

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

NAIRA LUCIA TORRES PEREIRA


**Desenvolvendo Atividade de ensino do Pensamento
Computacional para o GCompris**

GARANHUNS

2019

Naira Lucia Torres Pereira

Desenvolvendo Atividade de ensino do Pensamento Computacional para o GCompris

	EMPRESA:
	ÁREA DE ATUAÇÃO: Universidade Federal Rural de Pernambuco
	PERÍODO DE REALIZAÇÃO: 12 de agosto à 29 de novembro de 2019
	CARGA HORÁRIA SEMANAL: 300H
	REMUNERAÇÃO/AJUDA: sem remuneração
	SUPERVISOR/E-MAIL: lmts@ufrpe.br

Relatório de Estágio apresentado ao Curso de Ciência da Computação, como requisito necessário para avaliação final na disciplina de Estágio Supervisionado Obrigatório.

Orientador:

Dr^o. Mariel José Pimentel de Andrade

Garanhuns, 20 de dezembro de 2019

Relatório de Estágio apresentado ao Curso de Ciência da Computação, como requisito necessário para avaliação final na disciplina de Estágio Supervisionado Obrigatório, aprovada pela comissão examinadora que abaixo assina.

Dr^o. Mariel José Pimentel de Andrade - Orientador
Unidade Acadêmica de Garanhuns
UFRPE

Dr^o. Rodrigo Gusmão de Carvalho Rocha - Examinador
Unidade Acadêmica de Garanhuns
UFRPE

Examinador - Examinador

Examinador - Examinador

Resumo

Com o crescimento do desenvolvimento de sistemas de software focados na educação surge a demanda de compreender o universo dos usuários, suas visões expectativas, e em especial, as crianças; e de contribuir significativamente para a evolução de softwares que proporcionam esta experiência aos usuários. Neste relatório estão apresentadas as atividades desenvolvidas no Estágio Supervisionado Obrigatório, que teve como principal elemento contribuir e explorar o software GCompris, resultando em um roteiro de customização do ambiente de desenvolvimento do GCompris; na implementação de nova atividade de Pensamento Computacional; e na customização do design de uma atividade de pensamento computacional já existente no GCompris a partir da perspectiva dos usuários finais.

Palavras-chave: Pensamento Computacional. GCompris. Jogo Educativo

Abstract

With the growth of the development of software systems focused on education comes the demand to understand the users' universe, their expectations visions, and especially the children; and contribute significantly to the evolution of software that provides this experience to users. This report presents the activities carried out in the Mandatory Supervised Internship, whose main element was to contribute and explore the GCompris software, resulting in a roadmap for customizing the GCompris development environment; in the implementation of a new Computational Thinking activity; and in customizing the design of an existing GCompris computational thinking activity from the end-user perspective.

Keywords: Computational Thinking, GCompris, Educational Game.

Sumário

Lista de Figuras	iv
Lista de Tabelas	v
1 Introdução	1
1.1 Objetivo	1
1.2 Justificativa	2
1.3 Organização do relatório	2
2 Local e Período do Estágio	3
3 Fundamentação teórica	4
3.1 GCompris	4
3.2 Pensamento Computacional	5
3.3 Desenvolvimento de artefatos digitais para crianças	6
3.4 Reúso e customização	7
4 Atividades Desenvolvidas	9
4.1 Roteiro para processo de customização de ambiente de desenvolvimento do GCompris	9
4.2 Desenvolvimento de nova atividade de pensamento computacional	10
4.3 Validação do artefato	12
5 Conclusões	15
A Customização de Ambiente de Desenvolvimento do GCompris	16
Bibliografia	20

Lista de Figuras

4.1	Atividade Programming Loop	11
4.2	Estrutura da atividade Programming Loop	11
4.3	Estrutura da Atividade Programming Maze - Versão Original	13
4.4	Estrutura da Atividade Programming Maze - Versão criada neste trabalho	14
4.5	Estrutura da Atividade Programming Loop - Versão criada neste trabalho	14

Lista de Tabelas

3.1	Categorias das atividades do GCompris	4
-----	---	---

Lista de códigos

A.1	Código do padrão GCText	17
A.2	Código para String	17
A.3	Código para uso de anchors	18

Capítulo 1

Introdução

O incentivo ao uso de softwares educacionais cresce a cada dia na intenção de proporcionar aos estudantes qualidade de ensino e facilidade no aprendizado, além de contribuir para que os mesmos construam novos conhecimentos. O GCompris é um exemplo notável de software livre que possui atividades de cunho educacional com a finalidade de contribuir para o processo de aprendizagem tornando-o mais lúdico.

O GCompris não está vinculado a orientações curriculares de um país específico, apesar de está disponível em mais de 50 idiomas e ter como principal público crianças na faixa de 2 a 10 anos, fica a critério do professor utilizar as atividades de acordo com os objetivos educacionais estabelecidos pelos mesmos [1]. Além de possuir atividades relacionadas ao currículo escolar tais como ciências, matemática, letramento, há também atividades que propõem auxiliar o aluno na literacia digital estimulando o aprendizado sobre o uso de periféricos como o mouse e teclado.

No entanto, com a integração dos conceitos sobre pensamento computacional ao currículo do ensino básico ganhando espaço, faz-se necessário a atualização de atividades deste cunho no software, daí parte a necessidade de envolver crianças na criação de artefatos digitais para uso por este público, e a colaboração de desenvolvedores em softwares deste cunho.

Na intenção de colaborar com o desenvolvimento de evolução destes tipos de software surgiu o sentimento de colaboração e a demanda de entender e trabalhar com esta tecnologia para sua evolução.

1.1 Objetivo

O principal objetivo deste trabalho foi contribuir para o desenvolvimento e aprimoramento da suíte de Jogos Educacionais GCompris, produzindo uma atividade de pensamento computacional para ser integrada ao software.

1.2 Justificativa

O uso de recursos tecnológicos no ambiente educacional tem se tornado cada vez mais comum. Estes recursos proporcionam aos estudantes a oportunidade de reforçarem e aprimorarem o conhecimento adquirido de modo descontraído e com eficiência. Para proporcionar igual e maior alcance deste benefício, se faz necessário colaborar para o desenvolvimento de Softwares Livres, pois possuem um grande alcance. No GCompris temos atividades educacionais de diferentes cunhos: leitura, matemática, ciências, e inclusive atividades de cunho tecnológico que proporcionam ao usuário a compreensão do computador e seu uso. Uma das necessidades é desenvolver atividades de pensamento computacional e proporcionar esta atividade construída a partir da perspectiva de usuários.

1.3 Organização do relatório

A organização deste relatório segue da seguinte forma: o Capítulo 3 apresenta a fundamentação teórica dos conceitos utilizados nas atividades desenvolvidas; o Capítulo 4 apresenta o detalhamento das atividades desenvolvidas e, por fim, o Capítulo 5 mostra a conclusão do trabalho realizado e os resultados obtidos.

Capítulo 2

Local e Período do Estágio

O Estágio Supervisionado Obrigatório (ESO) em questão foi realizado no Laboratório Multidisciplinar de Tecnologias Sociais (LMTS), funcionando nas dependências da Universidade Federal Rural de Pernambuco (UFRPE - UAG) situada na Avenida Bom Pastor, s/nº - Boa Vista - Garanhuns/PE.

A realização do estágio ocorreu no período de 12 de agosto de 2019 à 29 de novembro de 2019 com carga horária de 4H diárias, totalizando 300H desde o dia do seu início até o fim.

As atividades realizadas neste estágio e o status do desenvolvimento do mesmo foram acompanhadas pelo Orientador em questão, Drº. Mariel José Pimentel de Andrade, e pelo diretor do laboratório Drº Anderson Fernandes de Alencar.

Capítulo 3

Fundamentação teórica

Nesta seção abordaremos os conceitos utilizados no decorrer deste projeto usados para embasamento para nosso trabalho.

3.1 GCompris

O GCompris é um software que consiste em um conjunto de atividades de cunho educacional com a finalidade de tornar o processo de aprendizagem mais lúdico e produtivo. É um software destinado à crianças de idade entre 2 e 10 anos, possui atualmente mais de 100 atividades que estão distribuídos em 7 categorias, classificadas como na tabela 3.1:

Tabela 3.1: Categorias das atividades do GCompris

Categoria	Atividades
Descoberta do Computador	teclado, mouse, touchscreen, ...
Leitura	letras, palavras, prática de leitura, digitação de texto, ...
Aritmética	operações com números, memorização de tabelas, enumeração, ...
Ciências	controle do canal, ciclo da água, energia renovável, ...
Geografia	países, regiões, cultura, ...
Jogos	xadrez, memória, ligue 4, forca, jogo da velha, ...
Outros	cores, formas, Braille, aprenda a dizer as horas, ...

Fonte: GCompris 2019

O GCompris é distribuído sob licença GNU/GPL, sendo um pacote GNU oficial. Este software está disponível para diversas plataformas GNU/Linux, Windows, MacOS e outras versões do Unix. Sua versão mais atual, até a escrita deste trabalho, é a 0.97 atualizada em 01 de Dezembro de 2019. As instruções para instalação nos diversos sistemas operacionais estão disponíveis no site do GCompris na página ‘Baixar’. Para Windows é possível comprar a versão completa na Microsoft Store, loja virtual do Windows, disponível a partir do Windows 10. Caso a versão desse sistema operacional seja outra, o usuário poderá obter a versão de demonstração do GCompris totalmente gratuita, disponível

para download no site, ou comprar o código de ativação.

O projeto GCompris é armazenado e desenvolvido pela comunidade KDE [3] que, através do esforço de milhares de contribuintes, objetiva oferecer uma experiência inovadora em computação e software livre. Os colaboradores da comunidade KDE podem se envolver nas atividades como: relatores de problemas do software, triagem de bugs, desenvolvedores, tradutores, design de interface visual e humano, colaboradores da documentação, suporte ao usuário, promotores do software, web design, colaboradores no processo de transformação para acessibilidade, e doadores de recursos para manutenção do software.

Assim, diante do exposto, podemos perceber que o GCompris possui uma série de características que o torna uma ferramenta educacional que contempla diversas áreas do conhecimento, além de, por ser livre, não gerar custos ao ser adotado nas redes de ensino e possuir uma comunidade de desenvolvimento ativa. No entanto, foi observado que, na versão do GCompris utilizada nesta pesquisa, foi implementada apenas uma atividade relacionada ao desenvolvimento do pensamento computacional. Tal conceito será discutido na próxima seção.

3.2 Pensamento Computacional

Segundo Wing (2006) o pensamento computacional é um método para solução de problemas baseado nos fundamentos e técnicas da Ciência da Computação. Sua proposta é aplicar habilidades utilizadas para criação de programas computacionais como metodologia para resolução de problemas gerais. Outras áreas do conhecimento também podem se beneficiar de parte das competências da Computação. Wing afirma que essas competências são agrupadas definido através das seguintes características:

- Conceituar ao invés de programar
- É uma habilidade fundamental e não utilitária
- É a maneira na qual as pessoas pensam, e não os computadores
- Complementa e combina a Matemática e a Engenharia
- Gera ideias e não artefatos
- É para todos, em qualquer lugar

Em seu estudo, Nunes (2011) afirma que o raciocínio computacional é intuitivo no ser humano e se manifesta já na idade infantil. Portanto, a criança naturalmente raciocina de forma computacional. Entretanto, a não exploração na formação básica desta habilidade, gera como consequência a perda do raciocínio computacional ao longo do crescimento e

da formação do indivíduo. Introduzir o pensamento computacional no ensino básico pode instigar o interesse pela área da computação, mostrando sua importância na resolução de problemas do mundo atual. Ainda que nem todos os alunos ingressam na área da computação posteriormente, mas essa estratégia educacional cria e intensifica relações interdisciplinares com outras áreas.

Barcelos (2012), através do estudo de Basawapatna et al. (2011), define pensamento computacional baseando-se na identificação de padrões do funcionamento de agentes de software em jogos digitais. Os agentes são programados por estudantes em atividades didáticas, usando uma linguagem de programação visual e baseada na construção de blocos. Se trata de uma proposta mais específica, porém seu objetivo é identificar a possível transferência, pelos estudantes, do conhecimento aos padrões para o domínio da simulação de fenômenos físicos e biológicos.

Os livros que ensinam programação estruturada em sua maioria abordam primeiramente o conceito de algoritmos, uma sequência finita de instruções bem definidas [8], e em seguida o conceito de estruturas de decisão e estruturas de repetição, estruturas usadas para executar, repetidamente, uma instrução ou bloco de instrução por determinada quantidade de vezes [7]. Essa metodologia também é abordada no ensino do pensamento computacional para crianças facilitando a compreensão da programação pela mesma e auxiliando absorver o conhecimento e exercitar o raciocínio lógico através processo de repetição de atividades e desafios.

Segundo Papert (1985) uma característica relevante da programação de computadores é que a criança dificilmente vai acertar na primeira vez que programar, então, ela vai persistindo e resolvendo os problemas até chegar ao resultado que pretende. Assim, os softwares utilizados devem possuir características desafiantes, mas, ao mesmo tempo, lúdicas e estimulantes para que gere engajamento das crianças. Para tanto, é importante levar em consideração as recomendações de design para o desenvolvimento de artefatos digitais para e com crianças, assunto este que será discutido na próxima seção.

3.3 Desenvolvimento de artefatos digitais para crianças

A participação do usuário final no processo de criação do artefato desde o início, permite que o produto seja moldado às suas necessidades e expectativas. O design centrado no usuário é um conceito que se baseia na ideia de que todo e qualquer processo de design precisa ter o usuário como origem e como finalidade na sua concepção e planejamento [13]. Segundo Norman (2006) o Design Centrado no Usuário é aquele capaz de reunir produtos compreensíveis e facilmente utilizáveis através da simplificação de tarefas.

Na construção de produtos envolvendo a participação de crianças é necessário levar em consideração suas características próprias e incluir estas no processo de uma forma

geral.

Para este processo podemos usar o método de Design Participativo que auxilia no processo de desenvolvimento de sistemas de informação, focando nas necessidades dos usuários finais e garantindo uma satisfação efetiva por meio do alto grau de usabilidade, acessibilidade e qualidade do mesmo. É possível usar o Design participativo para coletar, analisar e projetar informações neste processo.

Com a abordagem de design focado na participação da criança como autor, proporcionamos a esta a oportunidade de interagir com outras crianças exercendo seu senso crítico, conhecendo novas tecnologias, e explorando novas habilidades, da mesma maneira proporcionamos ao designer uma melhor compreensão do mundo infantil e uma perspectiva de visão sobre este.

Para sistematização das atividades realizadas no Design Participativo usamos o Design Thinking que é caracterizado como um processo para resolução de problemas, desenvolvido colaborativamente e centrado no humano, segundo Martin (2010), o Design Thinking promove o equilíbrio entre o pensamento analítico e intuitivo, o que permite às organizações gerar inovações para aumentar eficiência e competitividade.

Podemos iniciar a aplicação do Design Thinking que, segundo Brown (2010), possui como base: Inspiração: o problema ou oportunidade que motiva a busca de soluções; Ideação: é compreendido como o processo de gerar, desenvolver e testar ideias; Implementação: o caminho que conduz a ideia a partir da fase de projeto até chegar à vida das pessoas. Esta aplicação se dá com as seguintes fases:

- Descoberta: reconhecer que se tem um desafio e como se pode abordá-lo;
- Interpretação: perceber que aprendeu-se alguma coisa e como se pode interpretá-la;
- Ideação: enxergar uma oportunidade e pensar como pode criá-la;
- Experimentação: identificar a sua ideia e como esta pode ser concretizada;
- Evolução: experimentar algo novo e pensar como é possível aprimorá-lo.

Como afirma Oliveira [16] Design Thinking é um modelo de pensamento. Significa ter a possibilidade de acreditar que se pode fazer a diferença, desenvolvendo um processo intencional para se chegar ao novo, com soluções criativas e criar impacto positivo.

3.4 Reúso e customização

A reutilização é o processo de criar sistemas de software a partir de softwares existentes. Fox e Patterson (2015) afirmam em seu estudo que o reuso surgiu da necessidade de engenheiros de software acompanharem a evolução em massa do hardware produzindo

softwares à altura do poder de processamento que estava disponível associando isto à melhoria de produtividade, os engenheiros desenvolveram então quatro mecanismos fundamentais:

- Clareza via concisão
- Síntese
- Reuso
- Automação através de ferramentas

Enxergando da expectativa de mecanismo de produtividade, o reúso está relacionado à porções de projetos antigos para geração de novos ao invés de escrever tudo do zero. Reinhr [?] mostra em seu estudo que o reuso pode ser classificado de acordo com a forma como é implementado, podendo ser por composição ou gerado. No reuso por composição os componentes a serem reutilizados são vistos como blocos de construção, onde o desenvolvimento do sistema é feito em torno de componentes reutilizáveis disponíveis. No reuso por geração os componentes são produzidos para tratar necessidades específicas de uma aplicação e para serem utilizados posteriormente por outras aplicações semelhantes.

Capítulo 4

Atividades Desenvolvidas

A seguir a apresentação dos resultados deste Estágio Supervisionado Obrigatório.

4.1 Roteiro para processo de customização de ambiente de desenvolvimento do GCompris

O processo para customização do ambiente para desenvolvimento no GCompris está espalhado em diferentes referências, um dos objetivos e contribuições deste estágio é justamente sintetizar essas diferentes referências e facilitar a compreensão da customização do ambiente de desenvolvimento. Para isto foi estudado e investigado como se dá o processo de colaboração de um novo desenvolvedor à comunidade GCompris, um roteiro deste processo foi montado com a finalidade de facilitar a compreensão dos processos necessários para novos colaboradores do GCompris. Este roteiro explana alguns passos como:

- Ideação de uma nova atividade do GCompris
- Ambiente de desenvolvimento
- Inserção da nova atividade
- Padrão dos elementos de programação do GCompris
- Compilação

É importante destacar que não se trata de um tutorial e que não é objetivo deste roteiro explicar comandos de sistemas operacionais ou aplicações.

4.2 Desenvolvimento de nova atividade de pensamento computacional

Como já vimos anteriormente há a necessidade de criar atividades de pensamento computacional em um software que é usado em diversos países. Para isto utilizamos a técnica de reuso de código de uma atividade já existente que visava ensinar o pensamento computacional explorando os conceitos de algoritmo, e de funções.

A partir da atividade existente consideramos desenvolver uma nova atividade para ensino do conceito de loop.

A atividade trata-se de uma atividade onde o usuário poderá controlar um personagem usando instruções como “avance”, “virar à direita” e “virar à esquerda” para guiá-lo através de uma trilha pré-definida até o alvo. Considerando a idade dos usuários, a atividade está no cenário de fazenda onde o porquinho, personagem principal, é guiado até a sua comida, pela trilha já mencionada anteriormente.

Esta atividade é dividida em duas partes, as cinco primeiras fases da atividade o usuário exercita o conceito de sequência de algoritmos, abordado pelo ensino do pensamento computacional para crianças como o conceito chave para início do aprendizado.

A partir da sexta fase o conceito de repetição é abordado, o usuário deve montar o algoritmo a ser repetido e definir o número de vezes que deseja repetir este algoritmo. Entre estas fases temos instruções que orientam o funcionamento da atividade. Para acompanhar o processo de execução do algoritmo criado pelo usuário, a atividade foi programada graficamente para que a instrução executada naquele momento esteja em destaque, auxiliando assim a identificação de possíveis erros no algoritmo criado por este.

Também pensando no melhor aproveitamento das fases e na intenção de instigar os usuários a criarem um algoritmo correto, a atividade conta com um limite de instruções que podem ser usadas por fase. Quando o usuário atinge o objetivo da fase este recebe uma mensagem positiva indicando o bom desempenho e em seguida a fase seguinte é aberta, o contrário acontece quando o mesmo não atinge o objetivo, uma mensagem negativa é mostrada e a fase atual continua na tela com as mesmas instruções escolhidas anteriormente para que este faça as correções a partir de onde parou (Figura 4.1).

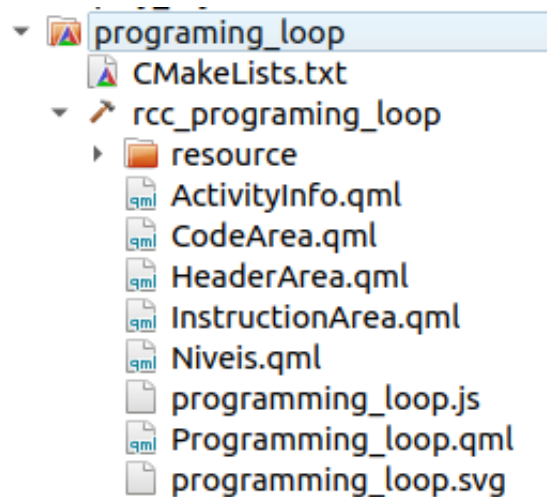
Figura 4.1: Atividade Programming Loop



Fonte: Dados da pesquisa (2019)

As atividades do GCompris possuem uma estrutura dividida em arquivos QML e arquivos JavaScript como representados abaixo (Figura 4.2)

Figura 4.2: Estrutura da atividade Programming Loop



Fonte: Dados da pesquisa (2019)

- *Resource* - contém arquivos de imagens que serão usados na atividade recorrente e os arquivos em qml que definem o comportamento de cada instrução *move-forward*, *turn-left*, *turn-right*, *loop*.
- Arquivos 'qml' correspondem à interface do sistema e à algumas funções específicas

- *ActivityInfo.qml* - direitos autorais da atividade, que devem ser atualizados pelo autor após criação da mesma com descrição da atividade, definição à qual categoria pertence, instruções de ajuda.
- *CodeArea.qml* - representação gráfica da área onde as instruções, escolhidas pelo usuário para cumprimento do desafio, serão inseridas visualmente. Função para acrescentar instruções. Função para reset dos valores da área de instruções.
- *HeaderArea.qml* - representação gráfica do cabeçalho ao qual as instruções escolhidas pertence. Na atividade de “*Programming Maze*” é possível escolher entre inserir instruções no ‘programa principal’ ou inserir instruções no ‘procedimento’. E na atividade de “*Programming Loop*” o usuário pode inserir instrução no ‘programa principal’ ou inserir no comando ‘loop’. Função para identificar quando a ‘*header area*’ foi escolhida.
- *InstructionArea.qml* - contém elementos visuais da instrução. Função de verificação de qual instrução foi selecionada. Função de verificação de onde a instrução será inserida.
- *Niveis.qml* - Arquivo com mapeamento das trilhas de cada nível do jogo.
- *Programming_loop.qml* - elementos visuais da atividade (personagem, trilha, cabeçalhos, botões de navegação, instruções).
- *programming_loop.js* - Declaração de possíveis instruções e onde encontrar o modo de comportamento de cada uma delas. Descrição dos tutoriais de uso do jogo. Funções de *start*, *stop*, iniciar nível, próximo nível, nível anterior . Funções de criação das instruções, função de destruição das instruções, função de rotação, função para executar algoritmo criado pelo usuário, função para executar próxima instrução, função para verificar se o personagem está apontando para o limite do caminho, função para identificar se alcançou o alvo, função para resetar posições do personagem alvo.

4.3 Validação do artefato

Para validação do artefato produzido aplicou-se o uso das técnicas de Design Thinking sob a perspectiva do Design Participativo com um grupo de 15 alunos, com idade entre 11 e 12 anos, estudantes da rede particular de ensino da cidade de Garanhuns, sendo um grupo diversificado de meninos e meninas. Alguns dos integrantes já tiveram contato com sites e repositórios que contêm atividades de ensino de programação para crianças.

Toda pesquisa foi realizada no Laboratório de Pensamento Computacional do Colégio Diocesano de Garanhuns. A participação dos alunos foi previamente autorizada pelos pais ou responsáveis. A instituição que acolheu o desenvolvimento da pesquisa foi informada,

por documento de abertura de projeto, sobre como se daria a realização do mesmo e estava de acordo com todos os procedimentos adotados. As identidades dos sujeitos da pesquisa, bem como das crianças e profissionais foram preservadas aos dados da pesquisa.

O objetivo principal era produzir um software a partir da perspectiva de potenciais usuários, levando em consideração a opinião destes no processo do desenvolvimento ressaltando o que é importante para eles e colocando as suas sugestões no artefato produzido o que facilita a utilização do produto pelo seu público alvo. A partir do processo de pesquisa dado estágio o design para atividade de Programming Loop e para a atividade Programming Maze, já existente, foram criados e implementados, gerando o resultado que pode ser conferido nas imagens abaixo:

Figura 4.3: Estrutura da Atividade Programming Maze - Versão Original



Fonte: GCompris (2019)

Figura 4.4: Estrutura da Atividade Programming Maze - Versão criada neste trabalho



Fonte: Dados da pesquisa (2019)

Figura 4.5: Estrutura da Atividade Programming Loop - Versão criada neste trabalho



Fonte: Dados da pesquisa (2019)

Capítulo 5

Conclusões

O projeto de desenvolvimento de nova atividade de Pensamento computacional para o GCompris foi finalizado, e também foi finalizado o projeto de criação de design para a atividade de pensamento computacional já existente no GCompris a partir da perspectiva do usuário final. As atividades realizadas neste estágio não foram publicadas devido ao processo de avaliação da equipe do GCompris que ainda está em andamento.

A colaboração gerando um roteiro de consulta para o entendimento da customização do ambiente de Desenvolvimento do GCompris foi realizada e este roteiro pode ser verificado neste documento na seção de apêndice.

O artefato produzido conta com o envolvimento de potenciais usuários, e foi produzido a partir da perspectiva deste.

Além do mais o estágio contribuiu para o aprendizado de novas tecnologias e habilidades, proporcionou a mim uma visão melhor do funcionamento do mercado, dado que o seu desenvolvimento foi realizado em um laboratório, e me proporcionou o conhecimento e a exploração de áreas da computação ainda não conhecidas por mim.

Apêndice A

Customização de Ambiente de Desenvolvimento do GCompris

O processo para customização do ambiente para desenvolvimento no GCompris está espalhado em diferentes referências, uma das contribuições deste apêndice é justamente sintetizar essas informações e facilitar a compreensão da customização do ambiente de desenvolvimento. É importante destacar que não se trata de um tutorial e que não é objetivo desta seção explicar comandos de sistemas operacionais ou aplicações.

Para criação de uma nova atividade no GCompris é importante criar um protótipo da mesma, definindo os seus objetivos e estruturando o que será desenvolvido. Este protótipo ajudará a guiar o processo de desenvolvimento e a identificar antecipadamente as possíveis dificuldades do projeto. Depois de esclarecer a ideia, é necessário fazer a proposta da atividade descrevendo o objetivo geral e como esse objetivo vai ser alcançado, além do protótipo da atividade e enviar esta proposta ao e-mail gcompris-devel@kde.org, incluindo também uma *timeline* do desenvolvimento estimando o tempo de uma tarefa. Esta etapa é importante para que a equipe do GCompris possa avaliar e validar a ideia, garantindo que a mesma ainda não tenha sido proposta ou implementada anteriormente por outro colaborador, e sugerindo melhorias na mesma, ou auxiliando em possíveis dúvidas [2].

É aconselhado para quem quer desenvolver uma nova atividade no GCompris, o estudo da tecnologia Qt Quick. O desenvolvimento deve ser feito, obrigatoriamente, numa distribuição GNU/Linux de sua preferência. Após a instalação da distribuição do GNU/Linux, é necessário baixar e instalar a versão mais recente do Qt, em seguida instalar o CMake.

O código do GCompris pode ser obtido no sistema de controle de versão do KDE Git ou no GitHub. O colaborador deve fazer um *fork* do projeto e criar uma *branch* para o desenvolvimento da sua nova atividade. É possível criar uma nova atividade automaticamente usando o script `createit.sh` disponível, através dos comandos:

```
»cd src/activities
»./createit.sh 'my-new-activity'
```

Ou manualmente criando um diretório para sua atividade em `src/activities`. Nele, se cria um `ActivityInfo.qml`, `CMakeLists.txt` e seu ponto de entrada do qml `my-new-activity.qml` [4].

Após a criação da atividade os direitos autorais devem ser atualizados no arquivo de "`ActivityInfo.qml`", após a compilação do GCompris é necessário a verificação da existência da nova atividade.

Uma boa prática para o novo colaborador é observar as atividades existentes no GCompris e verificar o código delas quando tiver um problema. Quando implementadas, as atividades devem ter uma boa aparência e serem utilizáveis em diferentes dispositivos: telefones, tablets e desktops. O teste deve ser feito para os três dispositivos pois a resolução e o valor de dpi podem diferir bastante.

O formato 'svg' é preferível para as imagens e ícones das atividades, caso as imagens não estejam disponíveis no openclipart.org usamos um script, em python, disponível no site, para conversão de imagens 'png' para 'svg' [10].

As atividades do GCompris também usam uma fonte padrão, definido por GCText para garantir a homogeneidade do texto exibido no aplicativo, como no exemplo do código abaixo:

Código A.1: Código do padrão GCText

```

1 GCText {
2   id: text
3   text: Text
4   fontSize: 14
5   font.weight: Font.DemiBold
6   color: white
7 }
```

Fonte: GCompris (2018)

As atividades devem adaptar seu conteúdo adequadamente quando a janela é redimensionada e devem suportar a rotação da tela.

No arquivo QML apenas pequenos elementos de JavaScript devem ser mantidos, toda a lógica do jogo deve estar em seu arquivo JavaScript, facilitando para outros colaboradores a leitura e entendimento da lógica da atividade. Os arquivos QML devem ser vistos como a descrição da interface gráfica e o JavaScript como a lógica do jogo.

Para facilitar suporte às ferramentas de tradução das atividades do GCompris, cada vez que aparece uma *String* a mesma deve conter `qsTr()`:

Código A.2: Código para String

```

1 GCText {
2   text: ApplicationInfo.isMobile ?
3     qsTr("Tap both hands at the same time, " +
4       "to make the ball go in a straight line.") :
```

```

5     QString("Press left and right arrow keys at the same time, " +
6           "to make the ball go in a straight line.")
7 }

```

Fonte: GCompris (2018)

Para melhor posicionamento dos itens visuais do seu projeto é necessário definir relacionamento entre estes usando *anchors*, isto facilita o processo de organização de diferentes itens na tela. Como no exemplo abaixo:

Código A.3: Código para uso de anchors

```

1 GCText {
2     id: instructions
3     anchors {
4         left: parent.left
5         right: parent.right
6     }
7 }

```

Fonte: GCompris (2018)

Quando a atividade estiver pronta para ser submetida para revisão é necessário fazer uma refatoração do código, garantindo assim que o *commit* não contenha variáveis não usadas ou arquivos temporários, e que todas as mensagens de advertências de compilação estejam resolvidas.

Para compilar o GCompris para Ubuntu você precisa ter um sistema operacional ubuntu instalado e o ubuntu sdk [10]. Em seguida clone o código fonte em uma pasta GCompris-qt. Logo após você deve criar o ambiente chroot e fazer login nele.

```
»sudo click chroot -a armhf -f ubuntu-sdk-14.10 create
```

```
»sudo click chroot -a armhf -f ubuntu-sdk-14.10 run
```

Dentro do chroot você precisa instalar manualmente o módulo QTQuickControls e executar o comando de compilação:

```
»apt-get install qml-module-qtquick-controls:armhf
```

```
»cmake -DWITH_UBUNTU=on -DUBUNTU_CLICK=1 -DQt5_DIR=/usr/lib/arm-
linux-gnueabi/cmake/Qt5/ ../GCompris-qt/ && export QT_SELECT=qt5-arm-linux-
gnueabi && make && make DESTDIR=/tmp/click/ install
```

E então, fora do chroot, você precisa executar o comando click para criar o pacote:

```
»click build /tmp/click/
```

Por padrão, nenhuma tradução está presente no repositório git. É possível obtê-los no KDE svn ¹ e acompanhar o status da migração no portal do GCompris.

O processo de compilar o aplicativo para outras plataformas está também disponível no site do GCompris na área de processo de desenvolvimento [10]

¹<https://websvn.kde.org/trunk/l10n-kf5/>

Bibliografia

- [1] GCompris *Manual*. <https://gcompris.net/wiki/Manual>. Version: 2016.
- [2] GCompris *New contributor*. https://gcompris.net/wiki/New_contributor. Version: 2017.
- [3] KDE *Get Involved/development*. https://community.kde.org/Get_Involved/development. Version: 2019.
- [4] GCompris *Qt Quick development process*. https://gcompris.net/wiki/Qt_Quick_development_process. Version: 2019.
- [5] BASAWAPATNA, A. ; KOH, K. H. ; REPENNING, A. Recognizing computational thinking patterns. *SIGCSE 2011* (2011).
- [6] BROWN, T. Designthinking: uma metodologia ponderosa para decretar o 3m das velhas ideias. (2010).
- [7] FILGUEIRAS, F. *JAVA – Estruturas de Repetição*. <https://tableless.com.br/java-estruturas-de-repeticao/>. Version: 2015.
- [8] FORBELLONE, A. L. V. Lógica de Programação – A Construção de Algoritmos e Estruturas de Dados. *São Paulo. MAKRON* (1993).
- [9] FOX, A. ; PATTERSON, D. A. ; KON, F. ; CORDEIRO, D. ; TRANSLATION OF: FOX, A. *Construindo software como servico (SaaS) : uma abordagem agil usando computacao em nuvem*. 2015. – ISBN 9780984881277.
- [10] GCOMPRIS *Reviewing an activity*. https://gcompris.net/wiki/Reviewing_an_activity. Version: 2018.
- [11] GIET, T. *GCompris*. https://gcompris.net/index-pt_BR.html. Version: 2019.
- [12] MARTIN, R. Design thinking: achieving insights via the “knowledge funnel”. *Emerald Group Publishing Limited* 38, 2 (2010), 37–41.

-
- [13] MARTINS, D. G. ; SCHMIDT, M. C. ; RODRIGUES, R. S. ; VANDRESEN, M. Desenvolvimento de Interfaces Digitais Infantis: Estudo Preliminar sobre Design Centrado na Criança. *Human Factors in Design* 6, 12 (2017), 46–57.
- [14] NORMAN, D. A. O design do dia-a-dia. (2006).
- [15] NUNES, D. J. “Ciência da Computação na Educação Básica”. (2011).
- [16] OLIVEIRA, A. A contribuição do design thinking na educação. *E-Tech* 2, Especial Educação (2014), 105–121.
- [17] PAPERT, S. Logo: computadores e educação. (1985).
- [18] SCHUMACHER BARCELOS, T. Proyecto LATIn View project Smart Ecosystem for Learning and Inclusion-ERANet17/ICT-0076 SELI View project. Version: 2012. <https://www.researchgate.net/publication/256439343>. 2012. – Forschungsbericht.
- [19] WING, J. M. Computational thinking. Communications of the ACM. *Journal of Biomedical Informatics* , 3 (2006), 33–35.