The Aedis Aegypt Chronicles

Guide de desenvolvimento

8º período

Vitória de Santo Antão 2016.1

Histórico de Revisão

|  |  |
| --- | --- |
| **Autor** | **Versão** |
| Daniel Batista | 1.0 |

Por onde começar a desenvolver um game?

O dever de todo desenvolvedor de jogos ao pegar o documento de Game Design pela primeira vez, é ler e reler o mesmo até entender o que o jogo é, onde ele se passa, qual a mecânica do jogo e por fim olhar para o lado técnico e definir como irá desenvolver o mesmo. Com a linguagem de programação definida (no caso do game em questão, Lua), o programador deve se atentar as qualidades e boas práticas de desenvolvimento para utilizar o potencial máximo do seu conhecimento junto com o potencial máximo da linguagem a ser utilizada.

**Pré-requisitos básicos para desenvolver um game:**

* O desenvolvedor tem que gostar jogar. E não apenas um jogo, mas sim, vários.
* Precisa conhecer o seu potencial e seus limites.
* Precisa de tempo para analisar, estudar e desenvolver suas partes.
* Precisa conhecer além do seu limite. Inglês é essencial para quem desenvolve games, já que para tal, é necessário muito estudo e a maioria dos materiais estão em inglês.
* Precisa se dedicar ao máximo para construir um game, pois, não é feito do dia para a noite.

**Conheça bem as tecnologias que utilizaremos para desenvolver**

Conhecer bem Lua(não ser um mestre Jedi), porém, saber os fundamentos principais da linguagem é de suma importância para desenvolver na mesma. Assim como aprender a linguagem de programação, deve ser aprendido a utilizar os Frameworks utilizados para facilitar o desenvolvimento na mesma. No caso do nosso game, o Love2d\* veio para nos ajudar.

(\*https://love2d.org/)

**Definição de prioridades e tarefas**

Assim como em qualquer projeto, existem tarefas e elas necessitam de prioridade para você saber quais são mais importantes e urgentes. No desenvolvimento de um jogo, muitas vezes as tarefas podem ter dependência com outras áreas, como arte ou som, porém mesmo com dependências podemos definir prioridades nas tarefas. Podemos olhar para elas pensando exclusivamente na programação, assim você saberá o que deve ser programado primeiro. Por exemplo, não adianta fazer a programação do movimento de um personagem sem antes ter um personagem na tela. Algumas tarefas podem ser difíceis de priorizar, pois muitas podem ter a mesma dependência básica. Por exemplo, com o personagem na tela você pode fazer ele andar, atacar, pular, correr, etc, mas essas tarefas podem não dizer claramente quais são as mais importantes. Para saber o que é mais importante na hora de programar, devemos olhar para o nosso objetivo. Se queremos fazer o personagem lutar, fazer ele se movimentar é mais importante do que fazer ele interagir com portas, por exemplo. Por isso, muitas vezes as tarefas são derivadas de uma tarefa maior, que na verdade é um requisito do jogo, uma mecânica, uma jogabilidade, uma feature, entre outras coisas. É importante também manter o foco, pois se o objetivo é ver o personagem lutar, não é necessário criar todo o nível nestas tarefas, pois o foco dela é jogabilidade.

Fonte: (<http://gamedeveloper.com.br/definicao-de-prioridade-de-tarefas/>)

**Por onde começar a desenvolve o game?**

A melhor alternativa é sempre começar pela mecânica principal do jogo, no nosso caso de um jogo point-and-click, onde o personagem de movimenta para onde o player o joga, a primeira coisa a ser desenvolvida deve ser os personagens que irão se movimentar pela pela, em seguida deve ser desenvolvido o movimento deles. Antes de inserir qualquer ação no personagem como os seus poderes, ou falas, deve-se fazer primeiro o personagem movimentar-se pela tela. Enquanto isso, o cenário do jogo vai sendo construído por outras pessoas, e quando finalizado as duas atividades, o desenvolvedor deve unir A+B e esperar C como resultado, que seria o personagem se movimentando no cenário a partir de um click do mouse. Após feito isso, o jogo estará com boa parte do trabalho em andamento, e poderá assim o desenvolvedor fazer os outros personagens, até mesmo, os estáticos.

*Mantenha tudo simples. Não é necessário um grande código para um grande jogo.*

**Design Patterns**

A respeito de Design patterns, vale salientar, que nem todos devem ser utilizados e para que o jogo fique bem feito, não é necessário utilizar vários. A escolha de um bom padrão de projeto, pode fazer o jogo ir do fracasso ao sucesso ou o contrário.

Segue uma dica de um ótimo design patterns para RPG : http://gamedevelopment.tutsplus.com/tutorials/using-the-composite-design-pattern-for-an-rpg-attributes-system--gamedev-243

Definindo Atividades

|  |  |  |  |
| --- | --- | --- | --- |
| **Atividade** | **Responsável** | **Inicio/Fim** | **Estado(Andamento)** |
| 1. Criar Cenário | Elcio | 16/03/2016 /  30/03/2016 | Em Andamento |
| 1. Criar personagens | ? | ? | Não Iniciado |
| 1. Criar movimentação dos personagens | ? | ? | Não Iniciado |
| Criar atributos dos personagens | ? | ? | Não Iniciado |
| Criar falas, sons e menus do game | ? | ? | Não Iniciado |
| Criar batalhas entre personagem x mosquito | ? | ? | Não Iniciado |
| Criar estágios, missões e ganho de experiencia | ? | ? | Não Iniciado |
|  |  |  |  |

OBS: A maioria das atividades podem ser feitas em paralelos com outras, porem, algumas dependem de outras para serem desenvolvidas.