agradecendo ao Eng. Dr. Milton Pires Ramos, como responsável pelo presente projeto. Agradecemos também aos nossos companheiros de projeto Diego Szpikula Cabral e Ângelo Rossi Hidalgo Junior pela criação da arte do jogo e pelo companheirismo. Pelo suporte técnico na área da veterinária agradecemos à Dra. Roberta Zuge e a Dra. Carmen Neusa Martins Cortada, assim como a João Paulo Gomes de Carvalho.

Por fim, agradecemos especialmente ao Instituto de Tecnologia do Paraná – TECPAR, que patrocinado pela Fundação Araucária usando recursos do CNPq viabilizou e deu suporte à realização deste projeto.

Referências

- ABOUT. Disponível em: <http://flashpunk.net/about/> [Acesso em 29 Jun. 2011].
- ABT, Clark C., 1987 *Serious Games*. London: University Press of America, pg. 11.
- AMERICAN'S ARMY. Disponível em: <http://www.americasarmy.com/downloads/?view=deploy> [Acesso em 04 Ago. 2011].
- ANIMAECO. Disponível em: <http://www.latec.ufrj.br/portalanimae-co/index.php/games/117-serious-games.html> [Acesso em 21 Jul. 2011].
- APRENDIZADO TANGENCIAL COMO SOLUÇÃO PARA JOGOS EDUCACIONAIS. Disponível em: <http://www.nonuba.com.br/eventos/1132-aprendizado-tangencial-como-solucao-para-jogos-educacionais> [Acesso em 04 Ago. 2011].
- ATLAS TEST BRAZIL. Disponível em: <http://apps.facebook.com/atlastestbrasil/> [Acesso em 04 Ago. 2011].
- BÚZIOS: ECOS DA LIBERDADE. Disponível em: <http://www.comunidadesvirtuais.pro.br/buzios/downloads/> [Acesso em: 07/08/2011].
- DIGIAMPIETRI, L. A. e KROPIWIEC, D. D., 2008. *Desenvolvimento de Jogos para o Aperfeiçoamento na Aprendizagem de Disciplinas de Ciência da Computação*. In: SBC – Proceedings of SBGames'08.
- GALAXIS FARM. Disponível em: <http://galaxisfarm.com/> [Acesso em 05 Ago. 2011].
- GOMES, A., HENRIQUES, J. e MENDES, A. J. 2008. *Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores*. In: Educação, Formação e & Tecnologias, vol 1(1), Coimbra, Portugal, 93-103.
- INSTRUÇÃO NORMATIVA 51. Disponível em: portal.mda.gov.br/o/776834 [Acesso em: 05 Ago. 2011]
- INTRODUCING THE ADOBE FLASH PLATAFORM. Disponível em: http://www.adobe.com/devnet/flashplatform/articles/flashplatform_overview.html [Acesso em 15 Jul. 2011]

- JGAME - A JAVA/FLASH GAME ENGINE FOR 2D GAMES. Disponível em: <http://www.13thmonkey.org/~boris/jgame/> [Acesso em 30 Jul. 2011].
- KHANACADEMY. Disponível em: <http://www.khanacademy.org> [Acesso em 05 Jul. 2011].
- KINNUNEN, P., e MALMI, L. 2006. *Why students drop out cs1 course?* In: ICER'06: Proceedings of the second international workshop on Computing education research, ACM, New Yourk, NY, USA, 97-108
- LEARN FLASHPUNK. Disponível em: <http://flashpunk.net/learn/> [Acesso em 07 Jul. 2011].
- MAIN PAGE. Disponível em: http://www.flashdevelop.org/wikidocs/index.php?title=Main_Page [Acesso em 28 Jun. 2011].
- MCDONALD'S VIDEO GAME. Disponível em: <http://www.mcvideogame.com/index.html> [Acesso em 04 Ago. 2011].
- O GLOBO. Disponível em: <http://oglobo.globo.com/tecnologia/mat/2009/06/01/os-chamados-jogos-serios-sao-uteis-em-areas-como-educacao-saude-treinamento-no-trabalho-756128625.asp> [Acesso em 21 Jul. 2011].
- PLATAFORMA ADOBE FLASH. Disponível em: <http://www.adobe.com/br/flashplatform/> [Acesso em 04 Ago. 2011].
- POWERED BY FLASHPUNK. Disponível em: <http://flashpunk.net/> [Acesso em 07 Jul. 2011].
- QUINTÃO, P. L., SEGRE, L. M. e RAPKIEWICZ, C., 2001. *Atualização de Profissionais de Tecnologias de Informação: Educação Continuada e Novos Métodos.* In: Anais do XXI Congresso da Sociedade Brasileira de Computação, Fortaleza.
- RADTKE, P. V. W. e BINDER, F. V., 2010. *Chien 2D: A Multiplataform Library to Teach the C Language Through Games Programming.* In: SBC – Proceedings of SBGames'10.
- SOUZA, D. M. T., 1998. *A formação e a prática do profissional de programação incluindo a discussão de estratégias para a aprendizagem continuada.* In: Dissertação de mestrado (Universidade Federal da Paraíba), 87
- TARTARUGAS. Disponível em: <http://www.elfland.com.br/jogos/tartarugas/index.html> [Acesso em 04 Ago. 2011].
- TEIXEIRA, J. S. F., SÁ, E. de J. V. e FERNANDES, C. T., 2011. *Jogo Parametrizável como Recurso Didático-Pedagógico: Que Rumos Seguir?* In: VII Seminário
- Jogos Eletrônicos, Educação e Comunicação, 28-29 Julho 2011 Fortaleza. Ceará – Brasil.
- VIDEO GAMES AND LEARNING. Disponível em: http://www.youtube.com/watch?v=rN0qRKjfx3s&feature=player_embedded [Acesso em 04 Ago. 2011].
- 80-89 EIGHT-EIGHTYNINE. Disponível em: <http://www.eightyeightynine.com/games/game-and-watch.html> [Acesso em 27 Jun. 2011].