

Um Comparativo entre Ferramentas para o Desenvolvimento de Jogos Educativos Computacionais

Rogério Paulo Marcon Júnior, Giani Petri

Curso de Tecnologia em Sistemas para Internet(TSI)

Universidade Federal de Santa Maria(UFSM)

Caixa Postal 54 – 98.400-000 – Frederico Westphalen – RS – Brasil

rpjunior_1@hotmail.com, gpetri@inf.ufsm.br

Abstract: *In current times, the area within the disciplines of computer science, games are being used to assist learning, these games can be computational or not computational, but for creating a computational game, should be a development platform. This paper presents some tools and show its advantages and limitations in the digital educational games development.*

Resumo: *Nos tempos atuais, dentro das disciplinas da área de Ciência da Computação, estão sendo utilizados jogos para auxiliar no aprendizado, estes jogos podem ser computacionais ou não computacionais, porém para a criação de um jogo computacional, deve haver uma plataforma de desenvolvimento. Este trabalho objetiva apresentar algumas ferramentas e mostrar suas vantagens e limitações no desenvolvimento de jogos educacionais digitais.*

1. Introdução

Ferramentas de desenvolvimento são de uso comum para desenvolvimento de jogos voltados ao lazer, porém, em alguns casos, estão sendo utilizadas para desenvolvimento de jogos computacionais voltados ao auxílio no processo de ensino e aprendizagem. Esta prática ainda se encontra tímida, porém, aos poucos está tentando se introduzir nos métodos de aprendizagem atuais.

No entanto, pode-se encontrar várias ferramentas que podem ser utilizadas para estes fins, cada qual com sua característica específica que pode, ou não ser de bom uso ao desenvolvedor. Com isso, uma boa maneira de descobrir qual seria a mais adequada para o desenvolvimento é um estudo qualitativo de cada uma, focando em o que ela pode fazer, identificando suas vantagens e suas limitações. Desta forma, este trabalho objetiva apresentar os resultados de um estudo comparativo entre ferramentas computacionais para o desenvolvimento de jogos educacionais digitais.

Os resultados preliminares deste trabalho apresentam as vantagens e as limitações das ferramentas, podendo ser usado como referência para desenvolvedores de jogos educacionais digitais.

2. Jogos Educativos

Dentro de algumas disciplinas da área de Ciência da Computação, mais precisamente dentro da área de Engenharia de Software, jogos educativos são muito utilizados para o desenvolvimento de certas habilidades (cooperação, liderança, comunicação, entre

outros), os jogos utilizados hoje em dia, dividem-se em jogos computacionais e não computacionais.

No entanto, com o uso de jogos computacionais, o educador pode facilmente moldar o ambiente de aprendizado para se adequar ao assunto abordado, e avaliar cada educando individualmente, testando-o para ver como reage a solução de problemas.

3. Metodologia

A metodologia que constitui este trabalho baseia-se em uma pesquisa básica qualitativa, para identificar as plataformas de desenvolvimento com maior número de desenvolvedores presentes em fóruns e grupos de desenvolvimento. Para identificar as ferramentas para serem objetos de estudo, foram efetuados testes de desenvolvimento de pequenas aplicações, assim podendo identificar se as ferramentas eram ou não próprias para um jogo educativo, enfatizando suas vantagens e suas limitações.

4. Descrição do Experimento

Nesta seção será descrito brevemente como os experimentos foram realizados utilizando ferramentas de desenvolvimento de jogos. Nos experimentos, jogos plataformas (jogos onde o personagem pula entre plataformas) foram desenvolvidos em cada ferramenta.

4.1 Cocos2D

Com a utilização da ferramenta Cocos2D [Cocos2D 2014], foi realizado um simples experimento de desenvolvimento de um jogo plataforma utilizando a sua versão IDE baseada em linguagem LUA. Nas Figuras 1 e 2, é apresentado o resultado e uma parte do código principal.

```
require "Cocos2d"

-- clog
local clog = function(...)
    print(string.format(...))
end

-- for CCluaEngine traceback
function _G.TRACKBACK(msg)
    clog("-----")
    clog("LUA ERROR: " .. tostring(msg) .. "\n")
    clog(debug.traceback())
    clog("-----")
    return msg
end

local function main()
    collectgarbage("collect")
    -- avoid memory leak
    collectgarbage("setpause", 100)
    collectgarbage("setstepmul", 5000)

    oo.FileUtils.getInstance():addSearchPath("src")
    oo.FileUtils.getInstance():addSearchPath("res")
end
```

Figura 1. Código Fonte Principal.

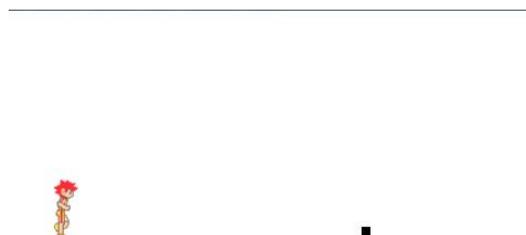


Figura 2. Resultado.

4.2 Construct2

Utilizando a ferramenta Construct2 [Construct2 2014] e o método de Drag and Drop, o jogo plataforma foi desenvolvido com a adição de elementos sendo arrastados e seus movimentos sendo inseridos através da inserção de eventos. As Figuras 3 e 4 apresentam o experimento desenvolvido usando o Construct2.

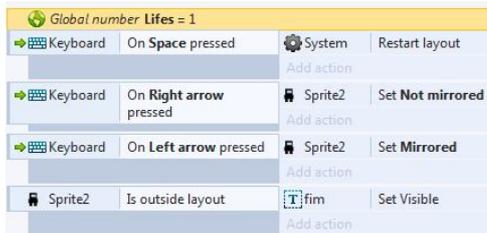


Figura3. Eventos.



Figura 4. Resultado.

4.3 Phaser

Utilizando o framework Phaser [Phaser 2014], o jogo de teste foi desenvolvido utilizando a IDE NetBeans com a linguagem JavaScript. As Figuras 5 e 6 apresentam o experimento desenvolvido usando o *framework* Phaser.

```

(function() {
  var game = new Phaser.Game(
    720, 384, // The width and height of the game in pixels
    Phaser.AUTO, // The type of graphic rendering to use
    // (AUTO tells Phaser to detect if WebGL is supported.
    // If not, it will default to Canvas.)
    'phaser-game', // The parent element of the game
    {preload: preload, // The preloading function
     create: create, // The creation function
     update: update}); // The update (game-loop) function

  function preload() {
    // Load the 'map.json' file using the TILED_JSON special flag
    game.load.tilemap('map', 'assets/map.json', null, Phaser.Tilemap.TILED_JSON);
  }
}
    
```

Figura 5. Código Principal.

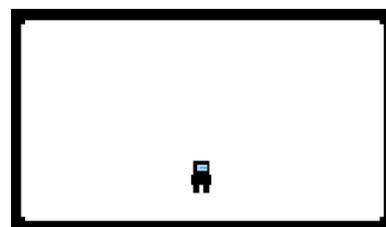


Figura 6. Resultado.

4.4 Stencyl

Utilizando o Stencyl [Stencyl 2014], o teste foi desenvolvido com o método Drag and Drop, com a inserção de eventos e comportamentos para os elementos do jogo, o uso de um kit de imagens também foi utilizado. As Figuras 7 e 8 apresentam o experimento desenvolvido usando o Stencyl.

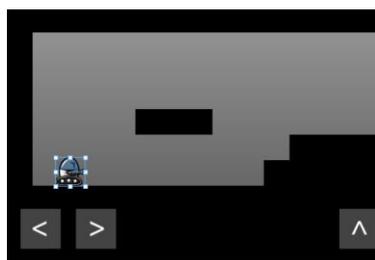


Figura 7. Desenvolvimento da cena.



Figura 8. Comportamentos.

5. Comparativo entre as Ferramentas

Nesta seção encontram-se os resultados preliminares das pesquisas referentes as ferramentas, que serão apresentadas na Tabela 1. A tabela se estrutura da seguinte forma: a primeira coluna apresenta um identificador de cada ferramenta, a segunda coluna apresenta o nome da ferramenta, a terceira coluna refere-se às vantagens encontradas, a quarta coluna apresenta as principais limitações encontradas nas pesquisas e nos testes de desenvolvimento e a quinta coluna, apresenta o modo e linguagem de desenvolvimento.

Anais do EATI	Frederico Westphalen - RS	Ano 4 n. 1	p. 281-285	Nov/2014
---------------	---------------------------	------------	------------	----------

Tabela 1. Ferramentas de desenvolvimento

ID	Nome	Principais Vantagens	Principais Limitações	Modo de Desenvolvimento
1	Construct2 [Construct2 2014]	Grande quantidade de plataformas para uma aplicação final	Versão free com poucos recursos	Drag and Drop com a inserção de eventos
2	Stencyl [Stencyl 2014]	Interface organizada, facilidade na inserção de objetos	Desenvolvimento com restrição de eventos	Drag and Drop com a inserção de eventos
3	Cocos2d [Cocos2d 2014]	Possui crossplataform, open source, leve, fácil aprendizado das APIs	Grande quantidade de tempo exigida para o desenvolvimento	JavaScript e Lua
4	Phaser [Phaser 2014]	Desenvolvimento simples devido à linguagem utilizada, fórum de publicações	Demanda grande de tempo e codificação	JavaScript

Com a utilização de ferramentas como Stencyl ou Construct2, torna-se simples a criação de uma aplicação devido ao Drag and Drop, com apenas alguns cliques, eventos e animações já estão prontas, porém, devido a isso, algumas aplicações podem ser prejudicadas devido a criação desses eventos, tornado-as assim, ferramentas um tanto quanto problemáticas na hora de criar um certo evento mais complicado, já ferramentas como Cocos2D e Phaser, que enquadram-se na categoria de *frameworks*, porém Cocos2D possuindo uma versão IDE, tornam a criação de eventos mais complicadas, simples, devido ao seu método de programação, porém com isso, o desenvolvedor precisa demandar muito tempo para a criação da aplicação, pois, além da programação, deve-se criar os *sprites* e animações com códigos.

Todos estes aspectos foram descobertos através de testes, criações de aplicações, trabalhos acadêmicos, pesquisas e leituras de manuais. Estes testes basearam-se, basicamente, na criação de jogos voltados ao auxílio de aprendizagem, os quais deveriam ter elementos muito específicos, assim, identificando quais seriam suas vantagens e limitações para o desenvolvimento de um jogo educacional.

5. Conclusão

Com a utilização de jogos computacionais em um ambiente acadêmico, o orientador pode moldar o seu método de ensino do jeito que desejar, para tornar suas aulas mais dinâmicas e treinar habilidades básicas exigidas dentro do próprio ambiente acadêmico, como também habilidades exigidas fora dele. Pode-se também utilizar jogos de auxílio em matérias básicas, incentivando a criatividade e resolução de problemas. A utilização dos mesmos não se restringe apenas ao ambiente acadêmico, com a mobilidade hoje em dia, os jogos educativos podem ser utilizados em qualquer lugar desejado.

Portanto, conclui-se que a escolha de uma ferramenta computacional para o desenvolvimento de jogos educacionais digitais é uma decisão estratégica no processo de criação de novas estratégias de ensino. Assim, este trabalho apresentou as principais vantagens e limitações encontradas durante alguns testes básicos das ferramentas e

serve como apoio para auxiliar a tomada de decisão de escolha de ferramentas usadas no desenvolvimento de novos jogos educacionais digitais construídos para auxiliar os professores e alunos envolvidos no processo de ensino e aprendizagem.

Referências

Cocos2d. (2014). Cocos2D - Thousand of Games. Disponível em: <http://www.cocos2d-x.org/>. Acesso em: 05 set. 2014.

Construct 2. (2014). Construct 2 - Create Games Effortlessly. Disponível em: <https://www.scirra.com/>. Acesso em: 05 set. 2014.

Phaser.(2014). Phaser - Desktop and Mobile HTML5 Game Framework.Disponível em: <http://phaser.io/>.Acesso em: 05 set. 2014.

Stencyl. (2014). Stencyl - Create Amazing Games Without Code. Disponível em: <http://www.stencyl.com/>. Acesso em: 05 set. 2014.