Educitec
Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

Desenvolvimento de uma arquitetura de game para auxiliar o processo de ensino-aprendizagem em programação de jogos digitais no curso técnico de informática

Development of a game architecture to aid the teaching-learning process in programming of digital games in the technical computer course

Jaidson Brandão da Costa
Instituto Federal de Educação, Ciência e Tecnologia do Amazonas
jaidson.costa@ifam.edu.br

Resumo

No contexto do ensino de linguagem de programação, os jogos digitais são destacados como possibilidade promissora no que tange à dinamicidade e autonomia nos processos de ensino. Nessa direção, o presente artigo busca apresentar os resultados alcançados no desenvolvimento de uma arquitetura de game para auxiliar o processo de ensino-aprendizagem em programação de jogos no curso técnico de informática. Pensado e desenvolvido em linguagem Java para plataforma *Android,* em virtude da ementa do curso utilizar o Java como a linguagem de programação a ser ensinada no curso. A arquitetura do game originou-se das concepções pedagógicas e práticas do autor. Ao verifica-se o alto índice de evasão e retenção por causa da disciplina de programação no curso técnico de informática, vislumbrou-se a produção de uma estrutura que fosse possível ensinar programação de computadores por meio de desenvolvimento de games. A aplicabilidade da arquitetura na disciplina de programação Java ocorreu com uso da teoria de Ausbel sobre aprendizagem significativa, isto é, coletando-se os conhecimentos prévios dos discentes para produção dos games e ensinamento de programação. Após a utilização da arquitetura os alunos conseguiram desenvolver habilidade técnicas em programação Java e *Android*, bem como conhecimentos em orientação a objetos, com isso eles produziram um game completo. O índice de reprovação na disciplina caiu para 43% em relação ao ano anterior e o de desistência, por causa da não assimilação na disciplina, caiu para 35%. Resultados significativos para sociedade e instituição.

Palavras-chave: Arquitetura. Ensino. Programação.

Abstract

In the context of programming language teaching, digital games are highlighted as a promising possibility in terms of dynamism and autonomy in teaching processes. In this direction, the present article seeks to present the results achieved in the development of a game architecture to support the teaching-learning process in game programming in the technical computer course. Designed and developed in Java language for the Android platform, by virtue of

Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

the course menu use Java as the programming language to be taught in the course. The architecture of the game originated from the pedagogical and practical conceptions of the author. When the high rate of evasion and retention is verified because of the programming discipline in the technical course of computer science, it was glimpsed the production of a structure that could teach computer programming through game development. The applicability of the architecture in the Java programming discipline occurred with the use of Ausbel's theory on meaningful learning, that is, collecting students' previous knowledge for game production and programming teaching. After using the architecture the students were able to develop technical skills in programming Java and Android, as well as knowledge in object orientation, with that they produced a complete game. The rate of disapproval in the discipline fell to 43% in relation to the previous year and the dropout rate, due to non-assimilation in the discipline, fell to 35%. Significant results for society and institution.

Key words: Architecture. Teaching. Programming

Introdução

Os recursos computacionais estão cada vez mais frequentes no processo de ensino-aprendizagem. As novas tecnologias se propõem a modificar os métodos tradicionais de ensino, facilitando o aprendizado dos alunos e também alterando a forma como o professor ministrará as aulas. Observou-se o crescimento no uso de recursos tecnológicos devido à diminuição dos custos dos computadores, *smartphones* e *tablets* e ao surgimento de novas ferramentas de software.

Entretanto, para Valente (1993) a mera utilização de recursos tecnológicos na escola, sem nenhuma associação pedagógica elaborada para as práticas não proporciona vantagem alguma no aprendizado dos discentes. Os egressos do ensino médio, de maneira geral, dos municípios do interior do Amazonas são jovens que não conseguiram ingressar no ensino superior, com isso o mercado de trabalho pode deixar de ser uma alternativa para ser a única possibilidade de vida desse cidadão e por este motivo procuram os cursos técnicos em instituições públicas, para fins de qualificação e preparação para o mercado de trabalho.

No entanto, no curso técnico de informática os alunos se deparam com as disciplinas de programação de computadores que abordam assuntos não triviais de serem compreendidos. Muitos estudantes tem dificuldades em algoritmos e linguagem de programação, consequentemente, por não adquirirem habilidades técnicas para desenvolverem um software, se desmotivam, e são levados muitas vezes a evasão no curso.

Para Prensky (2003) a motivação é o elo para a aprendizagem e que o ser humano com objetivo se torna fascinado em conquistá-lo, o que faz de jogos um excelente instrumento para aprender, sendo uma alternativa a ser implementada. Logo os Games proporcionam aos estudantes a liberdade e

motivação para o engajamento nos diversos assuntos que são abordados utilizando-se jogos digitais.

Um dos desafios do professor é despertar o interesse de seus alunos pelo aprendizado, embora a tecnologia auxilie no ensino, porém muitas vezes não se obtém êxito por inúmeros fatores dos quais se destacam conforme Simões, Redondo e Vilas (2013) os “nativos digitais” que são desprendidos do modelo de aprendizagem da escola, ocasionando níveis de motivação baixos, prejudicando seus resultados de aprendizagem em virtude de o ensino não correspondem suas expectativas e nível. No entanto, levando em consideração que a área de educação está sempre em busca de novas maneiras de prender a atenção dos estudantes e tornar o processo de ensino-aprendizagem mais significativo para os mesmos, viu-se a oportunidade com isso de pesquisar alternativas para o ensino de programação de computadores usando desenvolvimento de games.

Observando-se este cenário realizou-se o presente trabalho, que por sua vez trata-se do Desenvolvimento de uma arquitetura de game para auxiliar o processo de ensino-aprendizagem em programação de jogos digitais no curso técnico de informática.

O objetivo desta pesquisa propôs-se a utilização da arquitetura de um game como ferramenta de apoio para facilitar o processo do ensino-aprendizagem de linguagem de programação, com isso usando a produção de jogos digitais, durante o curso técnico de informática com intuito de minimizar a evasão e assim como também tornar os alunos empreendedores para comercialização dos seus próprios aplicativos e games nas lojas digitais.

A arquitetura de game desenvolvida para apoio ao processo de ensino-aprendizagem de programação

Notou-se em cursos técnicos de informática que a disciplina de programação de computadores, na maioria dos casos, há um grande índice de reprovação, ocasionando evasão e retenção. Contudo, isso ocorre por inúmeros fatores e um deles é a falta de motivação do discente por não conseguir desenvolver software. De acordo com Alves (1982) o professor é um fundador de mundos, mediador de esperanças, pastor de projetos é aquele que tem o papel de motivar os alunos no processo de ensino-aprendizagem.

Entretanto, Ausubel (1982), em sua teoria da aprendizagem conclui que se deve valorizar os conhecimentos prévios dos alunos, possibilitando a utilização de mapas conceituais agregando os conhecimentos já estabelecidos com outros que serão descobertos, viabilizando uma aprendizagem com eficácia fornecendo prazer a quem ensina e a quem aprende.

Pensando-se em alternativas para resolver ou minimizar a problemática no ensino de programação desenvolveu-se um arquitetura de game, isto é, uma *game engine* que conforme Battaiola (2000) é definida como o motor do jogo, ou seja, é o mecanismo que controla a reação do game em função das ações do jogador. Para Amory (2001) a utilização de jogos digitais no processo de ensino-aprendizagem é de suma importância, haja vista que os jogos motivam as

Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

funções cognitivas e a curiosidade do aprendiz, pois os games permitem a experimentação e a exploração do aluno.

Tendo em vista a abordagem dos autores percebeu-se nos conceitos que a utilização dos games digitais no ensino estimula os estudantes, neste trabalho optou-se em vez da utilização de jogos prontos, usou-se o desenvolvimento de games de autoria própria dos alunos, a partir dos conteúdos da disciplina de programação e utilizando a *game engine* como apoio no processo de ensino-aprendizagem.

Estrutura da arquitetura desenvolvida para facilitar a compreensão do aluno na disciplina de programação

Para iniciar o desenvolvimento da *game engine* deste trabalho precisou-se observar a ementa do curso técnico de informática, a fim descobrir qual a linguagem de programação orientada a objetos que foi definida para ser abordada no curso. Identificou-se que a linguagem de programação Java foi determinada a ser usada. Com essa informação partiu-se para a segunda etapa, que por sua vez se trata da escolha do ambiente onde se pretende executar os jogos produzidos.

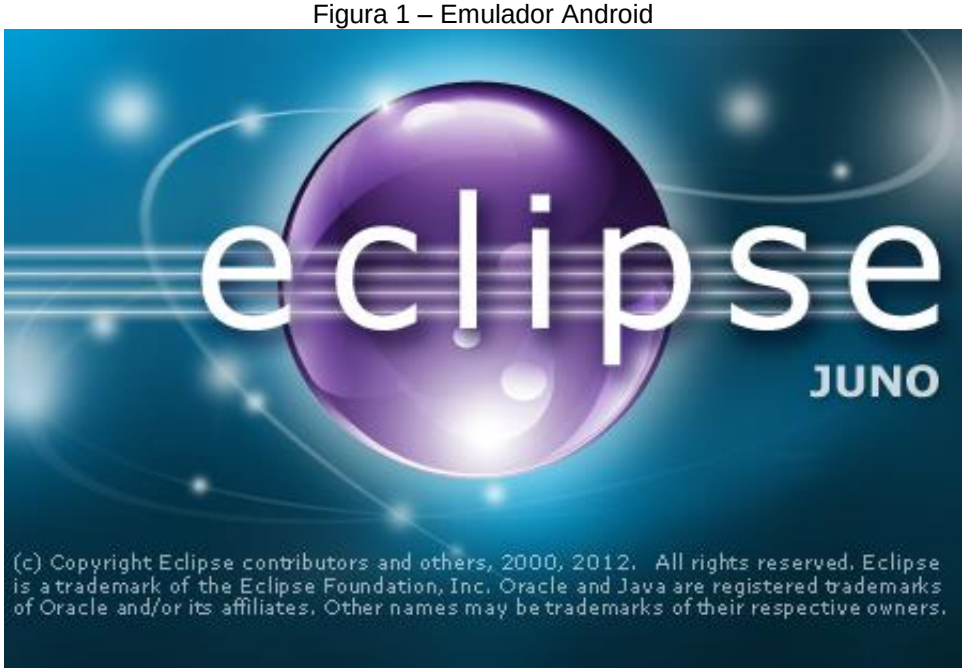
Com a expansão da utilização dos dispositivos móveis e, tendo em vista que conforme Lecheta (2010) a linguagem Java é utilizada no desenvolvimento na plataforma *Android* possibilitando a construção de aplicações ou jogos para celulares e tabletes. Logo, optou-se, para criação da *game engine,* o *Android* com o intuito de que os estudantes aprendam programação a partir da produção de games integrando o conhecimento para desenvolverem em dispositivos móveis, à medida que utilizem a arquitetura durante as aulas.

A partir da compreensão da arquitetura *Android* iniciou-se o desenvolvimento da *game engine* com classes bases para controle do som, imagens, cenários, botões de ações *multi touch* e persistências dos dados em arquivos. Criaram-se também mecanismos de acelerômetros para controle do *multi touch* para a utilização no emulador de celular *Android*, visto que as telas dos computadores dos laboratórios de informática não são *touch screem,* inviabilizando a ação na tela, porém o acelerômetro contribui para que haja essa ação no game pelo emulador.

Contudo, escolheu-se atender o uso do emulador, em virtude da demanda de estudantes que não possuem telefones ou tabletes, e assim eles podem emular um celular nos computadores do laboratório de informática. A Figura 1 ilustra a *Integrated Development Enviroment* (IDE) chamado de Eclipse Juno. E a Figura 2 ilustra o emulador produzido na IDE.

Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

Figura 1 – Emulador Android



Fonte: Próprio autor (2018)

Figura 2 – Emulador Android

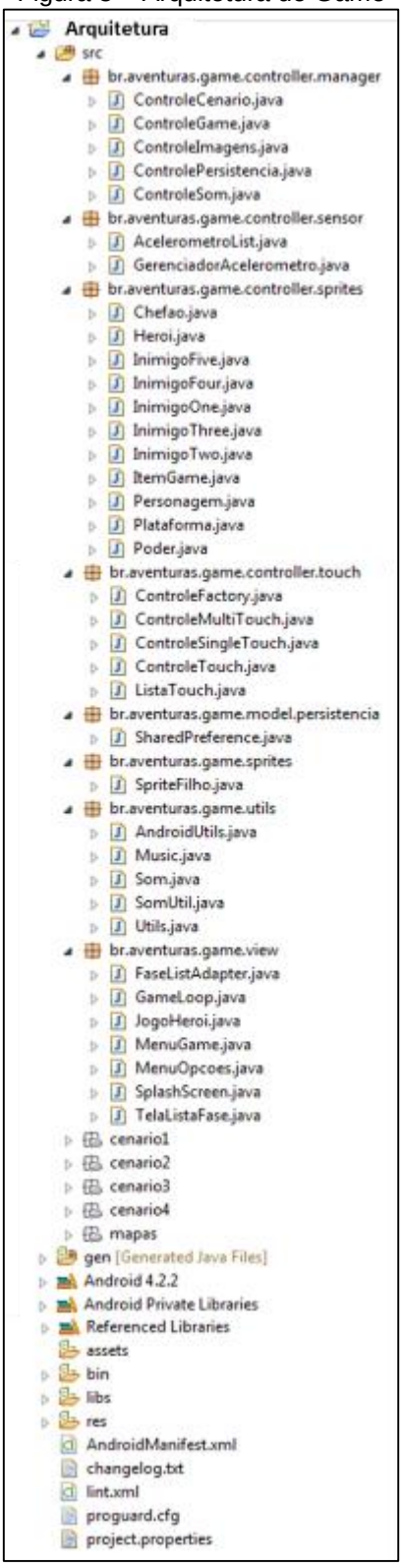
Fonte: Próprio autor (2018)

Baseando-se nos conceitos da composição de um game relatado por Battaiola (2000) que diz que há três partes básicas em um jogo: enredo, motor e interface interativa. Onde o enredo define o tema, objetivos e a sequência do jogo. O motor do jogo controla a reação em função das ações do jogador. E por último a interface interativa que controla as ações no game reportando graficamente um

Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

novo estado do jogo. Pensando-se nesta composição finalizou-se o projeto da arquitetura conforme a Figura 3.

Figura 3 – Arquitetura do Game



Fonte: Próprio autor (2018)

Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

A divisão apresentada na *game engine* conforme a Figura 2 facilita a produção do enredo, motor e interface interativa conforme demonstrado no Quadro 1 a explicação de cada pacote onde foram desenvolvidas as classes, cenários e mapas bases para criação de games a partir da arquitetura, via programação Java.

Quadro 1 – Explicação da Arquitetura (continua)

Pacotes da *Game Engine*	Finalidades	Personalização do código
br.aventuras.game.controller.manager	Foi desenvolvido para gerenciar por meio de classes em Java a lógica para criação de Cenários, Teclas de ação, Imagens, Armazenamento de informações de pontuações e vida, e por fim o controle do Som no game.	As personalizações nas classes deste pacote podem ocorrer no sentido da mudança dos seguintes itens: símbolos dos objetos no mapa, teclas de ação no game, atributos que serão armazenados, modificação na quantidade de vida ou poder no jogo, modificação no controle do jogo, modificação dos itens de pontuação no jogo e passagem de fases.
br.aventuras.game.controller.sensor	Foi desenvolvido para gerenciar por meio de classes em Java a lógica para controlar o acelerômetro e a rotação do dispositivo verificando se está *portrait* ou *landscape*.	As personalizações nas classes deste pacote podem ocorrer no sentido da mudança dos seguintes itens: fixação da orientação do dispositivo e desativar a logica do acelerômetro.
br.aventuras.game.controller.sprites	Foi desenvolvido para gerenciar por meio de classes em Java a lógica para os personagens do game: herói, inimigos, itens do jogo, poder todos herdam da classe do pacote br.aventuras.game.sprites.	A personalização nas classes deste pacote pode ocorrer no sentido da mudança dos seguintes itens: os Personagens podem ser criados novos, modificados a logica de ação e dano ou excluídos por completo. Os itens podem ser modificados alterando-se a logica de pontuação e colisão.
br.aventuras.game.sprites	Foi desenvolvido para gerenciar por meio de uma classe em Java a base da lógica para os itens e personagens do game.	A personalização na classe deste pacote é livre, porém as modificações nela afetará as classes do pacote br.aventuras.game.controller.sprites
br.aventuras.game.utils	Foi desenvolvido para gerenciar por meio de classes em Java a lógica para música do game, animações e leitura de arquivos.	A personalização nas classes deste pacote é livre podendo-se modificar a maneira de tocar a musica, e a maneira de apresentar as *splashscreen* do game.

Fonte: Próprio autor (2018)

Educitec
Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

Quadro 1 – Explicação da Arquitetura (continua)

br.aventuras.game. controller.touch	Foi desenvolvido para gerenciar por meio de classes em Java a lógica para os controles do jogo que são: botão para andar à direita, botão para andar à esquerda desenhados para no lado esquerdo da tela. Criou-se também botão para acionar o pulo e o poder do personagem herói desenhado no lado direito da tela. A logica atende a utilização do *multi touch* e acelerômetro.	As personalizações nas classes deste pacote podem ocorrer apenas para retirada de métodos ou inativação deles, pois o controle dos botões são definidos para serem *multi touch*, restando apenas a desativação dessa ação.
br.aventuras.game.model. persistencia	Foi desenvolvido para gerenciar por meio de classes em Java a lógica para salvar os dados inerentes a: fase, versão, pontuação, itens, poder e vidas.	A personalização na classe deste pacote é livre e sem restrições, visto que se pode escolher o que se deseja salvar no game e não apenas o que foi definido por esta arquitetura.
br.aventuras.game.view	Foi desenvolvido para gerenciar por meio de classes em Java e xmls as telas do game: *splashscreen*, menu, menu de fases, tela de carregamento de fases, telas das fases, tela de game over, tela de vencedor. Controle na restrição ao acesso às fases; Renderização dos mapas, personagens, chão e background do game que são carregados a partir dos pacotes cenario1,2,3,4 e pacote mapas.	A personalização nas classes deste pacote podem ocorrer no controle das fases, no controle dos personagens, no controle dos mapas, na quantidade de vida e poder iniciais, nas opções do menu e menu de fases. Ressalta-se que as modificações na estrutura da tela devem ser realizadas no *xml* na pasta layout.

Fonte: Próprio autor (2018)

Educitec
Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

Quadro 1 – Explicação da Arquitetura (conclusão)

cenario1, cenario2, cenario3 e cenario4	Foram desenvolvidos para gerenciar as imagens do chão e background das fases do game. As imagens devem possuir extensão *png*. A nomenclatura para o background segue essa própria definição e para as imagens do chão tile_a, tile_b até z. Cada vez que for criado um pacote com esse nome e um numero na sequência a arquitetura entende que são imagens de uma fase.	As personalizações nas imagens deste pacote podem ocorrer desde que siga as dimensões do modelo no pacote e a nomenclatura e extensão.
Mapas	Foi desenvolvido para gerenciar os mapas em *txt*, com números, caracteres e símbolos denotando o desenho do mapa. A nomenclatura para cada mapa criado é map1.txt, map2.txt e assim por diante.	As personalizações nos mapas deste pacote podem ocorrer desde que siga as dimensões do modelo no pacote e a nomenclatura, extensão, bem como colocar os números, símbolos e caracteres dos modelos.

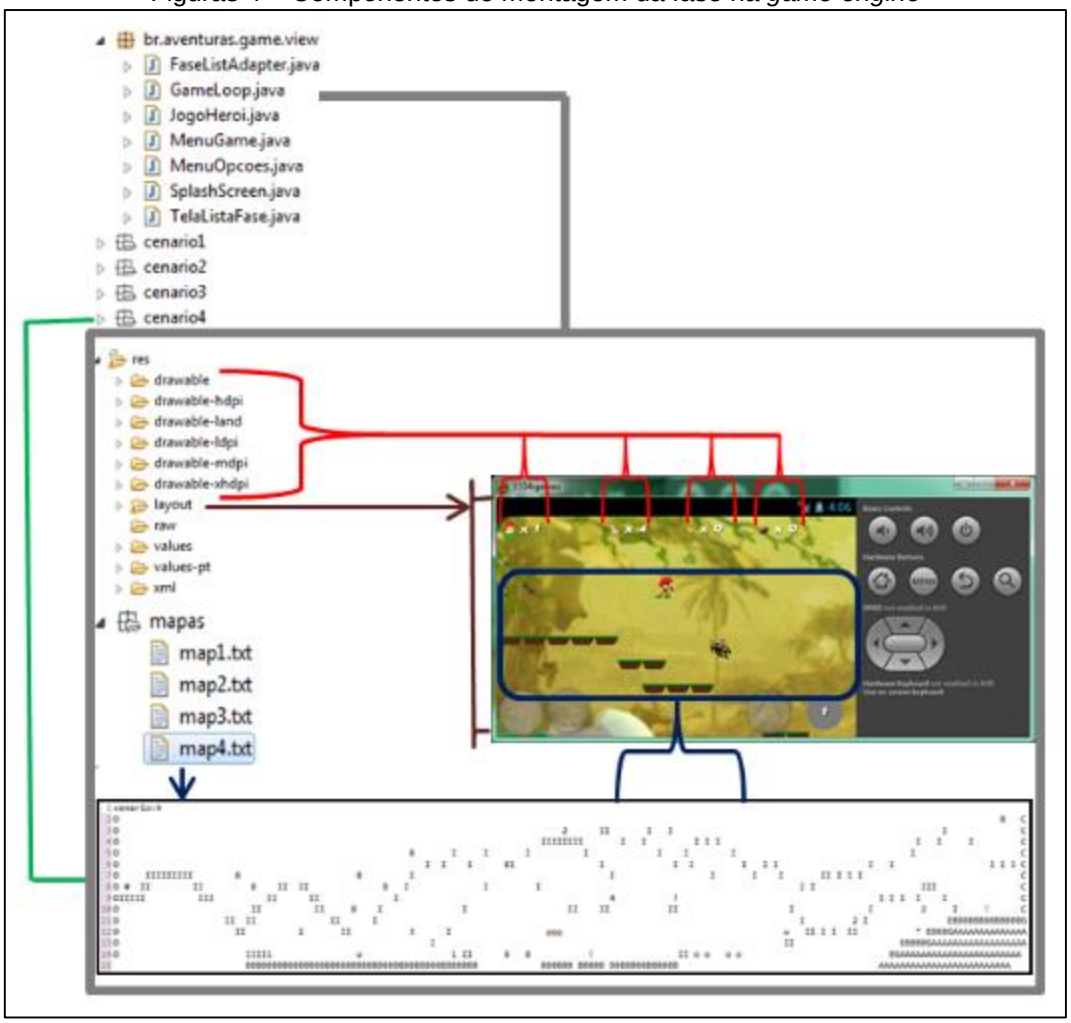
Fonte: Próprio autor (2018)

Para que a estrutura de imagens, mapas e telas na *game engine* na geração das fases se tornasse mais trivial aos alunos iniciantes tanto em linguagem de programação, quanto em desenvolvimento de games, separaram-se as imagens bases do *background* e do chão de cada cenário inerente as fases em pacotes diferentes; produziram-se mapas bases no formato de texto para que o estudante elabore seus próprios de forma flexível; quanto às imagens dos personagens e itens são colocadas nas pastas *drawable*, visto que nela o *Android* escolhe as imagens conforme o dispositivo que está sendo usado.

Na pasta *layout* estão às telas bases produzidas em *xmls*, tais telas são controladas e renderizadas via código, possibilitando ao discente entender todo o fluxo de montagem de uma fase de um game que possua: câmera com técnicas paralaxe, pontuação, controle de colisão, controle *touch screen*, controle de som, criação de mapas dinâmicos, técnicas de *tiles* e *sprites* para criação das imagens. A classe GameLoop do projeto estabelece a lógica para integração de todos os recursos de imagens e arquivos para o jogo. A Figura 4 ilustra uma fase na *game engine* já personalizada e há demonstração dos recursos para que a mesma seja exibida.

Educitec
Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

Figuras 4 – Componentes de montagem da fase na *game engine*



Fonte: Próprio autor (2018)

Diferenciais da arquitetura em comparação as *games engines* existentes para área educacional

Segundo Aranha; Medeiros; Silva (2013) os resultados dos estudos com uso das ferramentas empregadas para o ensino de programação tais como: *PyGame*, *RoboMind*, *Lego Mindstorms*, *Takkou*, *Scratch*, *Alice*, *iVprog*, *Scratch*, *Kodu*, *Game Maker* e *Construct* 2 e *RoboEduc*, foram avaliados de forma positiva no que tange o rendimento dos discentes. Apenas a ferramenta Alice, de acordo com os testes realizados pelos pesquisadores do artigo, não contribui para ensino-aprendizado dos alunos nas disciplinas de programação.

Todavia constatou-se realizando os testes com as ferramentas: *RoboMind, Lego Mindstorms, Takkou, Scratch, Alice, iVprog,Scratch, Kodu, Game Maker e Construct 2* e *RoboEduc* que não é necessário codificar para desenvolver um game, pois há possibilidade de tudo ser feito de forma automática. O estudante apenas irá criar o enredo do jogo, o restante da aplicação é feita praticamente através do mouse, com a ação de arrastar e soltar os objetos no cenário principal.

Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

Mas em contrapartida, a ferramenta *PyGame* exige um esforço maior para programação contribuindo dessa maneira para a abordagem dos conteúdos das disciplinas de linguagem de programação, contudo não apresenta mecanismo base que facilite a criação de mapas, imagens, cenários, objetos no game, obstáculos e física no jogo.

Na *game engine* desenvolvida nesta pesquisa tem inúmeras diferenças referentes às ferramentas empregadas no ensino de programação, destacando-se as seguintes:

a) completamente voltada para a codificação o aluno obterá os métodos de colisão e física para implementação da lógica do game;

b) pode ser utilizada tanto para o ensinamentos de programação estruturada, quanto para programação orientada a objetos em virtude da divisão de classes e métodos;

c) auxilia para compreensão das fases para produção de um game, pois elas são desenhadas em arquivos textos com caracteres referentes as imagens personalizadas no game que são lidas pelo código e substituídas por imagens do cenário definidas pelo programador;

d) voltada ao ensinamento de programação de jogos na plataforma móvel *Android* para estudantes de programação em qualquer nível, visto que a arquitetura do jogo disponibiliza as funcionalidades para movimentação, colisão e mudança de fases apenas por meio de chamada de métodos;

e) empregada para o ensino de linguagem Java na plataforma Android;

f) possui mapas, cenários, física, personagens, som e telas armazenados como base para iniciar um game;

g) flexível as mudanças no cenário, heróis, mapas, poderes e telas

Vantagens e desvantagens da arquitetura para utilização no ensino

As vantagens da utilização da arquitetura de *game* no ensino são atestadas pelo retorno que o professor obtém através da produção dos alunos na programação de games, onde os discentes tomam como base a estrutura da arquitetura, os conhecimentos do processo de desenvolvimento de um jogo e os ensinamentos dos assuntos de linguagem de programação Java. Com isso torna-se possível ao docente visualizar o desempenho dos discentes através dos portfolios gerados por eles.

Sabe-se em relação ao sistema de ensino atual, cujo é forjado em intencionalidades e atividades dirigidas, os games representam tarefas livres e prazerosas. É algo lúdico e está além dos conteúdos agregados ao jogo. Por outro lado, se o uso de jogos no ensino motivam os estudantes, isso já foi comprovado nas pesquisas de Sena e Coelho (2012) que relatam sobre engajamento através de jogos, no entanto, quando se trata de alunos de curso de informática que estudam programação e desistem do curso por não conseguir programar, faz-se necessário uma metodologia diferenciada no ensino.

Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

Pensando-se neste cenário projetou-se a *game engine* para o método de ensino de programação usando games digitais com a base já formada para o aluno, na disciplina de programação que tem em seu plano: Compreender o uso de algoritmos e de estrutura de dados para o desenvolvimento de programas; Elaborar algoritmos e proceder à verificação dos mesmos;Implementar algoritmos em linguagem de programação de alto nível; Elaborar programas para manipular estruturas de dados básicas armazenadas em memória principal e por fim desenvolver programas orientados a objetos.

Com a game *engine* o aluno personaliza o seu jogo na arquitetura, que por sua vez divide-se em: controladores de ação do usuário, aspectos de colisão de personagens, criação de mapas de fases, criação de personagens e cenários individuais via orientação a objetos e controle de pontuação.

Se a utilização de games motiva o aluno, propõe-se ao discente que ele desenvolva os seus próprios jogos, com isso a geração desse desafio oportuniza o engajamento na disciplina por parte dos alunos contribuindo para o ensino-aprendizado, sendo uma das vantagens do uso da arquitetura desenvolvida neste trabalho.

Quanto às desvantagens do uso da arquitetura no ensino de programação, estão relacionados à:

- Criação de imagens, pois caso o aluno desejar personalizar, deverá produzir as imagens em ferramentas e programas de design exigindo um conhecimento prévio para tal atividade;
- Viu-se como desvantagem também a utilização de *xml* para criação das telas do *game*, visto que sintaxe de arquivos com esta extensão não são contemplados na disciplina de programação.
- Apenas contempla jogos em duas dimensões, isto é, games 2D;

A aplicabilidade da arquitetura de game no curso técnico de informática

A aplicabilidade da *game engine* no curso técnico de informática e especificamente na disciplina de programação baseia-se na seguinte definição inerente ao professor da educação profissional:

> O professor da educação profissional deve ser capaz de descrever práticas profissionais (como, por quem e dentro de que condições uma atividade é realizada), de levar em conta o uso que quer fazer desta descrição no processo de ensino-aprendizagem (tipo de apropriação e grau de utilização das técnicas) e de estabelecer a diferença entre ensinar práticas e ensinar os saberes sobre estas práticas (construção mais ou menos elaborada, mais ou menos formalizada destas práticas) (MACHADO, 2008, p. 18).

Conforme a definição citada sobre o professor da educação profissional oportunizou-se o estabelecimento do ensinamento da prática aplicando a arquitetura no ambiente de desenvolvimento de *software* fazendo que o aluno aprenda configurar o seu local de programação e estruturando qual o game pretende desenvolver.

Educitec
Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

No que tange ensinar os saberes sobre essa prática, aplicou-se os conteúdos da disciplina de linguagem de programação Java, abordando teoria de programação estruturada e orientada a objetos, norteando o aluno também onde esses paradigmas são empregados, bem como demonstrando onde e como o mercado de aplicativos e games tem sido atuante. Com isso fazendo-se que o aluno fique curioso a respeito desse nicho de mercado e dessa forma abre fronteira para aplicar o ensino do processo de produção de games, a fim de que o discente consiga usar a arquitetura.

Contudo, a aplicação da *game engine* no curso técnico de informática ocorreu-se no primeiro semestre do ano de 2017. No entanto, a arquitetura foi desenvolvida no segundo semestre de 2016 finalizando-se o desenvolvimento apenas em dezembro do mesmo ano, levando-se 6(seis) meses para codificação. A turma do curso onde se aplicou a arquitetura do game para o ensino de programação possuía 40(quarenta) alunos, usou-se um laboratório de informática com um computador por discente.

O processo de ensino-aprendizado em desenvolvimento de jogos com o uso da arquitetura na disciplina de programação

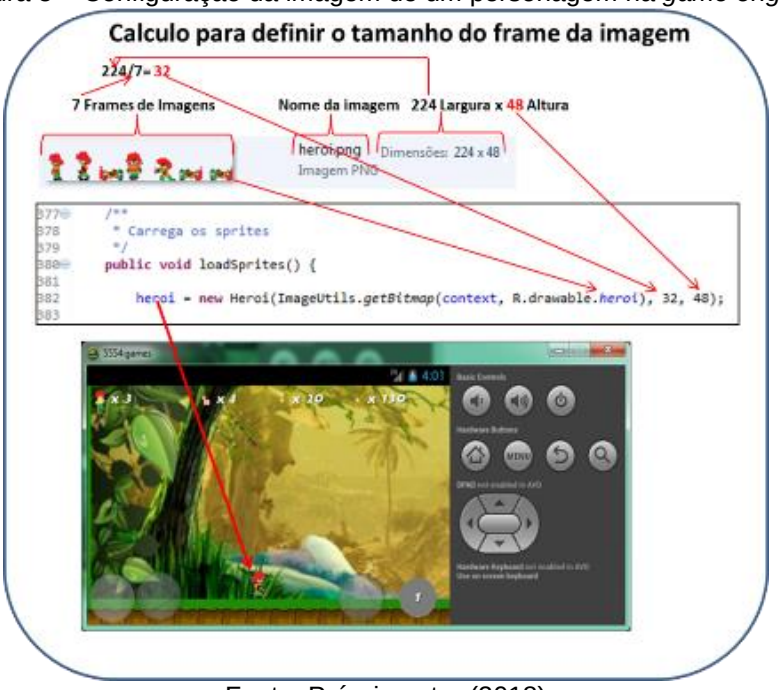
A utilização de tecnologias na educação se tornou uma prática comum nas instituições de ensino. Todavia, a mera utilização de tecnologia no ensino, sem a devida associação pedagógica para as práticas não proporciona uma vantagem no ensino-aprendizado dos alunos (VALENTE 1993).

Frente ao exposto, preocupou-se em utilizar uma das teorias pedagógicas para o uso da *game engine* no processo de ensino-aprendizado, a fim de que os alunos consigam aprender programação utilizando desenvolvimento de jogos. Optou-se pela teoria da aprendizagem significativa de Ausbel (1982) onde em sua teoria defende a valorização dos conhecimentos prévios dos alunos, com isso a construção, o desenvolvimento dos jogos foram diretamente ligados com os sabres das experiências dos alunos, projetando os games relacionados aos filmes que eles assistem, aos desenhos animados, aos desafios da vida cotidiana do ser humano e aos conhecimentos que adquiriram sobre preservação da natureza.

Quanto aos conhecimentos de matemáticas e algoritmos para a programação são utilizados experiências deles no cotidiano e implementado no game, tais como, colisão, correr, pular, atirar pedra, sofrer e ações inerentes à programação do jogo. A Figura 5 ilustra o uso da arquitetura para adicionar uma imagem de personagem na tela do game via programação, através da divisão de frame, isto é, a imagem possui 7 (sete) posições do personagem com altura de 48 cm e 224 cm de largura, para aparecer uma posição da imagem por vez no game, é necessário dividir a largura da imagem pela quantidade de posições do personagem, no caso em questão 224 será dividido por 7, dando como resultado 32 o tamanho do frame que será apresentado. Logo ficará um frame de 32 de largura por 48 de altura sendo exibido .

Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

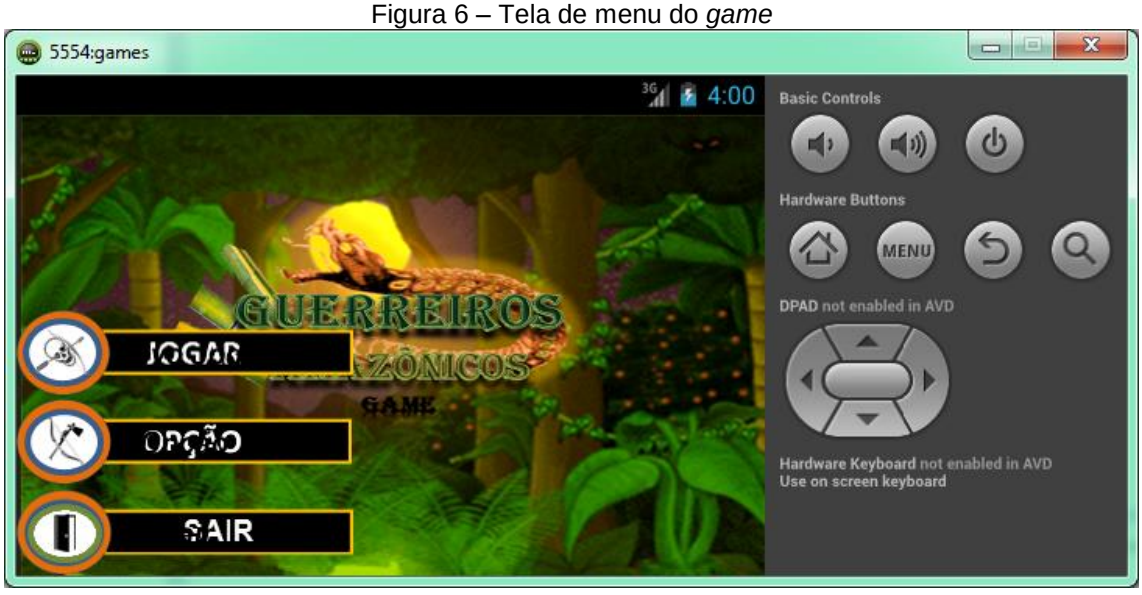
Figura 5 – Configuração da imagem de um personagem na *game engine*



Fonte: Próprio autor (2018)

Os alunos desenvolveram com uso da arquitetura habilidade técnicas em programação Java e *Android*, bem como conhecimentos em orientação a objetos, com isso eles produziram um game completo de preservação da fauna e flora da Amazônia, utilizando personagens das lendas Amazônicas. A Figura 6 ilustra a tela de menu do game.

Figura 6 – Tela de menu do *game*



Fonte: Próprio autor (2018)

Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

A Figura 7 ilustra a tela da terceira fase do game.

Figura 7 – Tela da terceira fase do *game*

Fonte: Próprio autor (2018)

Fatores críticos de sucesso na utilização da arquitetura no curso técnico de informática

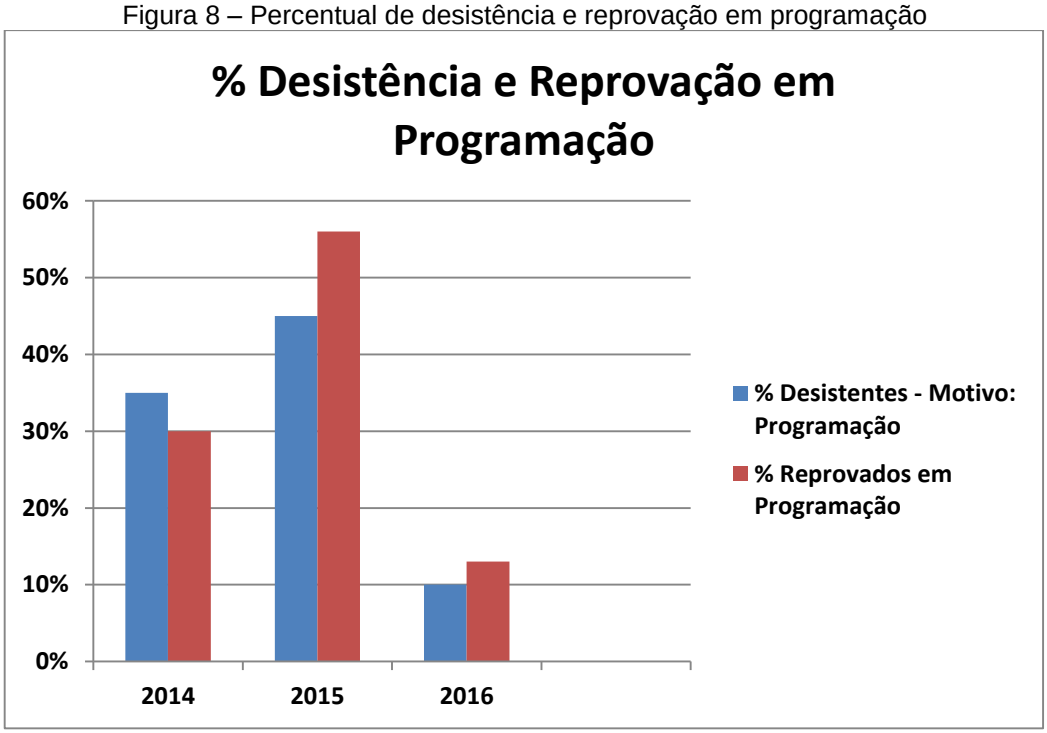
O uso da arquitetura desenvolvida nesta pesquisa contribuiu para a diminuição da evasão e reprovação em programação no curso técnico de informática comparando com dados das turmas nos anos de 2014 e 2015. Observou-se na turma do ano de 2016, apenas 10% dos alunos desistiram por causa da não compreensão da disciplina no total de 40 discentes. Se tratando de reprovação em programação apenas 13%. Isso quando foi utilizada a *game engine*.

Na turma do ano de 2014, 35% dos alunos desistiram por não conseguirem aprender programar e 30% dos que permaneceram reprovaram na disciplina de programação. Contudo, a turma do ano de 2015, aumentou o número de desistentes foi para 45% por causa do não aprendizado em programação e 56% reprovaram na disciplina. Ressalta-se ainda que nas turmas de 2014 e 2015 não foi utilizada a arquitetura e a metodologia baseada na produção de games.

Os dados foram coletados pelo setor pedagógico, esclarecendo-se também que não se utilizou os dados de percentuais inerentes à desistência de fatores de outra natureza a não ser pela desmotivação por não saberem programar, haja vista que tiveram desistências por outros motivos, porém adotaram-se apenas os dados que são relevantes no impacto da programação de computadores para os alunos. A Figura 8 demonstra os dados inerentes à desistência e reprovação em programação nas turmas do técnico de informática dos anos de 2014, 2015 e 2016.

Educitec
Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

Figura 8 – Percentual de desistência e reprovação em programação



Fonte: Próprio autor (2018)

Considerações finais

As contribuições deste projeto podem ser descritas em dois aspectos: pedagógicos e tecnológicos. Sob o ponto de vista tecnológico este projeto disponibiliza à comunidade de Ciência da Computação e áreas afins, especialmente aos desenvolvedores de jogos digitais, um conjunto de soluções baseadas em arquiteturas para desenvolvimento mobile utilizando a plataforma *Android* com a linguagem de programação Java, essa *engine* está em processo de registro de software com disponibilização via serviço de disco virtual google driver.

Sob o ponto de vista pedagógico colabora-se na discussão e ampliação dos resultados envolvendo o uso de desenvolvimento de games digitais como método no processo de aprendizagem significativa dos alunos em programação utilizando *game engine. O*bservou-se com o emprego dessa tecnologia e a técnica do ensino baseada na produção de games valorizando os conhecimentos prévios dos discentes, viu-se minimizar a evasão e a reprovação comprando-se com anos anteriores.

O sucesso da técnica e da ferramenta no ensino tornou-se possível a formação de estudantes com conhecimentos não apenas para produzirem sistemas, mas também jogos digitais. Logo, os resultados preliminares mostram-se interessantes. A documentação de todo o processo de construção da *game engine* não são destacados neste artigo por julgar-se mais adequados a um fórum mais específico. Salienta-se sobre o comportamento dos games produzidos pela arquitetura não ocorreram nenhuma inconsistência e sua implantação em tabletes e celulares funcionou sem erros.

Educitec
Revista de Estudos e Pesquisas sobre Ensino Tecnológico
ISSN: 2446-774X

Referências

ALVES, R. **Filosofia da ciências: introdução ao jogo e suas regras**. São Paulo: Brasiliense, 1982.

AMORY, A. Building an Educational Adventure Game: Theory, Design and Lessons, In: **Journal of Interactive Learning Research**, v.12 n. 23, p. 249-263. 2001.

ARANHA, Eduardo Henrique da Silva; MEDEIROS, Tainá Jesus; SILVA, Thiago Reis da. Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura. In: NOVAS TECNOLOGIAS NA EDUCAÇÃO, 11. 2013, Rio Grande do Sul. **Anais**..., Rio Grande do Sul: CINTED-UFRGS, 2013, p. 1-10.

AUSUBEL, David. P. **A Aprendizagem Significativa: a teoria de David Ausubel**. São Paulo, Moraes, 1982.

BATTAIOLA, A. L. Jogos por Computador – Histórico, Relevância Tecnológica e Mercadológica,Tendências e Técnicas de Implementação, In: Anais da XIX Jornada de Atualização em Informática, **SBC**, v. 2, p. 83-122. 2000.

LECHETA, R. R. **Google Android – Aprenda a criar aplicações para dispositivos móveis com Android SDK**. São Paulo: Novatec Editora. 2010.

MACHADO, Lucília Regina de Souza. Diferenciais inovadores na formação de professores para a educação profissional. **Revista Brasileira da Educação Profissional e Tecnológica**, v. 1, n. 1, jun. 2008. Brasília: MEC, SETEC, 2008. p. 08-22. Disponível em: <http://portal.mec.gov.br/setec/arquivos/pdf3/rev_brasileira.pdf>.

PRENSKY, M. **Digital game-based learning**. Computers in Entertainment. Editora: McGraw-Hill. University of Califórnia. 2003.

SENA, A., & COELHO, D. . **Gamificação - Uma Análise das Técnicas de Engajamento Atualmente Utilizadas**. SBC - ProceedingsofSBGames.2012

SIMÕES, J., REDONDO, R. D., VILAS, A. F. **A social gamification framework for a K-6 learning platform**. Computers in Human Behavior, 2013 pp. 345–353.

VALENTE, J. A. **Por Quê o Computador na Educação**. Em J.A. Valente (Org.), Computadores e Conhecimento: repensando a educação (pp. 24-44). Campinas, SP:Gráfica da UNICAMP. 1993.

Submetido em 26/08/2018.
Aceito em 21/02/2019.