

Metodologia OriGame: um processo de desenvolvimento de jogos

Rafael A. Santos Vinicius A. Góes Luis F. de Almeida

Universidade de Taubaté, Departamento de Informática, Taubaté, Brasil

Resumo

O desenvolvimento de jogos está relacionado a um cenário por muitas vezes caótico que mistura produção artística, produção de software, inconsistência de requisitos, necessidades particulares e tecnologias que evoluem constantemente. A falta de uma metodologia de desenvolvimento capaz de se adaptar a esses fatores leva equipes inteiras a seguir procedimentos sem um planejamento prévio e muitas vezes distante de normas ou padrões profissional que assegurem a qualidade do produto. Com o objetivo de orientar a criação de jogos, e incentivar novos interessados na área, esse trabalho propõe uma metodologia de desenvolvimento de jogos que se adapta as necessidades de cada projeto e visa garantir um produto final de qualidade.

Palavras chave: criação de jogos, desenvolvimento de jogos, metodologia para jogos.

Abstract

The game development is related with a often chaotic scenary wich involves artistic production, software production, requirement inconsistency, particular needs and constantly evolving tecnology. The lack of a methodology able to adapt to these factors leads entire teams to adopt procedures without a prior planning and many times away from professional standards that ensure product quality. Aiming to guide game creation and encourage new interested parties on the field, this labor proposes a game development methodology that adjusts to needs of each project and intends to ensure an final quality product.

Keywords: game creation, game development, game methodology.

Contato dos autores:

rafael.agt.santos@gmail.com
vinicius_trash@hotmail.com
luis.almeida@unitau.br

1. Introdução

Dada a grande diversidade dos jogos existentes e as peculiaridades encontradas de acordo com seus propósitos, a adoção uma metodologia de desenvolvimento rígida para este tipo de software torna-se inviável. Jogos como os *Role-Playing Games* (RPG), por exemplo, podem exigir uma vasta documentação detalhada que aborde a história do jogo, a descrição das personagens, como estas participam da

trama, e até mesmo a história de vida delas, enquanto nos jogos estilo *First Player Shooter* (FPS) a história pode ser supérflua ou inexistente.

Não somente a documentação como o perfil dos recursos utilizados no jogo sofrem mudanças para se adequar a sua concepção. Jogos para plataformas *web* e *mobile* não podem ser projetados visando alcançar o nível de complexidade audiovisual encontrado nos consoles caseiros, pois existem limitações técnicas que afetam diretamente o desempenho, qualidade e tamanho do jogo.

Conforme citado por Reis Junior et al [2002] “o desenvolvimento de jogos eletrônicos é uma área onde os processos, quando existem, são muito variados e extremamente flexíveis”. A produção artística somada à produção de software, à inconsistência dos requisitos e constantes evoluções da tecnologia resultam em um cenário caótico que leva equipes inteiras a seguir procedimentos “artesaniais” ou “artísticos”. Porém, para qualquer empresa de jogos, publicadora ou desenvolvedores independentes, a adoção de métodos, normas e padrões profissionais é necessária para que se garanta a qualidade do produto [Lavor 2009].

Empresas de desenvolvimento de jogos, conforme adquirem experiência, passam a modelar um processo próprio para a criação de jogos, que com o tempo, vai sendo aperfeiçoado para atender às necessidades encontradas, mas com a competitividade do mercado, dividir esse conhecimento publicamente não é algo comum.

O acesso a uma metodologia de desenvolvimento é então restrita ou não trivial, e os desenvolvedores de jogos costumam, por exemplo, não incluir em seu escopo o planejamento do jogo e partirem direto para sua implementação, o que tem como consequência altos níveis de retrabalho.

Entretanto, não existe uma metodologia conhecida, ampla e simples o suficiente, que atenda a todas as particularidades dos jogos e ao mesmo tempo não sobrecarregue o projeto com documentações e materiais não pertinentes, mas é possível estabelecer diretrizes que servem de base para a concepção de um jogo e mostrar maneiras de como modelá-las para atender as características desejadas.

Esse artigo propõe uma metodologia de desenvolvimento de jogos com base em um amplo estudo sobre as diferentes áreas envolvidas nesse

processo - documentação, criação de áudio, arte 2D, modelagem 3D e implementação - abordando as diretrizes a se seguir para contemplar a proposta de diferentes tipos de jogos, por onde começar e em que focar no início, que pontos se tornam mais ou menos importantes na documentação do jogo, que perfil de áudio escolher para ele, que imagens 2D ou modelos 3D são necessários, como tudo pode ser integrado durante a implementação e o que deve ser realizado em cada uma de suas etapas.

Dessa forma, almeja-se criar uma metodologia que possa ser usada tanto por quem procura aperfeiçoar um processo já adotado como por quem deseja iniciar um projeto de desenvolvimento de jogos, conduzindo o usuário de acordo com o jogo que pretende criar.

2. Trabalhos relacionados

Muitos autores têm adaptado processos clássicos da Engenharia de Software para o desenvolvimento de jogos [Rucker 2002; Flynt e Salem, 2004; Adams e Rollings 2007; Arnowitz et al 2007; Brathwaite e Schreiber 2009; Reyno e Cubel, 2009]. Irish [2005], por exemplo, sugere uma série de etapas na criação dos jogos - concepção, protótipo, pré-produção, produção e controle de qualidade, dentre outros - e apresenta algumas metodologias de desenvolvimento de software que podem ser usadas para essa fim (*Iterate Until You Drop*, *Agile Project Management*, *Code-Like-Hell*, *Fix-Like-Hell*, *Cascade*, *Increments to Completion*), porém, enfatiza que os jogos são diferentes de outros tipos de software, e assim também são os métodos usados para criá-los.

Uma das necessidades na criação de jogos, por exemplo, é o processo iterativo de desenvolvimento. Mesmo com uma documentação adequada, as distintas áreas que participarão de um projeto – arte, música, software – não devem trabalhar de forma isolada, pois o jogo será o resultado do conjunto de seus trabalhos. É necessário, então, um ciclo iterativo que envolva estas áreas e permita identificar quais mudanças são necessárias nos recursos produzidos, para que ao final, o resultado seja um produto uniforme e coerente.

Vários trabalhos relacionados ao desenvolvimento de jogos podem ser encontrados, e em sua maioria, abordam de maneira geral os aspectos envolvidos no processo. Clua e Bittencourt [2005] apresentam um desses trabalhos, e discutem de forma geral o processo de criação de um jogo computadorizado 3D, mas, como a maioria, não tem como objetivo criar uma metodologia de desenvolvimento.

Pereira [2006], por sua vez, além dos fundamentos dos jogos apresenta algumas metodologias utilizadas para sua concepção. Dentre elas, destaca-se a metodologia chamada de *Game Design*, ilustrada na Figura 1, que é voltada exclusivamente para criação de jogos eletrônicos.

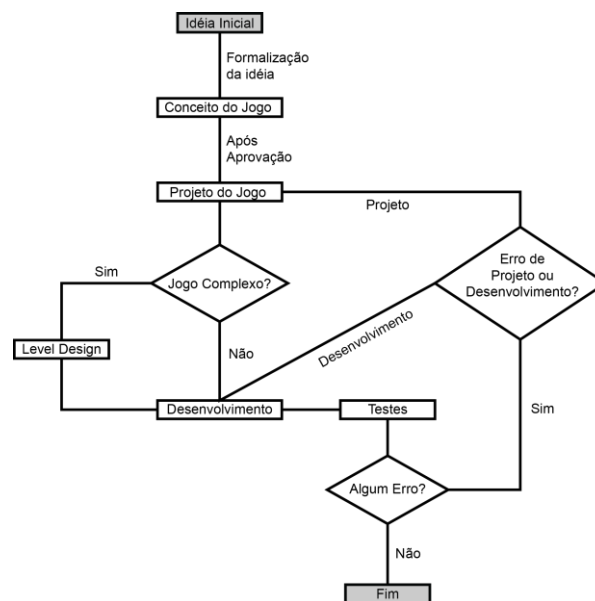


Figura 1: Metodologia *Game Design* [Pereira 2006]

No entanto, o diagrama apresentado na Figura 1 é limitado, não indicando ao usuário, por exemplo, os processos pertinentes ao “Projeto do Jogo” ou ao “Desenvolvimento”.

3. Metodologia OriGame

A metodologia OriGAME surgiu com a criação de um diagrama que relaciona as etapas existentes na produção de jogos em uma visão geral do processo. De acordo com as pesquisas realizadas, um jogo começa pela concepção de suas ideias, que após definidas, são documentadas e avaliadas para determinar a viabilidade do projeto e explicá-lo de maneira detalhada. A partir da documentação, a estética do jogo (áudio e vídeo) é produzida e usada na implementação do jogo, fase na qual os aspectos artísticos são integrados aos computacionais, tendo como resultado o jogo eletrônico.

Em meio a esse fluxo, foram inclusas diversas tomadas de decisão que servem para direcionar a produção do jogo de acordo com o seu planejamento, indicando o que deve ser feito e o que pode ser descartado. Além disso, estão ligados à metodologia os ciclos de testes necessários para garantir o correto desenvolvimento do projeto, passando por protótipos físicos, digitais e funcionais até ser aceito como um jogo completo e divertido.

3.1 Visão geral

A Figura 2 representa a visão geral que deu origem a metodologia OriGame. Todo jogo começa pela concepção de suas ideias, portanto a concepção é o início do processo. Da concepção muitos desenvolvedores costumam partir para a produção e implementação do jogo, mas, visando garantir a

qualidade e correto planejamento do jogo, sua documentação deve ser feita após a concepção.

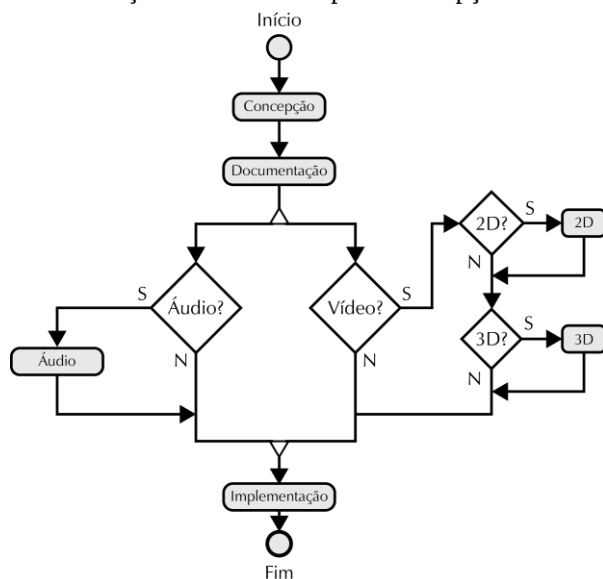


Figura 2: Visão geral da metodologia OriGame

O conteúdo produzido para os jogos se divide em áudio e vídeo, e um jogo pode ter tanto os dois tipos quanto apenas um. Jogos convencionais utilizam os dois tipos de recurso, enquanto jogos para deficientes visuais possuem pouco ou nenhum conteúdo visual, e de maneira semelhante, jogos para deficientes auditivos baseiam-se principalmente no vídeo.

A produção do áudio consiste, basicamente, em planejá-lo e produzi-lo ou adquiri-lo. Já a produção de vídeo divide-se em recursos 2D e 3D, que podem ser usados em conjunto ou não. Uma vez produzidos os recursos necessários, sua integração é feita na implementação do jogo, a última etapa da produção de jogos. Após todo processo de implementação o jogo eletrônico está pronto para ser produzido (salvo questões administrativas, burocráticas, de marketing, etc., que fogem do objetivo do trabalho).

3.2 Visão detalhada

A partir da visão geral da metodologia, destacam-se três fases: Design e Projeto, em que são analisados os requisitos do jogo e definidas as características desejadas; Produção, pautada na criação dos recursos necessários e aprimoramento da estética; e a Implementação, que consiste em unir o que foi produzido, codificar e testar o jogo. Cada uma destas três fases foi expandida em diversos processos e subprocessos, com base em metodologias de desenvolvimento ágil e prototipação, dando origem à visão detalhada da metodologia OriGame. A Figura 3, ao final do artigo, ilustra essa visão detalhada da metodologia.

Embora distintas, as três etapas apresentadas não formam uma sequência linear, ou seja, não se conclui, por exemplo, toda a parte de documentação para começar a produção, ou toda a produção para começar

a implementação. O processo de desenvolvimento é dinâmico, e mudanças de escopo, recursos e tecnologias podem acarretar em atualizações na documentação, mudanças na produção e refatoramento de códigos.

Apesar disso, o jogo provavelmente será restringido por um cronograma, e é importante que as grandes mudanças necessárias sejam detectadas logo no início do projeto, pois quanto mais perto de sua conclusão, menores são as alterações permitidas. Para identificar essas mudanças o quanto antes e garantir que o projeto esteja no caminho certo, um ciclo de testes, avaliações e revisões deve ser realizado durante todo o desenvolvimento, e a adoção de protótipos físicos, digitais e funcionais auxilia nesta tarefa.

3.2.1 Design e Projeto

O primeiro passo para a criação do jogo é a concepção. A partir dela é feito o planejamento do jogo, escrevendo as idéias em um documento de concepção, fazendo uma análise competitiva e estimando um orçamento e cronograma para o projeto. Este é um dos pontos críticos do projeto, pois muitas idéias são recusadas logo neste início. Caso o planejamento seja recusado, deve-se trabalhar em uma nova idéia para o jogo ou modificar a idéia original até que seja aprovada. Caso aprovado, o projeto começa a ser detalhado no documento de design, que descreve o resumo da história do jogo, a mecânica do jogo, o comportamento da inteligência artificial, a progressão do jogo, seus elementos e seu sistema de menus.

Jogos como os de RPG têm como característica serem baseados em sua história, e nesses casos muitas vezes esta história é extensa e contém a vida das personagens, a história antes do estado atual do jogo, e outros fatos que devem ser documentados. Para não sobrecarregar o documento de design, quando há necessidade de documentar este tipo de história, pode-se criar um documento específico para isso chamado *history bible*.

De maneira semelhante, se existe a necessidade de escrever diálogos para o jogo, narrações, textos explicativos que serão apresentados ou um roteiro de falas para as personagens, essas informações são documentadas no *script* do jogo.

O *level layout* é criado quando se pretende criar mapas, áreas jogáveis, fases, etc., para o jogo, o que é chamado de *level design*. Este documento ilustra como serão estas áreas, quais elementos participam de sua composição, qual a disposição destes elementos e a escala de proporções destas áreas. Variando de equipe para equipe ou de jogo para jogo, outros documentos também são criados de acordo com as necessidades encontradas no projeto, pode-se, por exemplo, escrever um documento com padrões de codificação ou manuais para o jogo.

Paralela à documentação do jogo, a construção de um Protótipo Físico pode ser usada para auxiliar na decisão e definição das idéias concebidas. Reproduzir o jogo que será criado usando cartas, dados ou um tabuleiro ajuda a definir as regras e a mecânica do jogo, eliminando a necessidade de programar alguma tecnologia ou produzir recursos estéticos.

Alguns documentos da fase de design e projeto não estão ligados ao fluxo principal da documentação. Isso se deve ao fato de tais documentos dependerem de outras fases da metodologia. A *art bible*, por exemplo, semelhante à *history bible*, serve para documentar toda parte visual do jogo sem sobrecarregar o documento de design, mas não faz sentido criá-la se o jogo não possui recursos visuais. Nela encontram-se imagens de referências para aparência do jogo, imagens conceituais criadas por artistas, esquemas de cores e descrições que possibilitem a produção da parte visual do jogo.

Se o jogo possui vídeos não interativos ou cenas de corte, o *storyboard* é uma maneira de planejá-los, desenhando as cenas, antes de produzi-los, e pode ser tanto parte da *art bible* quanto um documento separado. Já a documentação de áudio serve para manter o controle das faixas de áudio criadas, saber sua duração e em que parte do jogo utilizar, e ter uma breve descrição de cada uma, sendo geralmente criada de acordo com a produção de áudio. O documento de design técnico, por sua vez, contém a arquitetura geral do código do jogo, as principais classes que serão utilizadas e outras informações geralmente definidas na implementação do jogo.

Apesar da dependência de outras fases, é possível prever a necessidade destes documentos e criá-los, quando pertinente, junto com o fluxo principal da documentação, atualizando-os nas fases posteriores.

3.2.2 Produção

A produção do jogo pode começar assim que seu design e projeto tenha definido a maioria dos recursos a serem trabalhados. Grande parte dos jogos é orientada em seu conteúdo gráfico, salvo aqueles que são voltados ao público com deficiência visual, e para isso é necessário criar uma interface gráfica com a qual os usuários terão contato. Fazem parte desta interface a *graphical user interface* (GUI), que abrange os menus, botões e disposição dos itens em tela; e os *heads up display* (HUD), que incluem os mostradores apresentados durante o jogo, barras de energia, radares, pontuações, etc

Se o jogo possui elementos gráficos bidimensionais, a arte 2D destes elementos deve ser criada. *Tiles* (pedaços) são pequenas imagens que organizadas lado a lado, em conjuntos chamados Bricks (blocos), formam os cenários bidimensionais. Esses blocos podem ainda ser sobrepostos em *Layers* (camadas), que darão a sensação de profundidade do

cenário 2D. As Sprites, por sua vez, são imagens, geralmente animadas, que podem se mover livremente pela tela, usadas para representar personagens, inimigos, e outras coisas que não fazem parte do cenário.

Caso o jogo possua elementos tridimensionais, em conjunto ou não com os 2D, a Modelagem 3D destes elementos é realizada. Os elementos dinâmicos são os objetos, monstros, personagens e mecanismos, ou de modo geral, tudo aquilo com o que o jogador interage. Já os elementos estáticos são, por exemplo, os cenários, terrenos e estruturas do jogo. Completando a parte gráfica, muitos jogos possuem cenas animadas conhecidas como *computer generated imagery* (CGI), ou cenas pré-renderizadas, e após criar os elementos 2D ou 3D necessários, é possível gerar estas animações para o jogo.

Outro tipo de recurso utilizado em jogos é o áudio. Para sua produção é necessário determinar o espaço disponível que permitirá seu armazenamento, levando em consideração que o vídeo é que normalmente ocupa o espaço que for necessário. Feito isso, o próximo passo é determinar o áudio necessário para o jogo, ou seja, os tipos e quantidades de sons necessários.

Com estes dados é possível realizar uma estimativa de espaço necessário para a quantidade de áudio planejada. Caso este valor estimado seja aceitável, o áudio pode ser produzido ou obtido e os arquivos finais para o jogo são gerados, atualizando-se a documentação de áudio de acordo com a produção. Caso o valor estimado não esteja de acordo com o espaço disponível, deve-se ou reavaliar o espaço disponível, requisitando mais espaço, ou diminuir a quantidade de áudio necessário.

3.2.3 Implementação

Apesar de ser na fase de produção que os recursos necessários para o jogo são criados, inicialmente, esta fase não deve estar focada em criar todos os recursos do jogo, mas sim em produzir recursos suficientes para a construção de um protótipo digital, a primeira parte da implementação. Os recursos utilizados neste protótipo podem ser apenas provisórios, mas é a partir de sua demonstração que é tomada a decisão final sobre o projeto do jogo. Caso não seja aprovado, dá-se o fim do projeto, e já ter produzido todos os recursos do jogo nesta situação é um grande prejuízo. Caso contrário, o jogo entra em sua versão *alpha* de desenvolvimento e continua a ser implementado.

3.2.3.1 Versão Alpha

Na versão *alpha* do jogo os recursos estéticos restantes serão produzidos e ocorrerá sua integração com os aspectos computacionais do jogo. O *level design* documentado é implementado, os recursos audiovisuais são associados, as questões de tecnologia,

redes e inteligência artificial são codificadas, e o balanceamento do jogo é trabalhado.

Algumas correções de erros no código do jogo são realizadas, mas o foco dos testes unitários e de integração é identificar problemas com os recursos produzidos, por exemplo, uma animação que ficou insatisfatória, ou com o design do jogo, cuja mecânica planejada pode não estar funcionando adequadamente. Além desses testes, nesse estágio da implementação, é possível começar com o chamado *play testing*. É importante assegurar previamente no processo de desenvolvimento que o jogo esteja interessante e divertido, identificando elementos que não estejam adequados ou não proporcionam uma boa experiência durante o jogo. O *play testing* visa identificar esses pontos negativos e, quando feito desde o início e mantido regularmente, serve como um base consistente para a tomada de decisões durante o desenvolvimento.

Assim como no projeto do jogo foi adotado o protótipo físico para ajudar a definir como o jogo funcionará, durante a versão *alpha* do jogo, protótipos funcionais podem ser criados para testar e esclarecer como alguma parte do jogo será codificada. Quatro áreas de investigação podem ser exploradas desta maneira: mecânica, estética, cinestesia e tecnologia.

A mecânica se preocupa justamente com o funcionamento da mecânica do jogo, que pode não funcionar digitalmente como previsto no protótipo físico. A investigação estética é focada em testar como os recursos de áudio e vídeo serão codificados, quais as limitações encontradas em sua integração e se sua produção ficou satisfatória. A cinestesia do jogo é relacionada à como o controle do jogo é sentido. Utilizar mouse e teclado, por exemplo, promove uma cinestesia diferente de um controle convencional, um *joystick*, ou um detector de movimentos como o *kinect*. Por sua vez, a investigação da tecnologia visa principalmente provar que a equipe é capaz de implementar a tecnologia proposta, testar como isto pode ser feito e aplicar o que foi prototipado no jogo.

Durante estas investigações e a codificação do jogo, o documento de design técnico pode ser continuamente atualizado para manter toda equipe ciente das questões técnicas levantadas e esclarecer dúvidas relacionadas à implementação.

3.2.3.2 Versão Beta

A passagem da versão *alpha* para versão *beta* se da quando o jogo alcança um estado funcional, balanceado e internamente completo. Um jogo é considerado funcional quando os jogadores são capazes interagir com ele e entender seu funcionamento sem precisar de assistência.

O balanceamento do jogo está ligado à condição de existir um equilíbrio entre seus elementos e usuário. Uma raça, por exemplo, não deve possuir somente

vantagens perante outras, do contrário não haveria porque jogar com outras raças. Da mesma forma não deve existir uma tática infalível, um inimigo invencível, brechas nas regras que beneficiem um jogador, uma arma ou magia sem utilidade, ou pontos no jogo em que o usuário não possa prosseguir por não ter feito algo anteriormente. Enquanto a estar internamente completo, significa que todas as partes do jogo foram trabalhadas, todas as áreas foram implementadas, os recursos foram todos inclusos, e o sistema prevê soluções para todo tipo de ação do usuário.

O propósito da versão *beta* é testar exaustivamente o código do jogo em busca de erros no código, ou *bugs*, e corrigi-los, tornando o jogo estável. Além dos testes internos, nesta fase é possível realizar os testes *open beta*, liberando parte do jogo para um grupo restrito de indivíduos que ao fazerem uso do jogo podem ajudar a equipe de desenvolvimento a identificar erros ou pontos fracos no produto antes que este seja disponibilizado para todos os usuários.

3.2.3.3 Versão Gold

Quando o código *beta* se aproxima do final é submetido a outros tipos de testes, que caracterizam o início da versão *gold* do jogo. Os testes nesta fase visam garantir que o produto cumpriu com os requisitos planejados (testes de sistema), continua com sua essência divertida e está acessível para público alvo (testes de aceitação), ou seja, os usuários conseguem utilizá-lo por conta própria.

Se o jogo produzido for direcionado a alguma plataforma, o código *gold* é enviado ao fabricante da desta e será submetido a uma série de testes do fabricante, a fim de conseguir a certificação daquela plataforma para produção. As correções apontadas pelos testes nesta versão indicam as mudanças necessárias no planejamento do jogo, e estas são repassadas às fases correspondentes.

Ao final da versão *gold*, o jogo já teve sucesso nos testes de sistema e aceitação, possui o certificado do fabricante para produção, foi testado o quanto possível por erros de programação e está completo, funcional, balanceado e divertido; portanto é considerado o produto final do projeto e está pronto para produção e distribuição.

4. Conclusão

A produção de jogos envolve múltiplos fatores e áreas de conhecimento que dificultam seu processo de desenvolvimento. Sem uma metodologia para organizar esse processo, a falta de orientação pode levar diversos desenvolvedores a desconsiderar etapas importantes na criação de jogos e ter como resultado um produto falho.

Mesmo adaptando métodos da Engenharia de Software, o desenvolvimento de jogos necessita de um processo próprio para sua concepção por envolver fatores que o diferenciam de outros tipos de softwares, como a produção artística, a interatividade e a diversão.

Considerando estas particularidades, o presente trabalho propôs uma metodologia específica que abrange as principais fases de criação de jogos eletrônicos – concepção, documentação, produção e implementação – e prevê testes durante todo o ciclo de produção para detectar mudanças necessárias no projeto e a viabilidade do produto que está sendo criado.

Dessa forma, além de criar métodos, normas e padrões profissionais necessários para garantir a qualidade do jogo, é possível incentivar novos interessados na área de desenvolvimento de jogos, fornecendo a orientação necessária para entender o processo de criação e direcionando o usuário conforme o projeto que pretende realizar.

Por fim, vale ressaltar que a qualidade do jogo a qual se refere esse trabalho diz respeito a um jogo bem feito, e não a um bom jogo, e que apesar de criada com base em um amplo estudo na área de desenvolvimento de jogos, a metodologia OriGAME ainda não foi aplicada na produção de um jogo.

Portanto, para avaliar sua eficiência na prática, é necessário realizar um estudo de caso da metodologia proposta, possivelmente utilizando uma ferramenta de gestão configurável para sua adaptação.

Referências

- ADAMS, E. AND ROLLINGS, A., 2006. *Fundamentals of Game Design*, New Jersey: Pearson Prentice Hall.
- ARNOWITZ, J.; ARENAT, M.; BERGER, N., 2007. *Effective Prototyping for Software Makers*, Morgan Kaufmann Publishers, San Francisco, California..
- BRATHWAITE, B. AND SCHREIBER, I., 2009. *Challenges for Game Designers*, Course Technology Cengage Learning, Boston, Massachusetts.
- CLUA, E. W. G. AND BITTENCOURT, J. R., 2005. Desenvolvimento de Jogos 3D: Concepção, Design e Programação. *Anais da XXIV Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação*, p. 1313-1356, São Leopoldo, Brazil.
- FLYNT, J. P. AND SALEM, O. 2004. *Software Engineering for Game Developers*, Prentice-Hall, Sebastopol.
- IRISH, D., 2005. *The Game Producer's Handbook*, Thomson Course Technology, Boston, Massachusetts.
- LAVOR, R. M., 2009. *Metodologia utilizada no desenvolvimento de games*. Trabalho de Conclusão de Curso (Graduação em Informática para Gestão de

Negócios) Faculdade de Tecnologia da Zona Leste, Centro Paula Souza, São Paulo.

- PEREIRA, G. A., 2006. Projeto e desenvolvimento de jogos computacionais. Trabalho de Conclusão de Curso (TCC) apresentado ao Curso de Graduação em Ciência da Computação, Universidade do Estado de Santa Catarina, Joinville.
- REIS JUNIOR, A. S.; NASSU, B. T.; JONACK, M. A., 2002. *Um estudo sobre os processos de desenvolvimento de jogos eletrônicos (games)*. Departamento de Informática, Universidade Federal do Paraná, Curitiba.
- RUCKER, R., 2002. *Software Engineering and Computer Games*, Addison-Wesley, New Jersey.

Design e Projeto

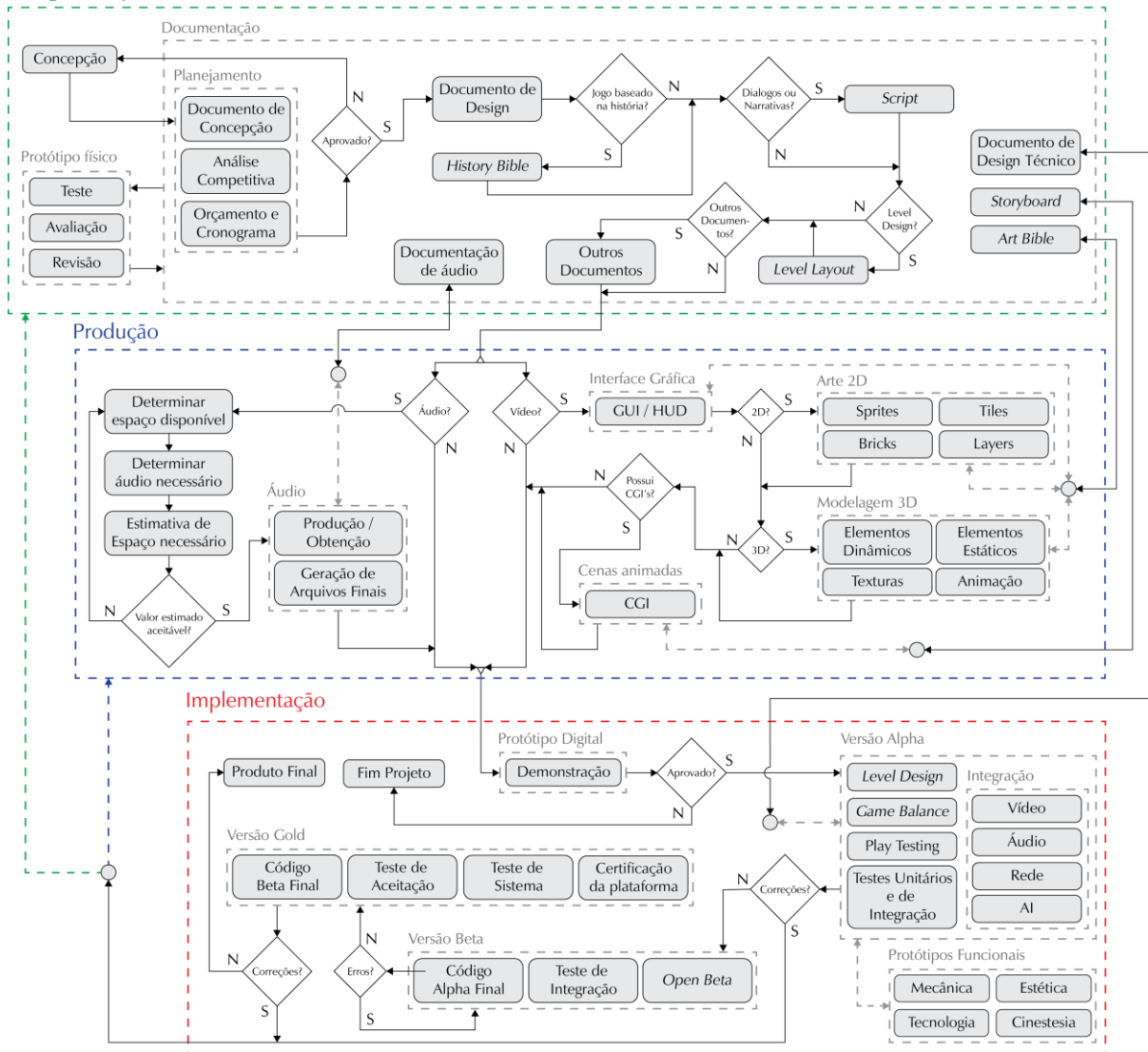


Figura 3: Esquemático analítico da metodologia OriGAME.