



Pós-Graduação em Ciência da Computação

**“BOSS: Concepção de Jogos Digitais Baseada
na Estratégia do Oceano Azul”**

Por

Gabriella Alves Bulhões Barros

Dissertação de Mestrado



Universidade Federal de Pernambuco

posgraduacao@cin.ufpe.br

www.cin.ufpe.br/~posgraduacao

RECIFE, FEVEREIRO/2014



UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

GABRIELLA ALVES BULHÕES BARROS

**“BOSS: Concepção de Jogos Digitais Baseada na Estratégia do
Oceano Azul”**

ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO
EM CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE
INFORMÁTICA DA UNIVERSIDADE FEDERAL DE
PERNAMBUCO COMO REQUISITO PARCIAL PARA
OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA DA
COMPUTAÇÃO.

ORIENTADOR: GEBER LISBOA RAMALHO

RECIFE, FEVEREIRO/2014

Agradecimentos

Agradeço inicialmente a Bruno, por me aguentar durante as crises, me ajudar sempre que foi necessário, estar comigo durante todo o processo, e nunca deixar de repetir que tudo daria (e deu) certo, e que na verdade minha depressão era TPM.

Ao meu orientador Geber Lisboa Ramalho, por ter aceitado me orientar mesmo com a grande quantidade de alunos que já ensinada, por toda ajuda e por tudo que me ensinou. E por me mandar jogar Diabo III.

À professora Patrícia Tedesco, por ter me ajudado tanto durante esses dois anos, sempre tirando minhas dúvidas e sempre feliz em passar seu conhecimento. Devo muito do que aprendi a ela.

Aos amigos queridos que trago de casa: Diogo, Hermes, Vitor, Lay, Amanda, Manu, Jet, João e Rosana.

E aos amigos queridos que fiz aqui: Pamela, Anderson, Thun Pin, Elyda, Luma, Fernanda, Lídia, TJ, Antônio e Giovane. Tomar café com vocês foi o melhor de Recife. Vocês tornaram esse trabalho possível, pois se não estivessem comigo com certeza eu teria desistido. Ou enlouquecido. Ou me mudado para às Bahamas.

À Pamela, em especial, por todo o apoio e os cafés e as conversas!

Aos meus pais, por terem arcado com meus gastos nos primeiros meses que saí de casa.

Aos Átila, Túlio, Vicente, Leonardo e todo o pessoal da Manifesto, por terem me ajudado tanto durante esse trabalho.

Às demais pessoas que me ajudaram nesses dois anos: obrigada!

Finalmente, aos meus jogos, minhas séries, meus livros, meus filmes, minhas músicas, às *fanfics*, ao Drarry, ao Destiel, ao Misha, ao Ben, ao Jensen, Martin e David. Porque sempre deixaram o trabalho menos cansativo =]

Resumo

A indústria de jogos é um mercado arriscado e com pequena margem para erros. A alta competição requer eficiência em seus processos de design, o que resultou em significativas evoluções ao longo dos anos. É na fase inicial do game design, chamada de conceituação, que são definidas as ideias que formarão a base do jogo, definindo o que ele apresentará de inovador em relação a seus concorrentes. Entretanto, apesar da sua importância no sucesso do jogo, a conceituação ainda é pouco explorada na literatura de jogos e tem tirado pouco proveito dos processos de inovação empregados em outras áreas. Duas empresas locais aplicaram, de forma pragmática e preliminar, a estratégia do Oceano Azul à fase de conceituação de dois jogos. Essa estratégia visa inovar na geração de produtos, tornando a competição irrelevante. Ambos os títulos obtiveram ótimos resultados comerciais, onde ficou evidente o papel da inovação. Entretanto, o processo aplicado ainda não havia sido propriamente formalizado e não contava com ferramental de apoio. Diante desse contexto, esse estudo objetivou: i) estruturar, formalizar e evoluir o processo iniciado nas empresas citadas; e ii) construir ferramentas de apoio a tal processo. A ferramenta e o processo evoluído foram comparados e avaliados em relação à versão inicial criada pelas empresas por meio de um caso de teste e questionários. Os resultados indicam melhorias em desempenho e diversidade de soluções em relação a seu predecessor.

Palavras-chave: Conceituação, game design, estratégia do oceano azul, inovação, jogos.

Abstract

Game industry is a risky Market, with little or no margin to make mistakes. Fierce competition requires that design processes become efficient, which results in important evolutions through the years. The initial stage of game design, conceptualization, defines the main ideas that will shape game's fundamentals, defining what it will present of innovative in comparison to competition. However, despite its relevance to a game's success, conceptualization is still little explored in game literature and has taken little advantage of innovation process used in other areas. Two local companies used, in a pragmatic and preliminary manner, the Blue Ocean Strategy in conceptualization of two games. This strategy aims at innovating product generation, making competition irrelevant. Both titles developed achieved great commercial results, where it became clear the role of innovation. Nevertheless, the process had not been properly organized and formalized, and did not have tools to assist it. Within this context, this study aims at: i) organize, formalize and evolve the process initiated by before mentioned companies; and ii) build tools that support such process. The tool and the evolved processed were compared and evaluated, regarding the initial version created by the company, by using a case study, questionnaires and interviews. The results indicate improvements in performance and diversity of solutions, in relation to its predecessor.

Keywords: Conceptualization, game design, blue ocean strategy, innovation, games.

1. INTRODUÇÃO	1
1.1 MOTIVAÇÃO E CARACTERIZAÇÃO DO PROBLEMA	1
1.2 OBJETIVOS.....	4
1.3 METODOLOGIA	4
1.4 ESTRUTURA DA DISSERTAÇÃO	5
2. GAME CONCEPT E INOVAÇÃO.....	6
2.1 GAME DESIGN	6
2.2 CONCEITUAÇÃO.....	9
2.3 CONSIDERAÇÕES FINAIS.....	9
3. PROCESSOS DE INOVAÇÃO.....	10
3.1 BUSINESS MODEL CANVAS (BMC)	10
3.2 LEAN STARTUP.....	12
3.3 BLUE OCEAN STRATEGY (BOS)	13
3.4 CONSIDERAÇÕES FINAIS.....	16
4. APLICANDO BOS À CONCEITUAÇÃO DE JOGOS DIGITAIS.....	17
4.1 EXPERIÊNCIA DA BIGHUT GAMES E MANIFESTO GAMES	17
4.1.1 Caso de Uso nº 1: Fruits Inc.	19
4.1.2 Caso de Uso nº 2: Boney The Runner	22
4.2 LIMITAÇÕES E FALHAS	25
4.2.1 Características muito dependentes do estilo.....	26
4.2.2 Geração trabalhosa de personas	26
4.2.3 Tendências dos designers e diversidade das curvas	26
4.3 CONSIDERAÇÕES FINAIS.....	27
5. BOSS: BLUE OCEAN STRATEGY SYSTEM	28
5.1 METODOLOGIA	28
5.1.1 Estudo da experiência da Manifesto Games e BigHut Games.....	28
5.1.2 Estudo da literatura de game design e inovação	28
5.1.3 Estruturação, formulação e evolução do processo BOSS.....	28
5.1.4 Estudo e desenvolvimento de uma ferramenta de apoio	29
5.2 EVOLUÇÃO DO PROCESSO.....	29
5.2.1 Lentes	29
5.2.2 Perfis.....	30
5.3 ESTRUTURAÇÃO DO PROCESSO.....	30
5.3.1 Definição dos Mercados Primário, Secundário e Terciário	32
5.3.2 Criação de personas ou perfis	32
5.3.3 Pesquisa dos produtos concorrentes.....	33

5.3.4	<i>Análise dos produtos concorrentes</i>	33
5.3.5	<i>Seleção de lentes dos produtos existentes</i>	33
5.3.6	<i>Criação das curvas de valor dos produtos existentes</i>	34
5.3.7	<i>Criação de novas curvas de valor</i>	34
5.3.8	<i>Análise das curvas criadas</i>	35
5.3.9	<i>Seleção do melhor título proposto</i>	35
5.4	FORMALIZAÇÃO DO PROCESSO	35
5.5	CONSIDERAÇÕES FINAIS	36
6.	FERRAMENTA BOSS TOOLS	38
6.1	ABORDAGEM SEGUIDA	38
6.2	DESCRIÇÃO DA FERRAMENTA	39
6.2.1	<i>Arquitetura do sistema e dados técnicos</i>	39
6.2.2	<i>Sobre o algoritmo de geração de curvas</i>	41
6.2.3	<i>Funcionalidades do BOSS Tools</i>	44
6.3	CONSIDERAÇÕES FINAIS	49
7.	EXPERIMENTOS E AVALIAÇÃO	50
7.1	DEFINIÇÃO	50
7.1.1	<i>Objetivos gerais</i>	50
7.1.2	<i>Medidas de avaliação</i>	50
7.1.3	<i>Tipos de avaliação utilizados</i>	51
7.2	EXPERIMENTOS SOBRE O PROCESSO BOSS	51
7.3	EXPERIMENTOS COM A FERRAMENTA BOSS TOOLS	51
7.3.1	<i>Experimento Quantitativo</i>	52
7.3.2	<i>Experimento Qualitativo</i>	52
7.4	RESULTADOS DOS EXPERIMENTOS	53
7.4.1	<i>Discussão dos resultados</i>	55
7.5	CONSIDERAÇÕES FINAIS	57
8.	CONCLUSÕES	58
8.1	TRABALHOS FUTUROS	58
	REFERÊNCIAS	60
	APÊNDICE A	63
	APÊNDICE B	68

Lista de Figuras

Figura 1.1: Telas do jogo <i>Fez</i> .	2
Figura 2.1: <i>Design</i> iterativo, proposto por Zimmerman (2003).	7
Figura 2.2: Processo proposto por Adams (2010).	7
Figura 2.3: Modelo geral proposto por Araújo e Roque (2009).	7
Figura 3.1: Modelo geral do <i>Business Model Canvas</i> .	12
Figura 3.2: Representação da estratégia Lean Startup.	13
Figura 3.3: Relação valor/custo em oceanos azuis e vermelhos.	15
Figura 3.4: Exemplo de um painel de estratégia para empresas cambiais, com 3 curvas de valor.	15
Figura 4.1: Resumo do processo da Manifesto Games e BigHut Games.	18
Figura 4.2: Curvas de valor que representam os jogos Hotel Dash e My Kingdom for the Princess II.	21
Figura 4.3: Exemplo de novas curvas de valor geradas.	21
Figura 4.4: Telas do Fruits Inc. À esquerda, o menu inicial. À direita, uma tela do jogo.	22
Figura 4.5: Telas do Boney the Runner. À esquerda, tela de abertura. À direita, uma tela do jogo.	24
Figura 4.6: Tela de combinação de itens.	25
Figura 4.7: Tela do caça-níquel de itens.	25
Figura 5.1: Fluxo geral de atividade do BOSS.	36
Figura 5.2: Diagrama de atividades do BOSS, com os artefatos de cada atividade.	37
Figura 6.1: Arquitetura do BOSS Tool.	40
Figura 6.2: Tela inicial do BOSS Tools.	41
Figura 6.3: Funcionalidades do BOSS Tools.	44
Figura 6.4: Tela de seleção de jogos.	45
Figura 6.5: Tela de seleção de lentes.	45
Figura 6.6: Tela que verifica se todos os jogos possuem valores para as lentes selecionadas.	46
Figura 6.7: Tela de geração de curvas.	47
Figura 6.8: Tela de inserção de um jogo novo (Parte 1): informações gerais.	47
Figura 6.9: Tela de inserção de um jogo novo (Parte 2): valores e comentários das lentes.	48
Figura 6.10: Tela de inserção de um jogo novo (Parte 3): painel de estratégia.	48
Figura 6.11: Tela de visualização de todos os jogos cadastrados.	49
Figura 7.1: Porcentagem de respostas à pergunta sobre a essência do processo após sua evolução, e classificação da evolução do processo (1 = muito ruim, 5 = muito boa).	53
Figura 7.2: Utilidade das funcionalidades da aplicação (1 = inútil, 5 = muito útil) de geração (como um processo num todo) e somente da geração.	54
Figura 7.3: Utilidade das funcionalidades específicas da aplicação (1 = inútil, 5 = muito útil).	54
Figura 7.4: Classificação da ferramenta (1 = muito ruim, 5 = muito boa), Dificuldade da ferramenta (1 = muito fácil, 5 = muito complicada).	55
Figura 7.5: Melhoria percebida na produtividade e nos resultados produzidos (1 = Atrapalhou, 5 = Ajudou muito).	55

Lista de Tabelas

Tabela 1.1: Fases da pesquisa.....	4
Tabela 3.1: Principais diferenças entre oceanos azuis e vermelhos.....	14
Tabela 5.1: Lentes selecionadas, com o número da lente no livro de Schell (SCHELL, 2008)	31
Tabela 6.1: Problemas sugeridos para cada etapa do processo.....	38
Tabela 6.2: Resultado da análise dos problemas e justificativa de seleção.	39
Tabela 7.1: Comparação entre o tempo gasto com e sem a ferramenta	53
Tabela 7.2: Comparação do tempo gasto por etapa, utilizando a ferramenta, com ou sem dados armazenados previamente.	54

Lista de Abreviações e Siglas

BOS	Blue Ocean Strategy
MVP	Minimum Viable Product
TM	Time Management
HO	Hidden Objects
ER	Endless Runner
EJ	Endless Jumpers
BP	Blitz Puzzles
DAU	Daily Active Users
MAU	Monthly Active Users
BOSS	Blue Ocean Strategy System
SPEM	Software & Systems Process Engineering Meta-Model

1. Introdução

Esse capítulo fornece uma visão geral do trabalho proposto. Ele apresenta o contexto da pesquisa, as motivações e justificativas desse projeto, objetivos gerais e específicos, as contribuições e a estrutura da dissertação.

1.1 Motivação e Caracterização do Problema

A indústria de jogos é um mercado multibilionário extremamente competitivo. Em 2012, ela gerou US \$20,77 bilhões em receita global, e ainda encontra-se em ascensão¹. No Brasil, o quarto maior mercado do mundo em relação a jogos digitais, sua receita chegou a US \$2,6 bilhões, com um crescimento mercadológico de 32% entre 2011 e 2012 e mais de 35 milhões de usuários². Estimativas apontam que, em 2016, o mercado de jogos na Europa chegue a US \$1,9 bilhão, e no Brasil esse valor alcance R\$ 4 bilhões³.

Nesse ambiente competitivo, criar jogos de sucesso é uma tarefa árdua e insegura. Não existem garantias do grau de aceitação de um título, e atividades essenciais, como descobrir o que um jogador deseja, podem não ser suficientes para garantir o sucesso de um jogo (ROUSE III, 2004). Entretanto, existem formas de tentar aumentar as chances de um jogo ser bem-sucedido. Wesley & Barczak (2010) sugerem que, assim como a indústria de jogos evoluiu, os fatores que influenciam no sucesso de um jogo também amadureceram. Por exemplo, a importância dos gráficos diminuiu enquanto interação e aspectos sociais tornaram-se mais requisitados. Dentre as características que ascenderam nas últimas décadas podemos ressaltar *inovação*. (WESLEY; BARCZAK, 2010).

Em um mercado onde existe quantidade massiva, diversificada e crescente de títulos, um jogo inovador pode chamar mais atenção que seus concorrentes. Entretanto,

¹ Números retirados da Associação de *Software* de Entretenimento (*The Entertainment Software Association*): <http://www.theesa.com/facts/salesandgenre.asp>

² Números retirados de Newzoo: <http://www.newzoo.com/infographics/infographic-2012-brazil/>

³ Números retirados do infográfico feito pela Brasil Game Day: <http://www.brasilgameday.com.br/saiba-um-pouco-mais-sobre-industria-de-jogos-brasil-mundo/>

isso cria um novo desafio para game designers: como inovar? É importante ressaltar que essa novidade não tem que ser grande, podendo ser apenas um detalhe. Por exemplo, *Fez*⁴ parece um jogo *puzzle* independente de exploração sem demais atrativos, mas sua movimentação tridimensional dentro de um ambiente bidimensional dá uma nova experiência ao jogador, como mostra a Figura 1.1. Outro exemplo é *Bastion*⁵, um RPG de ação com gráficos bem trabalhados, onde a adição de um narrador que acompanha o jogador durante o jogo, descrevendo o que o mesmo faz e como o mundo reage a ele, foi essencial para torná-lo um sucesso. É possível notar, também, que desenvolvedores independentes são mais adeptos à inovação que grandes estúdios (WESLEY; BARCZAK, 2010).

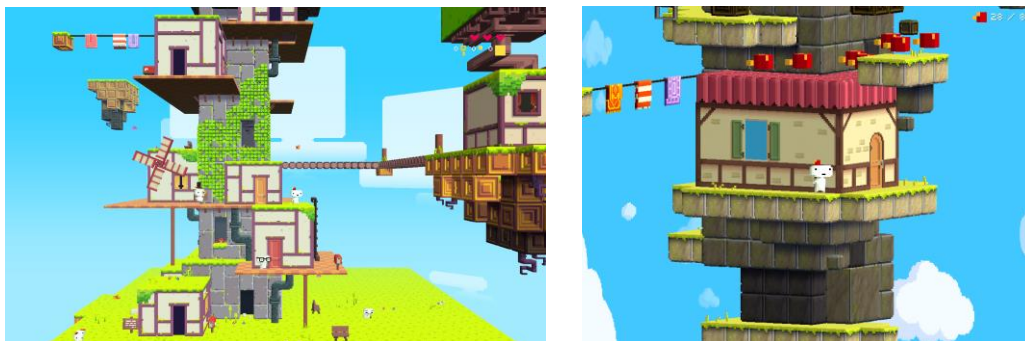


Figura 1.1. Telas do jogo *Fez*.

Na esquerda temos uma imagem do jogo em seu ambiente bidimensional. Ao apertar um botão, a tela gira, como pode ser visto na imagem à direita, mostrando outro lado do ambiente.

Game designers podem aplicar processos de design como forma de auxiliar o método criativo. Nesses processos, existe um estágio inicial chamado de conceituação (do inglês *conceptualization*) ou estágio de conceito (do inglês *concept stage*). Ele consiste em gerar e selecionar as principais ideias de um jogo, definindo suas regras e características fundamentais (FULLERTON, 2008). É um passo importante no processo de *design*, pois um bom conceito pode definir o sucesso ou fracasso de um jogo.

Apesar de sua importância, esse estágio foi pouco estudado, pois costumava ser ignorado em comparação com outras fases do *game design* (e.g. *level design*), e muitas vezes é feito sem uma análise mercadológica apropriada (CRAWFORD, 1982; KULTIMA; KÖÖNIKKÄ; KARVINEN, 2011). Essa falta de influência do mercado

⁴ *Fez* pertence à Polytron. <http://fezgame.com/>

⁵ *Bastion* pertence a SuperGiant Games: <http://supergiantgames.com/index.php/media/>

pode resultar em falta de público-alvo ou espaço comercial bem definidos. Portanto, é importante incorporar dados de mercado no processo. Uso de métricas e análises de títulos são duas formas de incorporar essas informações.

Até onde pudemos constatar empresas de jogos ainda não utilizam, na fase de conceituação do jogo, processos de inovação já relativamente bem difundidos em outras áreas. De fato, fora da indústria de jogos, estratégias variadas foram propostas para ajudar a gerar inovação associada ao mercado, como a Estratégia do Oceano Azul (do inglês *Blue Ocean Strategy* [BOS]), o *Lean Startup* (RIES, 2011) e a técnica Personas (COOPER, 2004), entre outros.

BOS é uma estratégia de mercado que tenta tornar a competição irrelevante criando novos espaços de negócios livres de concorrentes (KIM; MAUBORGNE, 2005). Seu foco é a geração de produtos ou serviços que sejam diferentes de outros já existentes, evitando assim competir com outras empresas. Persona é uma técnica de *design* usada na fase inicial do processo, não estando presa somente à área de jogos. Personas são descrições de usuários hipotéticos, contendo características, personalidades, gostos, desgostos e objetivos. Eles representam, em um único indivíduo, um tipo geral de usuário (BLOMQUIST; ARVOLA, 2002).

A junção de técnicas BOS e personas já foi aplicada ao processo de criação de jogos pelas empresas recifenses Manifesto Game Studio⁶ e BigHut Games⁷ (BARROS et al., 2013). Utilizando esse processo, elas desenvolveram dois produtos: Boney The Runner and Fruits Inc. Ambos obtiveram bons resultados comerciais, e críticas positivas indicaram inovação como um papel importante no processo. Entretanto, o mesmo ainda se encontrava em um estado muito pragmático e não havia sido suficientemente estruturado ou formalizado.

Esse trabalho visa estudar a aplicação das técnicas BOS e Personas a criação de conceitos de jogos, por meio da estruturação, formalização e evolução de um processo iniciado pela Manifesto Game Studio e BigHut Games. BOS permite a inserção de informações mercadológicas na criação de produtos, com intuito de gerar inovação. Personas buscam garantir guias para modelos de usuários que representem o público-

⁶ Manifesto Game Studio: <http://manifestogames.com.br/>

⁷ BigHut Games: <http://www.bighutgames.com/>

alvo. Com isso, espera-se aumentar as chances de um título ser bem sucedido. Ademais, esse trabalho propõe uma ferramenta que auxilie o processo.

1.2 Objetivos

Essa seção apresentará os objetivos gerais e específicos desse trabalho. O objetivo geral dessa dissertação consiste em (1) estruturar, formalizar e evoluir um processo de criação de conceito de jogos baseado em uma estratégia de negócio que visa inovação, com o intuito de aumentar as chances de sucesso de um título; e (2) desenvolver uma ferramenta de apoio ao processo.

1.3 Metodologia

Um conjunto de atividades foi executado para alcançar os objetivos descritos na seção 1.2. Essas atividades são descritas na Tabela 1.1.

Tabela 1.1: Fases da pesquisa

Fases	Descrição	Metodologia
Fase 1	Obter conhecimentos necessários sobre o tema pesquisado	Estudar a literatura existente sobre game design e conceituação;
		Estudar a literatura existente sobre processos de inovação;
		Estudar o processo utilizado pelas Manifesto Game Studio e BigHut Games.
Fase 2	Formalizar e evoluir o processo	Definir as principais atividades existentes do processo original;
		Estudar formas de evoluir o processo.
Fase 3	Projeto e desenvolvimento de um protótipo	Estudar formas de automatizar o processo;
		Desenvolver um protótipo da ferramenta de apoio.
Fase 4	Definir e executar métodos de avaliação	Definir métricas de avaliação;
		Definir um protocolo experimental;
		Executar experimentos;
		Analisar os dados obtidos
Fase 5	Conclusão do trabalho	Apresentar as conclusões do trabalho;
		Indicar as dificuldades e limitações da pesquisa;
		Publicar o trabalho através de artigos científicos

A fase inicial consistiu em agregar conhecimento sobre processos de *game design*, em especial no estágio de conceituação, e estudar o processo aplicado pelas duas empresas citadas anteriormente. A segunda fase objetivou estruturar o processo em

atividades, e procurar formas de evoluí-lo. Em seguida, essa estrutura foi analisada de modo a encontrar formas de automatizar o processo, e foi desenvolvida uma aplicação para auxiliar os *designers* durante a conceituação. Fase 4 foi responsável pela avaliação da abordagem, e a fase 5 trata da conclusão e documentação do trabalho.

1.4 Estrutura da dissertação

Essa dissertação está dividida em 8 capítulos, descritos abaixo:

Capítulo 2 apresenta os conceitos gerais sobre o processo de *game design*, em especial sobre sua fase inicial, a conceituação. Ele também apresenta os principais processos utilizados atualmente na indústria, e quais são suas limitações.

Capítulo 3 apresenta uma breve discussão sobre inovação em jogos e as principais estratégias de inovação utilizadas atualmente na indústria, fora do mercado de jogos.

No Capítulo 4 é relatada a experiência das empresas Manifesto Games e BigHut Games ao utilizar uma estratégia de inovação no processo de conceituação de jogos. Também são discutidos os principais resultados obtidos por eles no mercado, e as limitações descobertas durante esses experimentos.

O Capítulo 5 consiste na descrição da evolução, estruturação e formalização do processo iniciado pelas empresas acima.

Capítulo 6 descreve as especificações e o desenvolvimento de uma ferramenta de apoio ao processo descrito do capítulo anterior.

No Capítulo 7 encontram-se os experimentos e resultados para avaliação do processo e da ferramenta propostos.

Finalmente, o Capítulo 8 apresenta as principais considerações desse trabalho e os possíveis passos futuros.

2. *Game Concept* e Inovação

Essa seção objetiva apresentar alguns conceitos teóricos presentes nesse trabalho, em especial *game design* e conceituação, e os principais processos e desafios relacionados à conceituação.

2.1 Game Design

O recente amadurecimento da indústria de jogos implicou em menores margens para erros e riscos maiores. Isso resultou em empresas buscando constantemente maneiras de melhorar e desenvolver seus processos criativos, tornando-os menos artísticos e mais eficientes (CREDIDIO, 2007; TSCHANG, 2003).

Zimmerman (2003) argumenta que *game design* é um “*design* de segunda-ordem”, pois o projeto do jogo é feito de forma indireta. Ou seja, o sistema em si é projetável, mas a experiência do jogo não (KULTIMA; KÖÖNIKKÄ; KARVINEN, 2011). Em contrapartida, Rouse III (2004) define *game design* como o que determina as possíveis escolhas que os jogadores terão no jogo, e como essas escolhas influenciam no restante do jogo. Ele envolve aspectos como nível de dificuldade, comunicação com o ambiente, formas de controle e interação, critérios de vitória ou derrota. Apenas entendendo esses conceitos fundamentais é possível inovar (SALEN; ZIMMERMAN, 2014).

A área de *design* de jogos é diversa e variada, e muitos esforços foram feitos ao longo dos anos para evoluí-la (CRAWFORD, 1982; LAUREL, 2003; LUBAN, 2001; ZIMMERMAN, 2003). Zimmerman (2003) propôs o “*design* iterativo”, um processo cíclico que consiste em repetir três etapas: prototipação, análise e refinamento. Nele, um protótipo é evoluído iterativamente até ele alcançar versões avançadas. Assim, a iteração funciona como uma pesquisa evolutiva em busca do resultado ideal. O processo depende fortemente de testes, que permitem ajudar o *design* a melhor corresponder ao público-alvo. A Figura 2.1 demonstra esse processo.

Todavia, Adams (2010) argumenta que a solução do *design* iterativo não é viável em jogos médios ou grandes. Ele propôs um processo centrado no usuário, também dividido em três fases: concepção, elaboração e refinamento; como mostra a Figura 2.2.

Nesse processo, escolhas iniciais, como conceito principal, público-alvo e gênero do jogo, devem ser decididas no estágio de concepção, e não podem ser alterados posteriormente. A fase seguinte adiciona detalhes ao conceito inicial, gerando e testando protótipos. Uma vez satisfeitos, a fase de refinamento permite fazer pequenos ajustes, mas novas funcionalidades não podem ser inseridas.

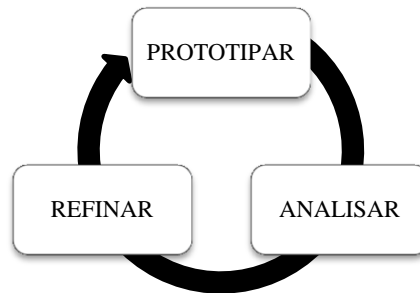


Figura 2.1: *Design iterativo*, proposto por Zimmerman (2003).



Figura 2.2: Processo proposto por Adams (2010).

Araújo e Roque (2009) descrevem um modelo de *design* dividido em conceituação, *design* e desenvolvimento; como mostra a Figura 2.3. A etapa de conceituação visa definir o conceito inicial do jogo, definindo os requisitos do jogo, as ideias iniciais, tema principal e público alvo. Na fase seguinte, esse conceito é transformado em um sistema de regras. Finalmente, a fase de desenvolvimento aprimora e avalia protótipos de jogo até que um produto satisfatório seja gerado. Caso haja a necessidade, é possível avançar ou regredir no processo quantas vezes forem necessárias.



Figura 2.3: Modelo geral proposto por Araújo e Roque (2009).

Credidio (2007), por sua vez, propôs um método em quatro passos. O primeiro consistia na exploração do problema de *design*, onde eram definidas as necessidades e

os requisitos do jogo. Em seguida ocorria a geração de alternativas, onde várias ideias eram criadas. O passo 3 analisava e selecionava essas ideias, e no passo 4 uma solução era apresentada como um documento de *design*. Para cada passo, ele sugeriu uma série de técnicas que podem ser aplicadas, originadas de outros processos de *design*.

O processo proposto por Luban (2001) consiste em quatro etapas. Na primeira, são definidos os objetivos, e times são organizados, com foco em selecionar pessoas que trabalharão nos próximos passos. Idealmente, essas pessoas serão de diferentes áreas e com diferentes expertises. Em seguida, são identificados parâmetros e seus valores. Luban define parâmetros como características independentes que servirão para descrever o jogo, como “segmento de mercado” ou “ambientação temporal”. Um exemplo de valor para “segmento de mercado” poderia ser “FPS”, e para “ambientação temporal” poderia ser “contemporâneo”.

O terceiro passo requer a filtragem de ideias e a definição de preferências. Como pode existir uma grande quantidade de combinações possíveis entre parâmetros e valores, é necessário restringir o escopo do jogo. Isso pode ser feito removendo conjuntos de ideias que tenham valores conflitantes. Por exemplo, uma pessoa pode argumentar que, do ponto de vista de mercado, um jogo de corrida sub-aquático pode não fazer sentido, e essa ideia seria rejeitada. Finalmente, o último passo consiste em analisar e selecionar as ideias de acordo as prioridades definidas no passo 3.

Embora existam divergências entre os processos descritos acima, é possível notar todos – exceto o de Luban, pois esse é focado apenas na primeira etapa do design – possuem pelo menos três estágios maiores: conceituação, refinamento e desenvolvimento. Dessas três fases, a que recebeu menor atenção foi a inicial. Os processos descritos por Zimmerman, Adams e Araújo e Roque são muito gerais e não apresentam detalhes específicos sobre como proceder na conceituação. Em contrapartida, Luban e Credidio deram um foco maior a essa etapa. Além disso, esses processos são muito dependentes da imparcialidade dos *designers*, pois muitas vezes eles desejam fazer seu “jogo ideal”, o que reprime ideias que não atendam a seus gostos.

Luban apresenta um processo focado primariamente na conceituação, e o uso de parâmetros e valores pode aumentar o exercício criativo. Entretanto, informações sobre o usuário e o mercado não são incorporadas no processo de forma clara. O modelo de Credidio, entretanto, tenta prover diretrizes gerais, mas fornece um conjunto de

possíveis técnicas absorvidas de outros processos como solução. Embora o conjunto de soluções e o processo sejam uma contribuição significativa, seria interessante avaliar como e quais dessas técnicas são melhores para gerar propósitos de inovação ou de mercado. Além disso, diferentemente dos outros autores, seu trabalho apresentou apenas resultados preliminares, não tendo sido testado num ambiente real da indústria de jogos.

2.2 Conceituação

A habilidade de gerar ideias é uma das características de empreendimentos bem sucedidos (GABLER et al., 2005). Em um mercado competitivo como o de jogos digitais, essa habilidade é essencial. Principalmente em sub-mercados, como o *mobile*, onde a ascensão de modelos de negócio como *Game as Service* tornou necessário fornecer constantemente novos conteúdos aos jogadores. Entretanto, a geração de ideias ainda é uma das áreas mais negligenciadas, principalmente pela academia, no mundo dos jogos digitais (PAAVILAINEN et al., 2009).

Essa habilidade mostra-se especialmente importante durante a conceituação, o estágio inicial do processo de design, onde se cria um conceito do que virá a ser o jogo. Esse conceito é a coleção de todas as ideias importantes que definem o produto, e é dividido em duas partes: uma parte reciclada (ideias que foram usadas e outros jogos, filmes ou livros, e.g. “o jogo será um plataforma”), e a parte inovadora (ideias que ainda não foram utilizadas) (HAGEN, 2009).

Entretanto, a criação do conceito do jogo ainda costuma ser feita sem métodos específicos. Como mostrado na seção anterior, processos de *game design* tendem a ignorar essa fase em função de outras etapas, resultando em lacunas nos processos criativos. Quando não a ignoram, mostram uma falta de interesse em incorporar informações do mercado. Ambas as situações são perigosas, pois aumentam as chances de falhas de *design* que podem afetar como o jogo será percebido pelo público e qual será seu posicionamento no mercado.

2.3 Considerações Finais

Esse capítulo apresentou os principais conceitos e trabalhos relacionados a *game design* e conceituação. No próximo capítulo, estratégias de negócio focadas em inovação serão discutidas.

3. Processos de Inovação

Inovar é um ponto chave na criação de um produto. O requisito básico para isso é, embora pareça redundante, a criação de algo que seja novo e tenha valor. Mas o que representa “novo”? Goldsmith e Foxall (2003) argumentam que “novidade” pode se referir a caráter recente, originalidade ou similaridade. Por caráter recente, entenda-se algo que foi encontrado ou adquirido recentemente. Originalidade já apresenta a ideia de desconhecimento ou estranheza. Por fim, similaridade indica o quão diferente algo é das coisas do mesmo tipo que já existem.

Além disso, inovação necessita de sucesso. Originalidade, similaridade ou caráter recente são irrelevantes quando o produto/serviço não é útil para ninguém. Portanto, inovação e novidade são distintas, pois novidade pode não ter valor.

Empresas pequenas e *startups* têm mais dificuldades em alcançar um espaço no mercado. Uma forma de diferenciar-se dos concorrentes é focar em inovação. Trabalhar com produtos que já existam no mercado requer produzir algo melhor ou mais barato, o que é muitas vezes inviável em pequenas empresas no início da carreira. Apostar em um mercado novo permite maiores chances de sucesso. Assim, surge uma relação entre empreendedorismo e inovação.

Estratégias de negócio podem representar fontes de inovação e vantagem competitiva no mercado (MORRIS; SCHINDEHUTTE; ALLEN, 2005). Estratégias muito inovadoras podem até mudar toda a estrutura da indústria. Um exemplo é a loja de músicas da *Apple*, *iTunes*, que começou como um serviço para promover um produto (*iPods*), e veio a se tornar um novo modelo de venda e distribuição.

Esse capítulo apresentará três processos de inovação utilizados atualmente na indústria. São eles: *Business Model Canvas*, *Lean Startup* e *Blue Ocean Strategy*.

3.1 Business Model Canvas (BMC)

O *Business Model Canvas*, ou Painel de Modelo de Negócios, é um modelo estratégico que objetiva auxiliar a criação de novos produtos ou serviços, de forma

rápida e visual (OSTERWALDER et al., 2010). Ele surgiu com o objetivo de reduzir o tempo perdido na elaboração de planos de negócio complicados e fictícios. O uso de um modelo de negócios com partes fixas, mas genérico, permite ao empreendedor visualizar melhor sua empresa, identificando áreas onde pode gerar melhorias e inovação.

O painel é composto por nove blocos, ou segmentos, conforme descritos abaixo:

1. **Bloco do Segmento do Consumidor:** Define os diferentes grupos de pessoas que a empresa visa atingir. Dentre as categorias possíveis, podemos citar mercado em massa ou mercado diversificado;
2. **Bloco da Proposta de Valores:** Descreve um conjunto de produtos que podem possuir valor para determinado segmento do consumidor. Podem ser completamente novos, ou variações de produtos existentes no mercado;
3. **Bloco dos Canais de Distribuição:** Como a empresa se comunica com os consumidores para entregar os produtos/serviços. Esses canais podem ser diretos ou indiretos, próprios ou terceirizados. Exemplos: entrega em domicílio, atendimento presencial, site de conteúdo, etc;
4. **Bloco dos Relacionamentos com os Consumidores:** Define os tipos de relacionamentos entre a empresa e um tipo específico de consumidor. Alguns exemplos são assistência pessoal, criação conjunta, SAC ou serviços automáticos;
5. **Bloco de Linha de Receita:** representa como a empresa pretende gerar receita a partir de cada bloco do consumidor. Venda direta, retorno em publicidade paga e aluguel são alguns exemplos;
6. **Bloco dos Recursos-Chave:** Define os recursos mais importantes para o bom funcionamento do modelo, como equipes, máquinas e investimentos;
7. **Bloco das Atividades-Chave:** Descreve as atividades que a empresa deve fazer para que o modelo funcione corretamente. Alguns exemplos são monitoramento de redes sociais ou construção de lojas;
8. **Bloco das Parcerias-Chave:** Redes de fornecedores e parceiros necessárias, como fornecedores de tecnologia ou recursos;

9. **Bloco da Estrutura de Custo:** Todos os custos envolvidos durante a operação do modelo, como manutenção de equipamentos ou pagamento de parceiros.

A Figura 3.1 apresenta uma representação visual dessa estratégia.



Figura 3.1: Modelo geral do Business Model Canvas.

3.2 Lean Startup

Eric Ries define uma *startup* como “uma instituição humana que objetiva entregar um produto ou serviço novo, em condições extremas de incerteza” (RIES, 2011). Por “incerteza extrema”, podemos entender um mercado em constante evolução. Assim, é um conceito de empreendedorismo focado no *feedback* contínuo, onde recomenda-se validar a ideia durante todo o processo, para verificar se existem compradores para o produto.

A ideia dos princípios *lean* é aumentar a eficiência da produção, reduzindo qualquer tipo de desperdício (de recursos humanos ou materiais, atividades inúteis ou custos) durante o processo. A aplicação desses princípios às *startups* deu origem ao *lean startup*. *Lean startup* é um método de inovação que sugere que a inovação mais importante é aquela criada onde existe uma demanda real. Ou seja, o maior desperdício surge com um produto ou serviço que ninguém precisa (MUELLER; THORING, 2012).

O processo *lean startup* consiste em três fases principais: construir, medir e aprender, conforme ilustra a Figura 3.2. Na fase inicial, “construir”, a empresa inicia a construção do produto ou serviço, gerando o que é chamado “*Minimum Viable*

Product” (MVP), ou “Produto Viável Mínimo”. O MVP é a versão de um produto que permite a maior quantidade possível de aprendizagem validada sobre clientes, criada com o mínimo esforço necessário. Com ela, é possível, na segunda fase, testar hipóteses e aprimorar o produto em uma menor quantidade de tempo. Nesse estágio, a empresa pode descobrir se está tendo progresso ou se está presa em métricas de vaidade, que são métricas que dão o resultado mais otimista possível (e.g. número de downloads ou usuários cadastrados) e podem ou não ter real importância para a empresa.

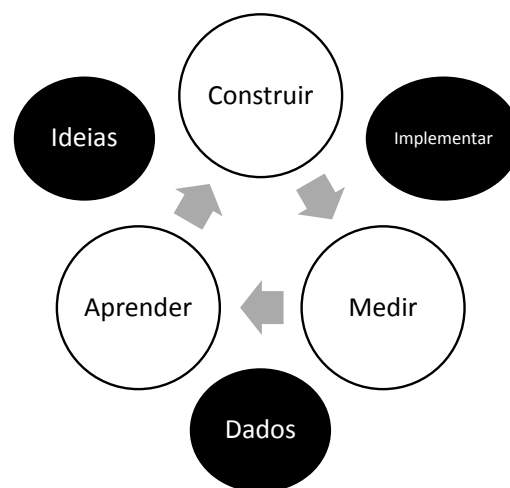


Figura 3.2: Representação da estratégia Lean Startup.

Em seguida inicia-se a fase de aprendizagem, onde, utilizando dados obtidos na fase anterior, são decididos os rumos do produto: continuação do mesmo ou criação de um “Pivô”, que consiste na alteração de um ponto primário do produto/serviço. A ideia do pivô parte de um princípio fundamental da *lean startup*, que diz que todo modelo de negócio, serviço e/ou produto vai falhar eventualmente, então é melhor falhar cedo, evitando investir mais e possibilitando aprender com os erros e corrigir futuros rumos.

3.3 Blue Ocean Strategy (BOS)

Kim e Mauborn (2005) criaram em 2005 uma abordagem de gerenciamento chamada *Blue Ocean Strategy* (BOS), ou Estratégia do Oceano Azul, onde se busca diminuir a concorrência através de novos espaços de mercado. Para tanto, eles propuseram os conceitos de oceano azul e vermelho. Oceanos vermelhos representam o espaço de mercado que já foi conquistado. Ou seja, onde empresas já buscaram lançar produtos ou serviços. Nele, o ambiente é competitivo e as fronteiras de mercado são bem definidas (KIM; MAUBORGNE, 2005). Conforme mais empresas buscam

ultrapassar suas rivais e o espaço torna-se cada vez mais escasso, os lucros e o crescimento tendem a diminuir. Assim, o “vermelho” serve como analogia para as empresas que não conseguem sobreviver nesse mercado hostil.

Oceanos azuis, por outro lado, representam todo o espaço que ainda não foi explorado (ABOUJAFARI et al., 2013; KIM; MAUBORGNE, 2005). Assim, cria-se demanda, tornando a concorrência irrelevante. Isso resulta em maiores chances de crescimento e lucro. Um oceano azul não precisa necessariamente ser algo novo, podendo também ser uma expansão de um oceano vermelho. Independente de como seja criado, ele gera uma mudança no ambiente de mercado e, posteriormente, na evolução do conteúdo. Assim, a cor azul representa inovação.

A Tabela 3.1 apresenta as principais diferenças entre os oceanos azul e vermelho. Kim e Mauborn argumentam que a principal diferença é a falta de uma restrição de escolhas entre valor e custo nos oceanos azuis. Em abordagens tradicionais, o custo do produto ou serviço é ditado pelo valor do mesmo. Uma empresa pode escolher oferecer um custo maior com um valor maior, ou um custo melhor com um valor menos. Oceanos azuis, entretanto, buscam tanto diversificação quanto baixo custo. A comparação dessas estratégias em relação ao valor do consumidor é ilustrada na Figura 3.3.

Tabela 3.1: Principais diferenças entre oceanos azuis e vermelhos

Oceano Azul	Oceano Vermelho
Criação de um espaço de mercado livre de competição	Competição sempre presente em mercados existentes
Competição é irrelevante	Necessário ultrapassar os concorrentes
Não existe substituição de valor e custo	Valor e custo dependem da demanda existente
Cria e captura novas fontes de demanda	Restrições entre valor e custo
Atividades da empresa voltadas em diferenciação e estratégias de baixo custo	Atividades da empresa voltadas em diferenciação ou estratégias de baixo custo

Outra inovação importante presente no BOS é a criação do conceito de *strategy canvas*, ou painel de estratégia. Seu objetivo é capturar as atividades atuais das empresas no mercado. Assim, é possível identificar onde a competição está investindo e quais produtos já estão disponíveis (KIM; MAUBORGNE, 2005). Esse painel é

representado por um gráfico com dois eixos, como mostra a Figura 3.4. O eixo horizontal mostra as características do produto. Em jogos, poderíamos ter preço, duração do jogo ou *replay value*, por exemplo. O eixo vertical quantifica os níveis dessas características. Por exemplo, no caso de *replay value*, um valor alto indicaria um *replay value* grande.

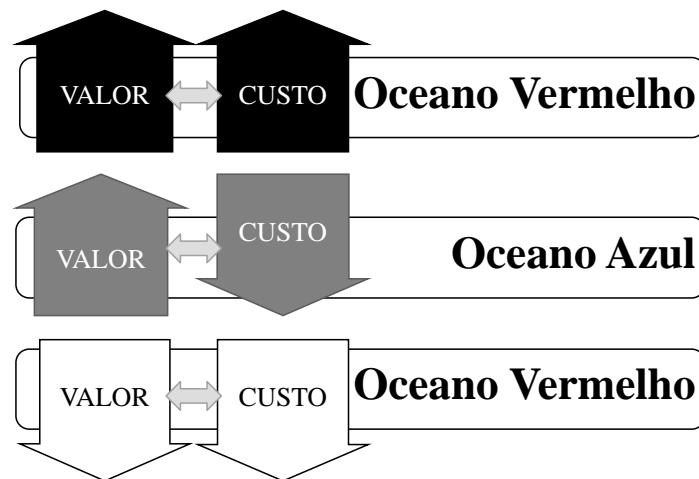


Figura 3.3: Relação valor/custo em oceanos azuis e vermelhos.

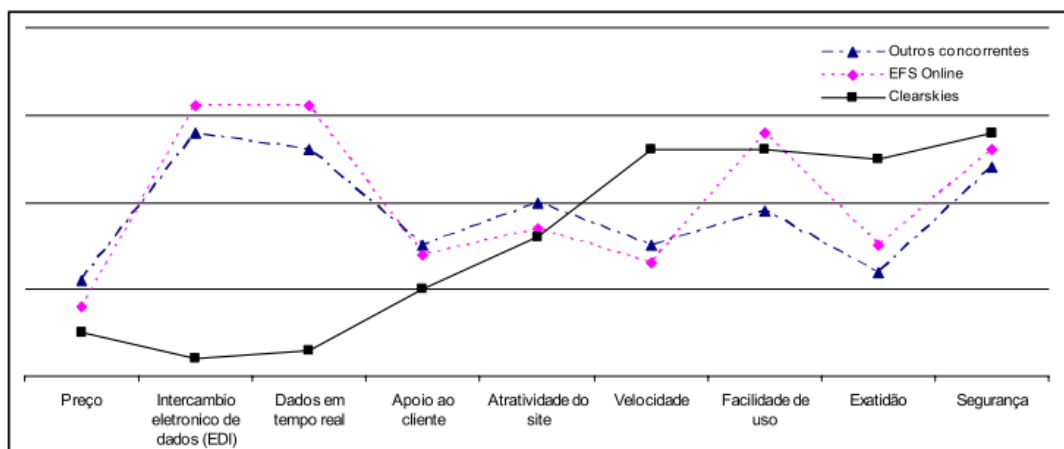


Figura 3.4: Exemplo de um painel de estratégia para empresas cambiais, com 3 curvas de valor (“outros concorrentes”, “EFS Online” e “Clearskies”). Fonte: (KIM; MAUBORGNE, 2005)

Os dados das empresas são adicionados ao painel, gerando a curva de valor (do inglês, *value curve*) da empresa. Essa curva representa seu desempenho no espaço de mercado. Em posse de tais curvas, deve-se criar novas curvas de valor, aumentando, diminuindo, excluindo ou adicionando níveis e características.

É importante ressaltar que curva de valor e painel de estratégia não representam a mesma coisa. Painel de estratégia é o ambiente onde uma ou mais curvas de valor serão

representadas, ou seja, engloba tanto as curvas, como os eixos horizontal e vertical. A curva de valor é um conjunto de pares <característica, valor>, que são pintados em um painel de estratégia. Figura 3.4 demonstra essa diferença.

3.4 Considerações Finais

Esse capítulo descreveu três estratégias de inovação comumente usadas no mercado atual. Nesse trabalho, utilizaremos princípios da estratégia do oceano azul para buscar inovação na fase de conceituação do *game design*. O próximo capítulo apresenta a experiência de duas empresas locais em unir essa estratégia e a conceituação.

4. Aplicando BOS à Conceituação de Jogos Digitais

Manifesto Games é uma empresa local criada em 2005, com o objetivo de criar jogos casuais. Em 2012, ela criou uma parceria com a C2Flag Studio, gerando a BigHut Games, que viria a ser responsável por grande parte dos empreendimentos *mobile* da companhia. Essa seção apresentará a experiência dessas empresas aplicando BOS ao processo de criação de jogos.

4.1 Experiência da BigHut Games e Manifesto Games

Os mercados atuais possuem uma grande quantidade de títulos disponíveis. O mercado de dispositivos móveis conta com 194.512 jogos apenas na Apple Store⁸, onde 138 novos jogos são submetidos por dia. Com o objetivo de gerar produtos com diferencial de mercado, as empresas Manifesto Games e BigHut Games decidiram aplicar os conceitos do BOS ao método de *design* de jogos, especificamente na fase de conceituação. Essa atividade foi feita de forma pragmática durante a criação de dois jogos. O processo foi adaptado da seguinte maneira: inicialmente, foram selecionados dois nichos de mercado, que foram representados por Personas. Uma persona é um usuário imaginário que representa o arquétipo de potenciais consumidores reais (COOPER, 2004). Ele possui nome, idade, gostos e desgostos, história pessoal, entre outros. Esse usuário hipotético é tratado pelos *designers* como uma pessoa de verdade. Assim, as funcionalidades do produto devem atender às necessidades e preferências da persona.

Foram criadas duas personas, uma primária e uma secundária. Embora diferentes, como o objetivo do jogo é agradar completamente a primária, ao mesmo tempo em que não incomode a secundária, os jogos escolhidos podem acabar sendo parecidos. Uma vez escolhidas as personas, um jogo de um terceiro mercado era selecionado, de modo a

⁸ Dados de <http://148apps.biz/app-store-metrics/>, atualizado em 17 de fevereiro de 2014, última visita em 19 de fevereiro de 2014.

fornecer um olhar diferente sobre os demais produtos. Assim, os jogos dos mercados representados eram selecionados e avaliados, a fim de definir um conjunto de características capazes de descrevê-los.

Essas características eram, então, colocadas no eixo horizontal de um painel de estratégia, como descrito na seção 3.3. Em seguida, os dados dos jogos existentes eram adicionados ao gráfico na forma de uma curva de valor. Esse gráfico era copiado e a equipe participando no processo era dividida em dois grupos, que eram separados em duas salas. Cada grupo buscava criar uma nova curva de valor a partir das existentes, usando as quatro operações do BOS (aumentar, diminuir, excluir e adicionar). Após determinado tempo, cada curva gerada era entregue ao grupo oposto, que deveria propor jogos que pudessem ser representados por aquela curva. A Figura 4.1 apresenta um resumo das fases desse processo.

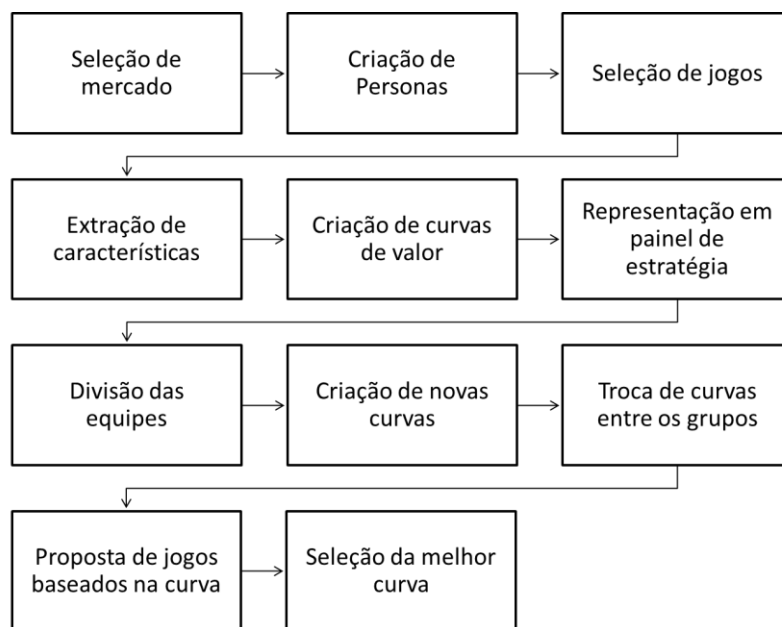


Figura 4.1: Resumo do processo da Manifesto Games e BigHut Games. A partir da terceira fileira (após o passo Divisão das equipes), as atividades são feitas em duplicado por cada time, salvo a última etapa (Seleção da melhor curva).

Utilizando esse processo, as empresas criaram e lançaram dois produtos, que serão descritos a seguir.

4.1.1 Caso de Uso nº 1: Fruits Inc.

4.1.1.1 Sobre o desenvolvimento do conceito inicial

Os mercados primário, secundário e terciário escolhidos para o primeiro caso de uso foram, respectivamente, *time management* (TM), *puzzles* e *hidden objects* (HO). TM é um gênero focado em gerenciar o tempo gasto em várias tarefas de modo a obter máxima eficiência. *Puzzles* são jogos de desafios ou enigmas de lógica, matemática, estratégia, reconhecimento de padrões ou solução de problemas no geral. Finalmente, HO é um sub-gênero de jogos de *puzzles* onde o jogador deve encontrar itens escondidos numa foto ou ambiente. A escolha desses gêneros foi influenciada pela necessidade de aliar oportunidades de mercado com as habilidades da equipe. Jogos do tipo TM possuem menos atributos de arte que HO, e são mais fáceis de refinar, do ponto de vista de *design*, que *puzzles*. Além disso, na época do desenvolvimento, todos os jogos eram muito populares com jogadores casuais.

Utilizando mineração de dados em redes sociais para aprender informações sobre consumidores em potencial, foram criadas duas personas:

- **Persona primária:** *Diane, 42 anos, arquiteta, solteira. Vive em Detroit, Michigan. Tem um filho, Jonathan (22). Possui pouco tempo livre e ocasionalmente assiste filmes de comédia-romântica. Seu programa de televisão favorito é “Friends”, mas ela também aprecia outros programas de comédia ou drama. Diane ouve rock e lê livros de suspense e ficção. Ela é uma mulher ocupada, que joga videogames para aliviar o estresse.*
- **Persona secundária:** *Amanda, 28 anos, dona-de-casa, casada. Vive com o marido em San Diego, California, e tem dois filhos: Elizabeth (4) e Sean (3). Entre seus filmes de ação favoritos estão “Piratas do Caribe” e “Onze Homens e um Segredo”. Ela gosta de assistir séries de investigação e reality shows. Ela também escuta música pop e contry, e lê livros de aventura.*

Os jogos escolhidos para cada gênero foram:

- **Time Management:** *My Kingdom for the Princess II, Royal Envoy, The fifth gate, Hotel Dash, Emily’s Taste of Game.*

- ***Hidden Objects:*** *Mystery Case Files, Robinson Crusoe, Haunted Manor, and Devil's Triangle.*
- ***Puzzles:*** *Zuma's Deluxe, Chuzzle Deluxe, Bejeweled 2 Deluxe, Jewel Quest.*

Cada um desses jogos foi analisado separadamente. Dois tipos de análise foram realizados: por especialistas e por meio de críticas de jogadores. A análise por especialistas foi feita por alguns membros do time, e se tratava de uma avaliação mais técnica. Seu propósito era avaliar aspectos como gráficos, som, jogabilidade, longevidade e progressão do jogo. Exemplos de comentários feitos foram “Algumas telas/*popups* realmente precisam de melhores estéticas” (gráficos) e “Existe uma falta de *mini-games* ou níveis bônus” (jogabilidade).

As críticas dos jogadores foram obtidas em sites. Elas ajudaram os designers a entender como os jogadores perceberam as funcionalidades dos jogos e o que eles consideraram bom ou ruim. Exemplos de comentários foram “Eu gosto do fato que temos que rever todas as opções rapidamente para escolher a melhor ação” ou “Eu provavelmente compraria esse jogo se não fossem pelos *mini-games*”.

Com essas análises foi possível identificar um conjunto de atributos para cada gênero, como mostrados abaixo:

- ***Time Management:*** Objetivo de longo prazo, melhoria gradual, item de propósito único, personagens de suporte, customização, apertar repetidamente um botão.
- ***Hidden Objects:*** *Mini-Puzzles*, opções de ajuda, progresso muito lento, aventura.
- ***Puzzles:*** Chefões, *power-ups*, combos escaláveis, coordenação entre olhos e mãos.

Cada jogo foi representado usando esses atributos numa curva de valor, e um painel de estratégia foi criado. A Figura 4.2 apresenta o exemplo de duas curvas de valor que representam os jogos *Hotel Dash* e *My Kingdom for the Princess II*.

Após a representação dos jogos já existentes, foram criadas novas curvas de valor. Essas curvas tinham como foco os atributos do mercado primário (TM). Então, alguns desses atributos foram removidos ou diminuídos, e atributos dos outros mercados foram

adicionados. Algumas das curvas de valor criadas são mostradas na Figura 4.3. Essas curvas foram avaliadas e ranqueadas pela equipe. Foram verificados os custos em função dos benefícios de cada proposta, e, finalmente, a curva melhor ranqueada foi selecionada e desenvolvida até resultar em um produto final: *Fruits Inc.*

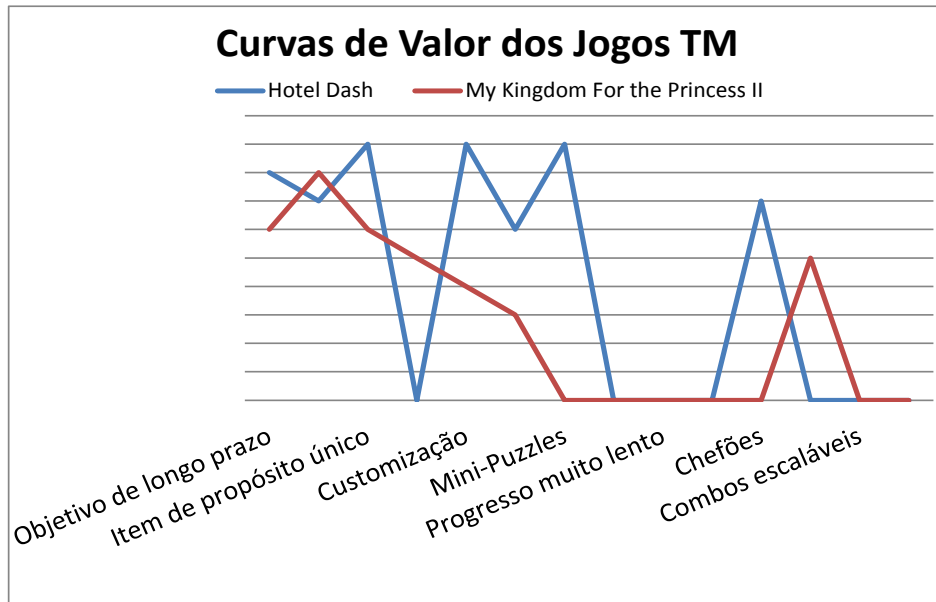


Figura 4.2: Curvas de valor que representam os jogos Hotel Dash e My Kingdom for the Princess II.

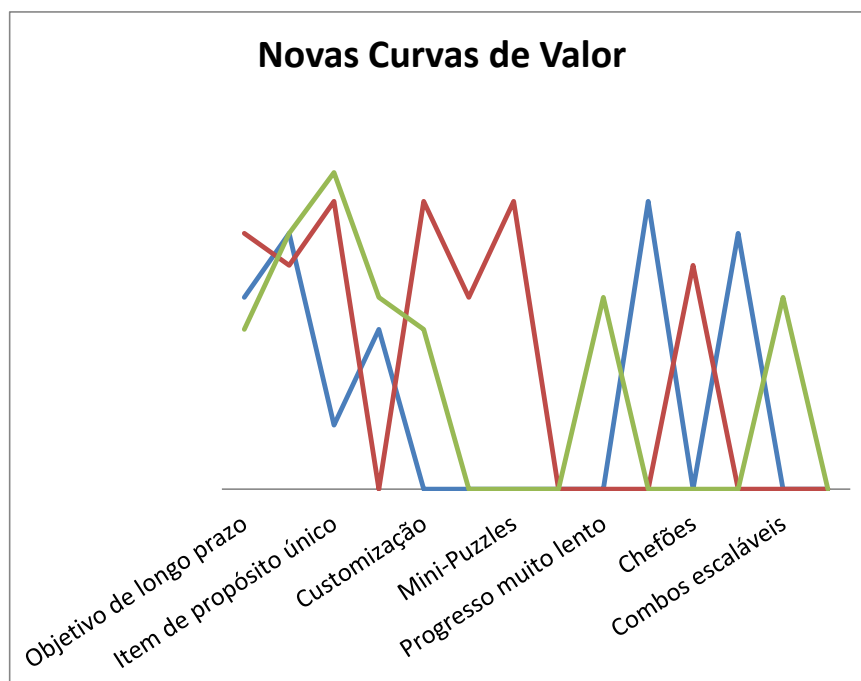


Figura 4.3: Exemplo de novas curvas de valor geradas.

4.1.1.2 Sobre o produto final e seus resultados

Fruits Inc. é um jogo de *time management*, com algumas características dos gêneros HO e puzzle. Desde seu lançamento no mercado, onde está disponível para PC, Mac e ainda em uma versão online para navegadores, ele já foi traduzido para 11 línguas. Seus *downloads* nas versões PC e Mac ultrapassam 3.2 milhões, e já atingiu mais de 15 milhões de sessões de jogos na versão para navegador. Figura 4.4 mostra o menu inicial e uma tela do jogo.

Embora tenha recebido algumas críticas com relação ao ritmo ou balanceamento, muitos usuários consideraram a experiência positiva, e muitos elogiaram alguns aspectos como ambientação nova (*“Fresh setting and challenging game mechanics meet problematic pace and balance”*⁹), ou “ambientação nova e mecânicas de jogo desafiadoras encontram ritmo e balanço problemáticos”, em tradução livre) ou originalidade (*“It’s overall a more original game than most”*¹⁰), ou “em geral é um jogo mais original que a maioria”, em tradução livre). Sua nota no *GameZebo*¹¹ foi 4,5 / 5,0.



Figura 4.4: Telas do Fruits Inc. A esquerda, o menu inicial. À direita, uma tela do jogo.

4.1.2 Caso de Uso nº 2: Boney The Runner

4.1.2.1 Sobre o desenvolvimento do conceito inicial

Em contraste com o mercado-alvo do primeiro caso de uso (PC/Mac), para o segundo caso foi escolhido desenvolver um jogo para dispositivos móveis. Nesse caso,

⁹ <http://www.gamezebo.com/games/fruits-inc/review>

¹⁰ <http://forums.bigfishgames.com/posts/list/80/159123.page>

¹¹ <http://www.gamezebo.com/games/fruits-inc/review>

os mercados primário, secundário e terciário foram, respectivamente, *endless runners (ER)*, *endless jumpers (EJ)* e *blitz puzzles (BP)*.

O gênero ER consiste numa ideia simples: o jogador deve correr até que seu personagem seja destruído. Isso significa que eles têm o potencial de resultar em partidas infinitas. Na prática isso não ocorre, pois eles tendem a ser feitos de modo a ficarem mais difíceis com o passar do tempo. EJ são parecidos com ER, com uma diferença: no ER o eixo é horizontal, mas nos EJs o personagem deve pular, tentando sempre alcançar uma distância vertical maior. Finalmente, BP são jogos de *puzzle* onde se combina itens rapidamente, de modo a fazê-los desaparecer. Por exemplo, BPs podem ser sobre deslizar pedras coloridas, de modo a juntar três ou mais, o que as faz desaparecer.

As personas foram criadas utilizando dados de um jogo lançado anteriormente pela BigHut Games, chamado *Dino Jump*. Esses dados incluíam informação como idade, gênero e tempo gasto no jogo. As personas criadas foram:

- **Persona primária:** *Brandom, 28 anos, solteiro, advogado. Sua renda anual é US\$ 100.000,00, permitindo que tenha uma vida bem confortável. Ele vive com sua namorada, Joan. Está sempre conectado, nunca deixando seu smartphone, que ele usa principalmente para trabalhar e, às vezes, para divertir-se. Ele gosta de jogos casuais, pois não ocupam muito do seu tempo.*
- **Persona secundária:** *Elle, 23 anos, universitária, solteira. Ela vive na Inglaterra e acabou de deixar a casa dos pais. Vive com três amigos, todos universitários. Sua renda atual é US\$ 80.000,00. Gosta de filmes de comédia e passa grande parte do seu tempo com amigos ou em redes sociais.*

Assim como no caso anterior, foram selecionados jogos para cada mercado, conforme descritos abaixo:

- **Endless Runner:** *Jetpack Joyride, Run like hell, Temple Run.*
- **Endless Jumper:** *Doodle Jump, Mega Jump, Dino Jump.*
- **Blitz Puzzles:** *Bejeweled Blitz, Zuma Blitz.*

E os atributos extraídos dos jogos foram:

- **Endless Runner:** Mecânicas de jogo variáveis, *power-ups*, *upgrades* de ferramentas, moedas, linha de morte.
- **Endless Jumper:** *Upgrades*, moedas, *upgrades* coletáveis, linha de morte.
- **Blitz Puzzles:** Economia consumível, jogabilidade com tempo.

O restante do processo foi similar ao caso anterior.

4.1.2.2 Sobre o produto final e seus resultados

Boney the Runner foi o resultado do segundo caso de estudo. É um jogo *mobile*, lançado nos mercados iOS e Android. O jogo é do tipo *endless runner*, onde o personagem está sempre correndo em direção à direita, e o jogador deve tocar a tela para fazer Boney (o esqueleto protagonista) pular, evitando obstáculos e pegando moedas. O jogo termina quando Boney é alcançado por um grupo de cachorros. A Figura 4.5 apresenta a tela inicial e uma *screenshot* do jogo.

O jogo difere de outros ERs, pois possui características de jogos de azar, características de jogos de combinação e funcionalidades consumíveis. É possível obter itens durante as sessões do jogo ou comprando-os *on-line*. Esses itens podem ser combinados (Figura 4.6), gerando *power-ups* que podem ser usados dentro do jogo, alterando a jogabilidade (e.g. fazendo Boney correr mais rápido, ou mudando o tipo de moeda para um valor mais alto). Já os atributos de jogos de azar aparecem na forma de um caça-níquel (Figura 4.7) que pode ser utilizada para pegar itens aleatórios, ou item nenhum caso o jogador não tenha sorte.



Figura 4.5: Telas do Boney the Runner. À esquerda, tela de abertura. À direita, uma tela do jogo.



Figura 4.6: Tela de combinação de itens.

Boney the Runner possui mais de 300 mil downloads, acima de 11 mil jogadores diários ativos (*daily active users*, ou DAU) e mais de 110 mil usuários mensais ativos (*monthly active users*, ou MAU). DAU e MAU são a quantidade de jogadores que interagem com o jogo em um dado dia ou mês, respectivamente. A média de uma sessão de jogo é 12.5 minutos por sessão. Além disso, o jogo possuiu críticas positivas: 4,6 / 5,0 no Google Play¹² (mais de 2.332 usuários deram nota 5 / 5) e 4,5 / 5,0 no iTunes¹³.



Figura 4.7: Tela do caça-níquel de itens.

4.2 Limitações e Falhas

Com base no processo e na experiência descritos acima, essa seção apresentará as principais limitações e falhas encontradas no processo.

¹² https://play.google.com/store/apps/details?id=com.mobage.ww.a1088.Boney_The_Runner_BH_Android&hl=pt_BR

¹³ <https://itunes.apple.com/us/app/boney-the-runner/id573242168?mt=8&ign-mpt=uo%3D2>

4.2.1 Características muito dependentes do estilo

Como as características eram selecionadas em função de cada estilo, a reutilização dos jogos ou a comparação deles com outros se tornava complicada. Por exemplo, um jogo pode ter uma característica “Itens transformam-se em personagens”, que por coincidência está presente em outros jogos selecionados em determinado estilo. Todavia, um jogo pode fazer parte de vários estilos. Se em outra iteração do processo, esse mesmo jogo for selecionado em outro estilo, essa característica pode não ser utilizada, e a análise desse jogo terá de ser feita novamente.

Era preciso, portanto, encontrar descritores de jogos que fossem mais abstratos ou gerais, de forma a poder comparar diferentes estilos de jogos e reutilizar informações.

4.2.2 Geração trabalhosa de personas

Personas são ferramentas úteis na criação de um produto, pois fornecem ao designer uma visão próxima do jogador. Porém, sua criação, por ser detalhada, mostra-se custosa. Muitos dados, embora úteis, não são essenciais ao processo. Por exemplo, embora seja benéfico saber alguns títulos de filmes e livros que a persona gostaria, durante o processo proposto essa informação não é realmente essencial. Assim, diminuir o tempo gasto conjecturando e buscando tais filmes e livros, pode ser interessante no fim.

4.2.3 Tendências dos designers e diversidade das curvas

Analisando as fases de seleção de características e criação de curvas originais, é possível notar que o resultado final é muito dependente dos participantes. Diferentes pessoas podem argumentar que um jogo tem características diferentes, ou dar notas diferentes para a mesma característica em um mesmo jogo. Mais complicado, entretanto, é a geração de novas curvas, pois os participantes tendem a: (1) querer fazer um jogo que eles gostariam, ou (2) propor ideias dentro de um escopo limitado.

Quando um participante faz uma proposta para um jogo, ele deve ser capaz de ignorar seus gostos pessoais, em prol do que o usuário realmente gostaria. No mundo real, isso é uma tarefa complicada de ser realizada. Além disso, muitas pessoas são limitadas a pensar somente no que conhecem ou que estão acostumadas. Isso resulta em ideias que não fogem de certo padrão. Na fase de conceituação de um jogo, isso é considerado ruim, pois limita a quantidade possível de jogos propostos e a diversidade

entre eles. “Pensar fora da caixa” pode resultar em possibilidades que venham a ser mais interessantes e diferentes.

4.3 Considerações Finais

As empresas locais Manifesto Studio e BigHut Games buscaram meios de inovar em seus jogos. Com este fim, elas aplicaram os conceitos do BOS ao processo de conceituação de dois jogos distintos. Esses jogos foram criados com públicos-alvo diferentes e para plataformas variadas. Ambos os jogos foram bem sucedidos no mercado, com uma grande quantidade de usuários e críticas positivas. Dentre as críticas, pudemos observar que existiam elogios a funcionalidades que não pertenciam aos estilos principais dos jogos, e surgiram mediante o uso do BOS. Isso sugere que o uso dessa estratégia tenha tido papel fundamental do sucesso dos títulos.

Entretanto, nesse estágio o processo utilizado pelas empresas ainda encontrava-se muito pragmático e pouco estruturado. Outros problemas encontrados foram o tempo gasto na criação das curvas, na seleção dos mercados e criação das personas, nas ideias tendenciosas dos participantes, na pouca variabilidade de soluções e, principalmente, na dificuldade do reaproveitamento de material em futuras aplicações do processo. A próxima seção apresentará uma estruturação e formalização do processo e uma tentativa de prover melhorias ao mesmo, através de uma iteração no processo e uma ferramenta de apoio.

5. BOSS: *Blue Ocean Strategy System*

Esse capítulo apresenta o *Blue Ocean Strategy System* (BOSS), que engloba o processo evoluído pela aplicação da estratégia do oceano azul à fase de conceituação. Inicialmente será explicado o método seguido para estruturação e formalização do processo, a partir da experiência descrita anteriormente, na Seção 0. Em seguida, na Seção 5.3, serão discutidos os principais problemas existentes na versão inicial do processo, e que solução foi proposta para os mesmos. Finalmente, a Seção 5.3 apresentará o processo estruturado e a Seção 5.4 mostrará as representações da formalização do mesmo.

5.1 Metodologia

Para desenvolver BOSS, a seguinte metodologia foi seguida, dividida em quatro fases:

5.1.1 *Estudo da experiência da Manifesto Games e BigHut Games*

Inicialmente, o método utilizado pelas empresas foi estudado e avaliado. O objetivo dessa atividade era entender como a BOS podia ser aplicada à conceituação, quais mudanças seriam necessárias, e como ela podia auxiliar essa atividade. O estudo ocorreu a partir de entrevistas com membros da equipe e documentos de *design* remanescentes dos casos de uso.

5.1.2 *Estudo da literatura de game design e inovação*

A segunda etapa consistiu em um estudo dos processos existentes de *game design* e das estratégias de inovação presentes no mercado. O conhecimento dessas áreas era essencial para definição do problema atacado e análise dos problemas comuns encontrados.

5.1.3 *Estruturação, formulação e evolução do processo BOSS*

Em posse dos dados coletados nas entrevistas e nos documentos de *game design* no passo 5.1.1, foram levantados os principais pontos comuns ao método e um processo

com seis etapas foi proposto. Em seguida, as principais dificuldades encontradas nessa metodologia foram avaliadas e soluções foram sugeridas.

5.1.4 *Estudo e desenvolvimento de uma ferramenta de apoio*

A última fase consistiu no levantamento dos principais pontos de melhoria que poderiam ser resolvidos, parcial ou totalmente, por meio de ferramentas de apoio. Uma análise desses pontos, em função de custo e benefício, identificou aqueles que poderiam ser resolvidos. Uma ferramenta de apoio foi proposta e desenvolvida, e uma avaliação de teste foi feita com a equipe da Manifesto Games.

5.2 Evolução do Processo

A experiência da Manifesto Games e BigHut Games mostrou que a aplicação da BOS à fase de conceituação é relevante, no sentido comercial. Seus dois casos de teste foram bem sucedidos, em diferentes plataformas e mercados, o que indica que o processo funciona, ao menos naquelas circunstâncias. Entretanto, o método ainda possui algumas falhas, como mostrado na Seção 4.2. Abaixo seguem as mudanças realizadas no processo, de modo a diminuir ou resolver uma parte dos problemas citados:

5.2.1 *Lentes*

Para resolver o problema da dependência de estilo dos jogos na escolha de seus descritores, conforme descrito na seção 4.2.1, adotamos o conceito de lentes, proposto por Jesse Schell (2008). As lentes são formadas por algumas perguntas que devem ser feitas durante o processo de *design*, sobre o *design* criado, a fim de avaliá-lo. Em outras palavras, elas funcionam como perspectivas sobre determinado conceito, onde cada lente é uma forma diferente de enxerga-lo.

Em seu livro, Schell descreve 100 lentes diferentes. Embora elas não sejam perfeitas, e sua lista esteja incompleta (novas lentes podem ser criadas), elas são uteis como fonte de perspectiva fixa do *design* (SCHELL, 2008). Por exemplo, a lente número 15, chamada Lente do Brinquedo, é formada pelas seguintes perguntas:

- *O jogo ainda seria divertido se não tivesse objetivo? Se não, como podemos mudar isso?*
- *Quando alguém encontra esse jogo, ela deseja interagir com ele, antes mesmo de saber o que ele faz ou como funciona?*

Em outras palavras, essa lente serve como indicador sobre se a experiência principal do jogo, ignorando os aspectos funcionais da jogabilidade, é agradável ao jogador. Por se tratar de uma perspectiva, e, portanto, ser subjetiva à pessoa que a utiliza, é possível utilizá-la para descrever jogos variados. Por exemplo, um jogo de bichinho virtual, tal como o Tamagotchi¹⁴, pode ser considerado forte do ponto de vista da Lente do Brinquedo, pois a pessoa sente-se impelida a interagir com o boneco e diverte-se com o mesmo, até se não souber utilizá-lo de forma “correta”.

As lentes foram selecionadas após leitura extensiva do livro de Schell e discussões com os *designers* das empresas citadas anteriormente. Elas podem ser encontradas na Tabela 5.1.

5.2.2 Perfis

Para resolver o problema da dificuldade da identificação de personas relatada na seção 4.2.2, a solução proposta foi o uso de um perfil que tenha as principais informações, como faixa etária do público, seus mercados de jogos e sua plataforma de distribuição, por exemplo. Não é preciso saber se essa persona é solteira ou casada, tem filhos ou não, empregada ou não. Entretanto, esse problema é menor, e em alguns casos o uso dessa técnica pode melhorar significativamente o resultado. Assim, essa solução fica como opcional, podendo-se utilizar tanto personas como perfis.

5.3 Estruturação do Processo

Nessa seção explicaremos o processo proposto, chamado *Blue Ocean Strategy System* (BOSS), ou Sistema de Estratégia do Oceano Azul. BOSS consiste em um processo dividido em nove passos, listados abaixo:

- 1) *Definição dos mercados primário, secundário e terciário.*
- 2) *Criação de personas ou perfis.*
- 3) *Pesquisa dos produtos concorrentes.*
- 4) *Análise dos produtos concorrentes.*
- 5) *Seleção de lentes dos produtos existentes.*

¹⁴ Tamagotchi era um brinquedo virtual no formato de um ovo, onde o jogador criava um animal de estimação.

- 6) *Criação das curvas de valor dos produtos existentes.*
- 7) *Criação de novas curvas de valor*
- 8) *Análise das curvas criadas.*
- 9) *Seleção do melhor título proposto.*

Esses passos serão descritos em detalhes nas subseções seguintes.

Tabela 5.1: Lentes selecionadas, com o número da lente no livro de Schell (SCHELL, 2008)

Número	Nome	Número	Nome
1	Lente da Experiência essencial	36	Lente da Competição
2	Lente da Surpresa	39	Lente do Tempo
4	Lente da Curiosidade	40	Lente da Recompensa
5	Lente do Valor endógeno	43	Lente da Elegância
6	Lente da Solução de problemas	44	Lente da Personalidade
7	Lente da Tétrade elementar	45	Lente da Imaginação
15	Lente do brinquedo	46	Lente da Economia
17	Lente do prazer	49	Lente da Progreção visível
18	Lente do fluxo	50	Lente do Paralelismo
19	Lente das necessidades	52	Quebra-cabeças (puzzle)
20	Lente da avaliação	53	Lente do Controle
21	Lente do espaço funcional	57	Lente do <i>Feedback</i>
24	Lente da ação	60	Lente dos Modos
27	Lente da Habilidade	63	Lente da Beleza
28	Lente do valor esperado	64	Lente da Projeção
30	Lente da equidade	65	Lente da Máquina narrativa
31	Lente do Desafio	71	Lente da Liberdade
32	Lente das escolhas significativas	74	Lente do Mundo
34	Lente da Habilidade vs probabilidade	81	Lente da Transformação do personagem
35	Lente da Mente e das Mãos	84	Lente da Amizade

5.3.1 Definição dos Mercados Primário, Secundário e Terciário

A definição dos mercados-alvo é o passo inicial do processo. Seu objetivo principal é definir o público-alvo. Nesse caso, os mercados representam gêneros de jogos. Eles podem ser gêneros gerais, como estratégia ou jogos de tabuleiro, ou gêneros específicos, como *tower defense* ou xadrez.

A ordem de importância o público-alvo é representada pelos mercados primário, secundário e terciário. Por exemplo, se o objetivo é desenvolver um jogo casual para dispositivos móveis, uma pessoa pode escolher jogos casuais como mercado primário, simulação como secundário e estratégia como terciário.

Artefatos ao final desse passo

Ao final desse passo, a equipe deve ter em mãos:

- *Três mercados definidos: primário, secundário e terciário;*

5.3.2 Criação de personas ou perfis

Nesse passo, devem ser desenvolvidos dois perfis (ou personas) de jogador. Idealmente, seriam desenvolvidas duas personas, mas a criação de personas é uma atividade mais trabalhosa que a criação de um perfil (que só apresenta dados gerais, como faixa etária, e alguns jogos principais). Durante esse trabalho, entretanto, chamaremos de perfil. O primeiro perfil representa o público-alvo principal. É a audiência que deve ser agradada a qualquer custo. O segundo perfil representa clientes que não devem ser insatisfeitos, mas não são o consumidor principal. Em outras palavras, eles jogariam o jogo se não tivessem opções mais próximas de suas preferências. Esses dois perfis devem ser relativamente similares. O uso de perfis muito diferentes, ou uma quantidade muito grande dos mesmos, pode resultar em um jogo que não tem uma audiência bem definida, e, portanto, não agrada ninguém.

As personas ou perfis podem ser criados utilizando *data mining* ou *user analytics*. Um exemplo seria usar dados reais de jogos lançados anteriormente. *User analytics* pode prover um conhecimento mais profundo sobre os jogadores, tornando possível aprender sua idade, gênero, comportamentos comuns, entre outras informações úteis (DRACHEN; CANOSSA; EL-NASR, 2013).

Artefatos adicionados ao final desse passo

Ao final desse passo, o seguinte artefato é acrescentado aos demais gerados:

- *Duas personas ou perfis: primária e secundária.*

5.3.3 Pesquisa dos produtos concorrentes

Em seguida, devem-se pesquisar os jogos mais populares de cada mercado. Isso importa na hora de descobrir que jogos são populares e, principalmente, por quê. Como dois perfis já foram criados, é possível escolher um conjunto de jogos de seus títulos favoritos. Todos os mercados devem ser representados por ao menos alguns jogos. Após alguns testes, encontramos um bom balanço numa quantidade de 5 jogos, onde 2 são primários, 2 secundários e 1 terciário.

Artefatos adicionados ao final desse passo

- *Conjunto de jogos: conjunto de jogos para cada mercado. Quantidade ideal: 5; 2 primários, 2 secundários e 1 no terciário.*

5.3.4 Análise dos produtos concorrentes

Em seguida, é feita uma análise por um conjunto de pessoas de diferentes especialidades. Produtores, programadores, designers, artistas, pessoas de área de marketing e gerentes devem trabalhar juntos. Se possível, ao menos uma pessoa de cada domínio deve estar presente. Isso gera um grupo diversificado, onde cada um defende seu ponto de vista. Um artista, por exemplo, pode dar ideias estéticas importantes, permitindo um programador discutir ativa e imediatamente se tal ideia impacta ou não a programação.

O grupo, então, analisa os jogos mencionados acima. Essa avaliação, chamada de avaliação interna, é usada para progredir o desenvolvimento. Aspectos discutidos incluem arte, jogabilidade, o que deu certo ou errado em cada título, entre outros.

Artefatos adicionados ao final desse passo

- *Avaliação dos jogos selecionados no passo anterior;*

5.3.5 Seleção de lentes dos produtos existentes

Após analisar cada jogo, a equipe seleciona um conjunto de lentes que melhor represente aqueles jogos. Essas lentes, propostas por Schell (2008), foram descritas em

mais detalhes na Seção **Erro! Fonte de referência não encontrada.** Para cada mercado, o time deve definir um conjunto diferente de lentes. Elas descrevem o jogo em uma faixa de 0 (pouco ou nenhuma apresentação dessa lente nesse jogo) a 5 (essa lente está muito presente no jogo). A experiência da Manifesto Games e BigHut Games indica que 10 lentes é um número ideal, onde 5 lentes representam os jogos primários, 3 os jogos secundários, e 2 o jogo terciário.

Artefatos adicionados ao final desse passo

- *Conjunto de lentes: conjunto de 10 lentes que representam os jogos.*

5.3.6 Criação das curvas de valor dos produtos existentes

As curvas de valor dos jogos selecionados devem ser criadas, utilizando as lentes selecionadas na etapa anterior. Essas curvas devem, então, ser plotados no painel de estratégia. As lentes preenchem o eixo horizontal, e o eixo vertical contem valores de 0 a 5. É importante notar que, embora a quantidade de jogos utilizados seja limitada, eles devem ser capazes de representar seus respectivos mercados, indicando as características semelhantes entre si.

Artefatos adicionados ao final desse passo

- *Curvas de valor dos jogos concorrentes: descrição dos jogos no painel de estratégia usando as lentes selecionadas.*

5.3.7 Criação de novas curvas de valor

Os itens gerados nos passos anteriores indicam características interessantes de cada mercado. Com eles representados no painel de estratégia, o passo seguinte é a proposta de novas curvas de valor. Essas curvas são geradas alterando os valores das demais curvas. Dá-se preferência às lentes do mercado primário, de modo a garantir alguma similaridade com o mercado do público-alvo principal. Ainda assim, algumas lentes desse mercado devem ser diminuídas ou removidas. Então lentes do mercado secundário e terciário devem ser adicionadas ou aumentadas.

Idealmente, a equipe participando no processo deveria se dividir e gerar as curvas em dois grupos separados. O objeto é diminuir um pouco sua predisposição natural em gerar jogos que eles gostariam de jogar ou que já estão acostumados. Embora essa predisposição não seja eliminada, pois ainda existe durante a geração, o impacto é

diminuído na fase seguinte, quando as curvas geradas são trocadas. Assim, o time oposto se vê forçado a propor um jogo para uma curva que não criou.

Artefatos adicionados ao final desse passo

- *Curvas de valor originais: curvas geradas com base nos concorrentes.*

5.3.8 Análise das curvas criadas

Esse passo consiste em analisar os conceitos propostos utilizando as curvas de valor resultantes das etapas anteriores. Embora não exista um melhor ou pior método de análise, uma forma simples pode ser ranquear cada proposta por algum atributo (e.g. conceito mais diferente dos concorrentes, conceito com maior valor em determinada lente, conceito com menor custo de desenvolvimento, etc).

Outra forma seria realizar uma votação, onde cada membro do time ranqueia as propostas em determinada ordem. Assim, o jogo com *ranking* geral mais positivo seria o melhor.

Artefatos adicionados ao final desse passo

- *Uma análise de cada curva criada.*

5.3.9 Seleção do melhor título proposto

Nesse passo, a equipe escolhe a melhor curva criada, baseando-se na análise do passo anterior. Independente do método de seleção, o conceito escolhido possui um conjunto de valores para cada lente, representando um esqueleto do que virá a ser o *design* final do jogo.

5.4 Formalização do Processo

A representação visual da formalização do processo foi feita utilizando o *Software & Systems Process Engineering Meta-Model 2.0* (SPEM). Ele é um *framework* conceitual e um meta-modelo de processo de engenharia. SPEM permite modelar, documentar, apresentar, gerenciar e desenvolver processos (OMG, 2008). Utilizando SPEM, a Figura 5.1 apresenta o fluxo geral de atividades do BOSS, enquanto a Figura 5.2 demonstra o diagrama de atividades, com os artefatos criados em cada atividade.

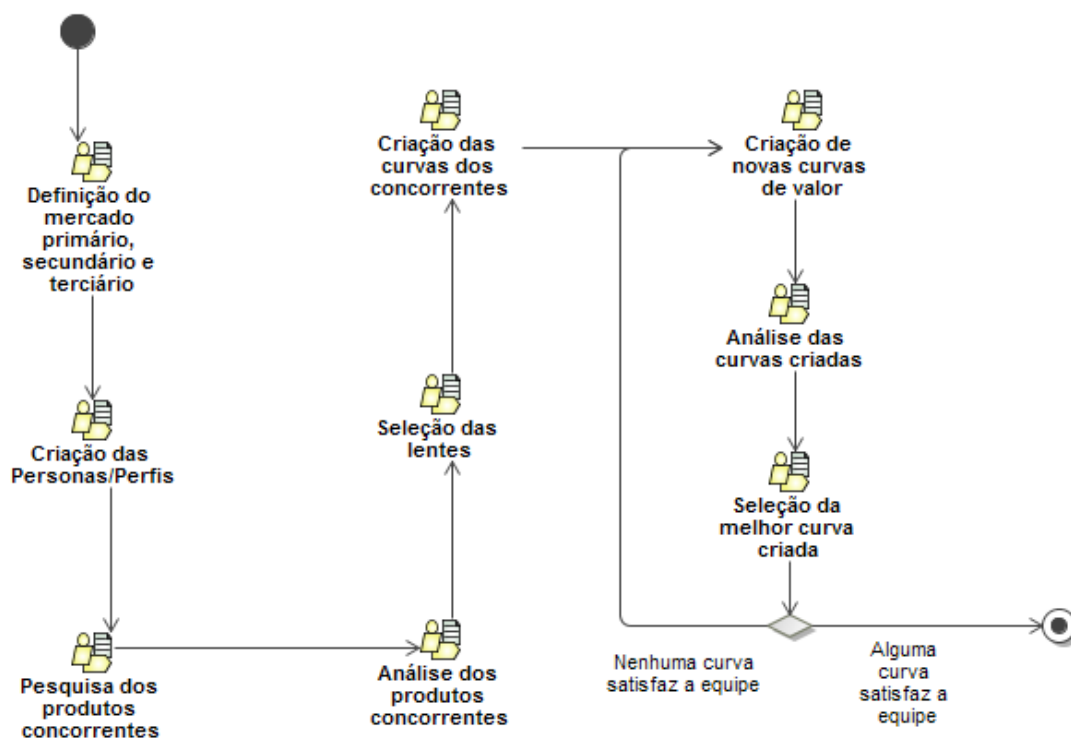


Figura 5.1: Fluxo geral de atividade do BOSS.

5.5 Considerações Finais

Esse capítulo apresentou o processo BOSS, estruturado e formalizado a partir da experiência da Manifesto Games e BigHut Games. Esse processo utiliza conceitos da estratégia de inovação do oceano azul, e objetiva aumentar as chances de sucesso de um jogo através de um processo de conceituação. O próximo capítulo apresenta uma ferramenta de apoio a esse processo.

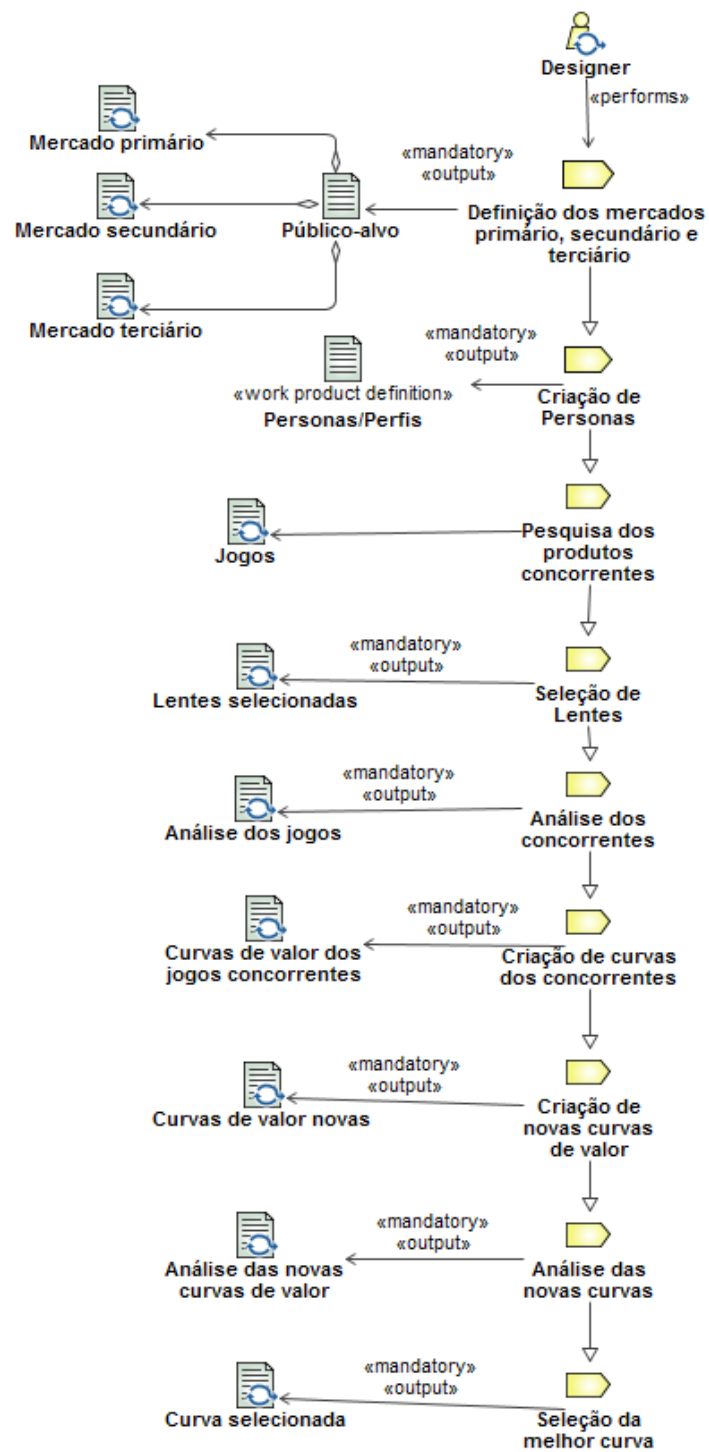


Figura 5.2: Diagrama de atividades do BOSS, com os artefatos de cada atividade.

6. Ferramenta BOSS Tools

Essa seção descreve o desenvolvimento da ferramenta de apoio ao processo proposto no capítulo anterior, chamada BOSS Tools. Inicialmente, formas de auxiliar o processo foram conjecturadas, conforme descrito na Seção 6.1. Alguns problemas foram selecionados e uma ferramenta foi desenvolvida. A Seção 6.2 apresenta a ferramenta final e suas principais funcionalidades.

6.1 Abordagem seguida

Tendo em mente o processo descrito na Seção 5.3, e as falhas e limitações listadas na seção 4.2, buscou-se encontrar maneiras de auxiliar o processo através de uma ferramenta de apoio. Desse modo, para cada fase do processo foram descritos seus principais problemas. A Tabela 6.1 sumariza esses dados.

Tabela 6.1: Problemas sugeridos para cada etapa do processo

Problema		Desafios
<i>P1</i>	Definição dos mercados	Identificar que estilos estão em tendência; quais os melhores estilos diferentes para trabalhar; descrever os jogos dos estilos; utilizar estatísticas de mercado em função de cada estilo.
<i>P2</i>	Criação de personas / perfis	Como saber quem joga determinado jogo; como obter as características do jogador; dentre certo grupo de pessoas, qual é a mais representativa.
<i>P3</i>	Escolha de jogos para representar um mercado	Dado um mercado, que jogos são mais representativos.
<i>P4</i>	Escolha das lentes	Que lentes são mais apropriadas, como definir o valor dessas lentes (0-5).
<i>P5</i>	Criação de curvas dos produtos concorrentes	Qual a melhor ordem das lentes na curva; representação visual da curva.
<i>P6</i>	Criação de curvas originais	Como selecionar os valores para cada lente; que informações utilizar como entrada; representação visual da curva.
<i>P7</i>	Análise e seleção de curvas originais	Como avaliar as curvas; que parâmetros utilizar; como definir a melhor proposta.

Em seguida, cada problema sugerido foi analisado para verificar se prover uma solução para o mesmo seria factível, dado o escopo e tempo disponível na realização do

presente trabalho. Essa análise foi executada junto a integrantes da equipe das empresas citadas anteriormente, de modo a termos uma visão do que melhor os ajudaria. O resultado e as justificativas para cada problema são apresentados na Tabela 6.2.

Tabela 6.2: Resultado da análise dos problemas e justificativa de seleção.

Problema	Resultado	Justificativa
<i>P1</i>	Descartado	Embora seja um problema interessante, depende de um estudo mais profundo de mineração e processamento de linguagem natural, e assim torna-se inviável para desenvolvimento durante esse projeto.
<i>P2</i>	Descartado	Foi descartado devido à quantidade de trabalho necessária se maior que o tempo disponível, e o benefício não superar o custo.
<i>P3</i>	Descartado	Os melhores jogos selecionados podem não ser os mais comentados ou com melhores críticas e mais fãs, já que a escolha também depende do tipo de produto final desejado ou dos demais jogos selecionados. Assim, o nível de subjetividade era alto demais para solucioná-lo satisfatoriamente.
<i>P4</i>	Descartado	Novamente, problema subjetivo demais para ser automatizado satisfatoriamente.
<i>P5</i>	Descartado parcialmente	Embora a parte de definição dos valores das lentes dependa muito do ponto de vista do avaliador, a representação visual da curva pode ser gerada automaticamente.
<i>P6</i>	Selecionado	Um dos problemas mais custosos, em relação a tempo, e manuais. Também foi o que a equipe das empresas teve mais interesse em ver solucionado.
<i>P7</i>	Descartado	Como os problemas P3 e P4, a análise e seleção de curvas é um problema muito subjetivo e não foi atacado no presente trabalho.

6.2 Descrição da ferramenta

6.2.1 Arquitetura do sistema e dados técnicos

Com o objetivo de auxiliar o processo BOSS, foi desenvolvida a ferramenta BOSS Tools, que permite o armazenamento e a criação de curvas de valor. A arquitetura do sistema é uma arquitetura cliente-servidor, conforme mostra a Figura 6.1. Os módulos dessa arquitetura são brevemente descritos abaixo:

- **Camada da aplicação cliente:** e é responsável pela mais parte do funcionamento da ferramenta, incluindo interface gráfica e lógica do programa. Essa lógica inclui a geração de curvas de valor e edição de dados armazenados.

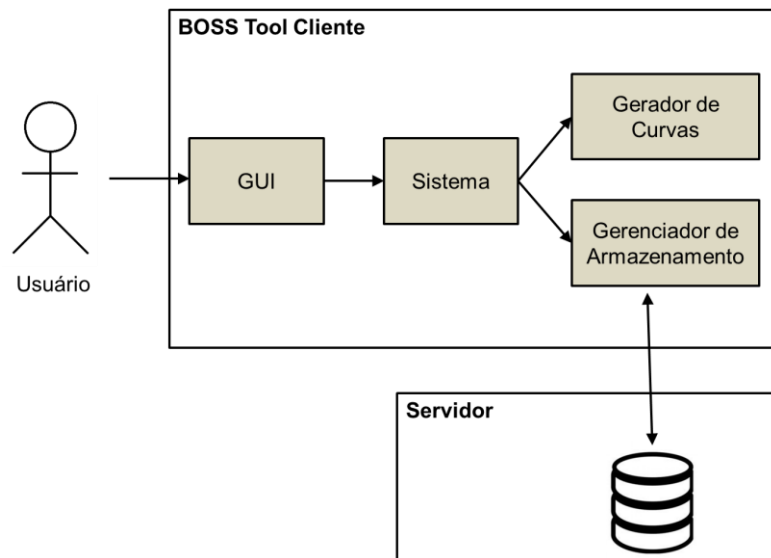


Figura 6.1: Arquitetura do BOSS Tool.

- **GUI (*Graphical User Interface*):** módulo responsável por fornecer uma interface visual para o usuário, contendo as funcionalidades do sistema. Seu objetivo é permitir que a interação entre o usuário e o sistema ocorra de forma mais natural.
- **Módulo do sistema:** responsável pela comunicação entre a interface gráfica e os componentes internos do sistema (lógica e persistência de dados).
- **Gerador de curvas:** módulo de geração de curvas de valor. Recebe como entrada um conjunto de jogos e lentes e, utilizando do algoritmo de geração descrito na Seção 6.2.2, retorna uma nova curva de valor.
- **Gerenciador de armazenamento:** responsável pela comunicação entre o banco de dados e o restante do sistema.
- **Camada do servidor:** armazena as informações dos jogos e lentes cadastrados, sendo responsável pela persistência dos dados.

A ferramenta foi desenvolvida na linguagem JAVA, utilizando a API *JFreeChart* para geração dos gráficos. O ambiente de programação escolhido foi o Eclipse. Também utilizou-se o SGBD *Postgres* 9.3 para armazenar os dados dos jogos e lentes cadastrados no sistema. A Figura 6.2 apresenta a tela inicial da ferramenta.

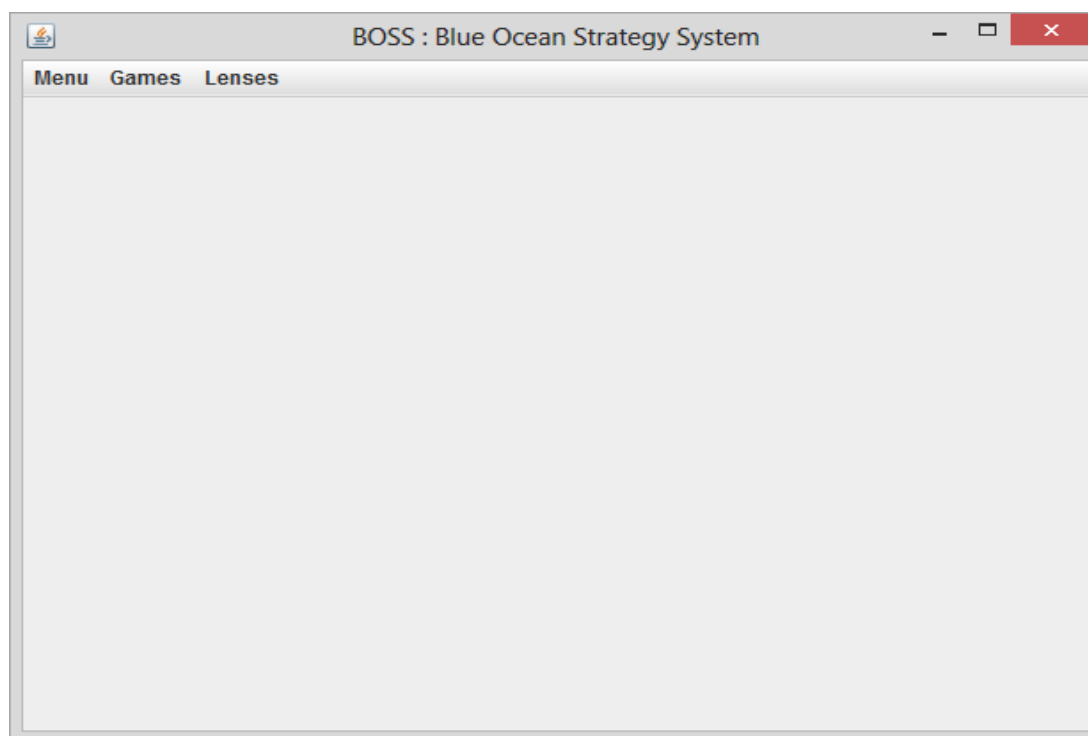


Figura 6.2: Tela inicial do BOSS Tools.

6.2.2 Sobre o algoritmo de geração de curvas

Criar o algoritmo de geração de curvas gerou um novo problema: como garantir que as curvas façam sentido? Ou seja, que os valores não sejam absurdos, do ponto de vista do oceano azul. Por exemplo, não faz sentido gerar uma curva onde todos os valores estão altos, pois se ataca muita coisa ao mesmo tempo, não dando prioridade a nada.

Esse problema foi contornado com o acréscimo de dois fatores: o conceito de Restrição e uma regra de porcentagem de valores. A restrição determina que comportamentos específicos uma lente deve ter. Por exemplo, pode-se definir que certa lente é importante para o produto final, então ela deve ter valores altos. Ou que outra lente já foi muito utilizada, e é melhor sempre excluí-la. Assim, foram definidas 7 restrições:

- ***NO_RESTRICTION*** : determina que não se deve aplicar nenhuma restrição àquela lente;
- ***DONT_CHANGE*** : determina que o valor da lente não deve ser alterado. Assim, se essa lente é secundária, ela terá o valor de um dos jogos secundários selecionados. Por exemplo: suponha que tenhamos dois jogos selecionados, A

e B, e uma lente X. O valor da lente X em A é 4, e em B é 2. Então os valores possíveis de X na nova curva de valor são 2 ou 4;

- **CHANGE_ONLY_UP** : todas as curvas geradas devem ter um valor maior que o do de um jogo aleatório daquele mercado. Por exemplo: suponha que tenhamos dois jogos selecionados, A e B, e uma lente X. O valor da lente X em A é 4, e em B é 3. Então os valores possíveis de X na nova curva de valor será 3 ou 4, dependendo de qual jogo seja sorteado;
- **CHANGE_ONLY_DOWN** : similar a **CHANGE_ONLY_UP**, mas todos os valores serão inferiores;
- **CANT_BE_ZERO** : determina que a lente não pode ter valor 0;
- **CANT_BE_FIVE** : determina que a lente não pode ter valor 5;
- **DETERMINANT** : define que a lente é essencial, e seu valor deve ser sempre maior ou igual a 4.

A regra de porcentagem, por outro lado, determina que exista um limite máximo de lentes que podem ter valores altos. Assim, apenas uma porcentagem de cada mercado recebe um valor alto.

Como entrada, o algoritmo recebe um conjunto de lentes primárias, secundárias e terciárias, um conjunto de jogos primários, secundários e terciários, e um conjunto de restrições associadas às lentes. Sua saída será uma curva de valor que contem um conjunto n de pares <Lente, Valor>, onde n é a quantidade de lentes selecionadas.

O algoritmo adiciona valores às lentes, na seguinte ordem de prioridade de restrição, do primeiro ao último: *DONT_CHANGE*, *DETERMINANT*, *CHANGE_ONLY_UP*, *CHANGE_ONLY_DOWN*, *CANT_BE_FIVE*, *CANT_BE_ZERO*, *NO_RESTRICTION*. Um pseudocódigo do algoritmo encontra-se abaixo, em Algoritmo 1, e o algoritmo completo pode ser encontrado no Apêndice A.

Algoritmo 1: GeraçãoAutomáticaDeCurvas(C_J, C_L, R)

ENTRADA: Conjunto de curvas de valor dos jogos concorrentes C_J , onde cada curva de valor representa um conjunto de pares <Lente, Valor>.
 Conjunto de lentes selecionadas C_L .
 Conjunto de restrições das lentes R .

VARIÁVEIS: x, y, z, l

SAÍDA: Curva de valor C.

```

1   $C \leftarrow \emptyset$ ;
2  para cada  $l$  em  $C_L$  faça
3      se  $R[l] = R.DO\_NOT\_CHANGE$  faça
4           $y \leftarrow$  gere um valor randômico maior ou igual a 0 e menor que a quantidade de
5          jogos em  $C_J$  com mesmo mercado que  $l$ 
6           $x \leftarrow$  valor da lente  $l$  da curva  $C_J[y]$ 
7           $C \leftarrow \langle l, x \rangle$ 
8      fim do se
9  fim do para
10 para cada  $l$  em  $C_L$  faça
11     se  $R[l] = R.DETERMINANT$  faça
12          $y \leftarrow$  gere um valor randômico  $\geq 4$  e  $< 5$ 
13          $C \leftarrow \langle l, y \rangle$ 
14     fim do se
15 fim do para
16 para cada  $l$  em  $C_L$  faça
17     se  $R[l] = R.CHANGE\_ONLY\_UP$  faça
18          $y \leftarrow$  gere um valor randômico maior ou igual a 0 e menor que a quantidade de
19         jogos em  $C_J$  com mesmo mercado que  $l$ 
20          $x \leftarrow$  valor da lente  $l$  da curva  $C_J[y]$ 
21          $z \leftarrow$  gere um valor randômico maior ou igual a  $x$  e menor ou igual a 5
22          $C \leftarrow \langle l, z \rangle$ 
23     fim do se
24 fim do para
25 para cada  $l$  em  $C_L$  faça
26     se  $R[l] = R.CHANGE\_ONLY\_DOWN$  faça
27          $y \leftarrow$  gere um valor randômico maior ou igual a 0 e menor que a quantidade de
28         jogos em  $C_J$  com mesmo mercado que  $l$ 
29          $x \leftarrow$  valor da lente  $l$  da curva  $C_J[y]$ 
30          $z \leftarrow$  gere um valor randômico menor ou igual a  $x$  e maior ou igual a 0
31          $C \leftarrow \langle l, z \rangle$ 
32     fim do se
33 fim do para
34 para cada  $l$  em  $C_L$  faça
35     se  $R[l] = R.CANT\_BE\_ZERO$  faça
36          $y \leftarrow$  gere um valor randômico maior que 0 e menor ou igual a 5
37          $C \leftarrow \langle l, y \rangle$ 
38     fim do se
39 fim do para
40 para cada  $l$  em  $C_L$  faça
41     se  $R[l] = R.CANT\_BE\_FIVE$  faça
42          $y \leftarrow$  gere um valor randômico maior ou igual a 0 e menor que 5
43          $C \leftarrow \langle l, y \rangle$ 
44     fim do se
45 fim do para
46 para cada mercado  $x$  em {primário, secundário} faça
47     enquanto quantidade de lentes de  $x$  com valor acima de 3 em  $C < 2/3$  da quantidade de
48      $C_L$  faça
49          $l \leftarrow$  lente em  $C_L$  que ainda não existe em  $C$ 
50          $C \leftarrow \langle l, \text{valor randômico maior ou igual a 3 e menor ou igual a 5} \rangle$ 
51     fim do enquanto
52     enquanto existirem lentes de  $x$  fora de  $C$  faça

```

```

53          $l \leftarrow$  lente em  $C_L$  que ainda não existe em  $C$ 
54          $C \leftarrow \langle l, \text{valor randômico maior ou igual a 0 e menor que 3} \rangle$ 
55     fim do enquanto
56 fim do para
57 para cada lente  $l$  do mercado {terciário} fora de  $C$  faça
58      $C \leftarrow \langle l, \text{valor randômico maior ou igual a 0 e menor ou igual a 5} \rangle$ 
59 fim do para
60 retorne  $C$ ;

```

Fim GeraçãoAutomáticaDeCurvas

6.2.3 Funcionalidades do BOSS Tools

A Figura 6.3 apresenta as principais funcionalidades do sistema. Elas podem ser encontradas no menu da ferramenta, como mostrou a Figura 6.2. Elas também são descritas abaixo:

Criar curva de jogo

Encontra-se como “Create curve”, sob a opção “Menu” do menu da ferramenta. Permite a criação de uma nova curva, utilizando o algoritmo descrito na seção anterior. Consiste em um processo de quatro etapas:

Inicialmente ela abre a tela de seleção de jogos (Figura 6.4), onde o usuário deve escolher os jogos que deseja (1) e adicioná-los à lista de jogos (2). Para cada jogo escolhido, o usuário deve escolher se ele pertence ao mercado primário, secundário ou terciário (3), antes de avançar para a próxima tela.

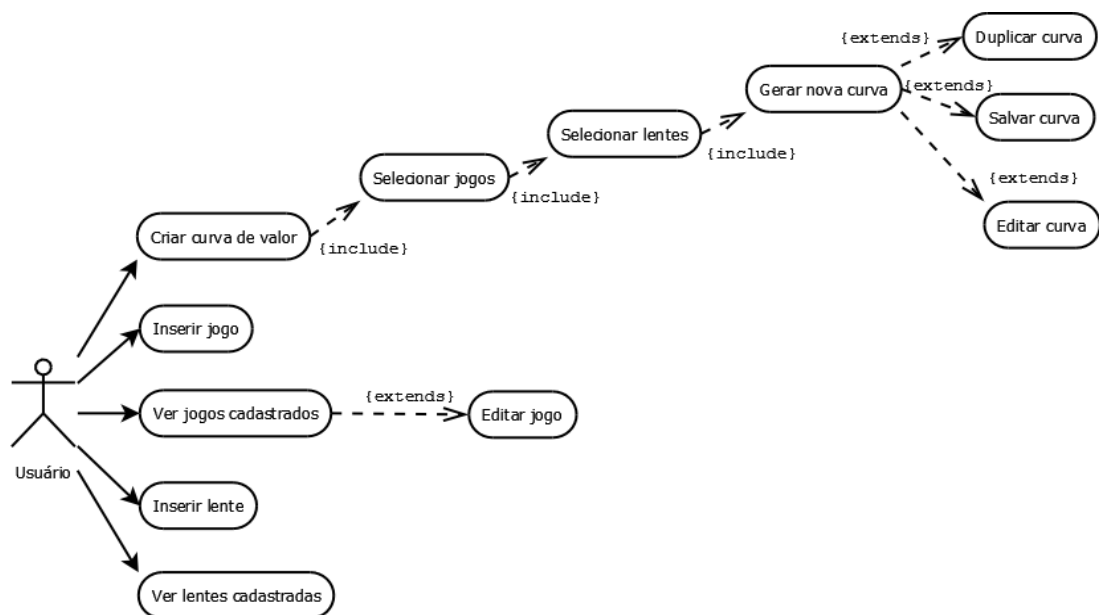


Figura 6.3: Funcionalidades do BOSS Tools.

A tela seguinte é a de seleção de lentes, conforme mostra a Figura 6.5. O usuário pode escolher uma lente (1) e ver sua descrição (2). Se desejar, o usuário pode adicioná-la a lista. É preciso, ainda, escolher o mercado e um conjunto de restrições àquela lente (3). Ainda nessa tela é possível selecionar 10 lentes aleatórias (4), visualizar os jogos escolhidos na etapa anterior (5), ou retornar uma etapa (6).

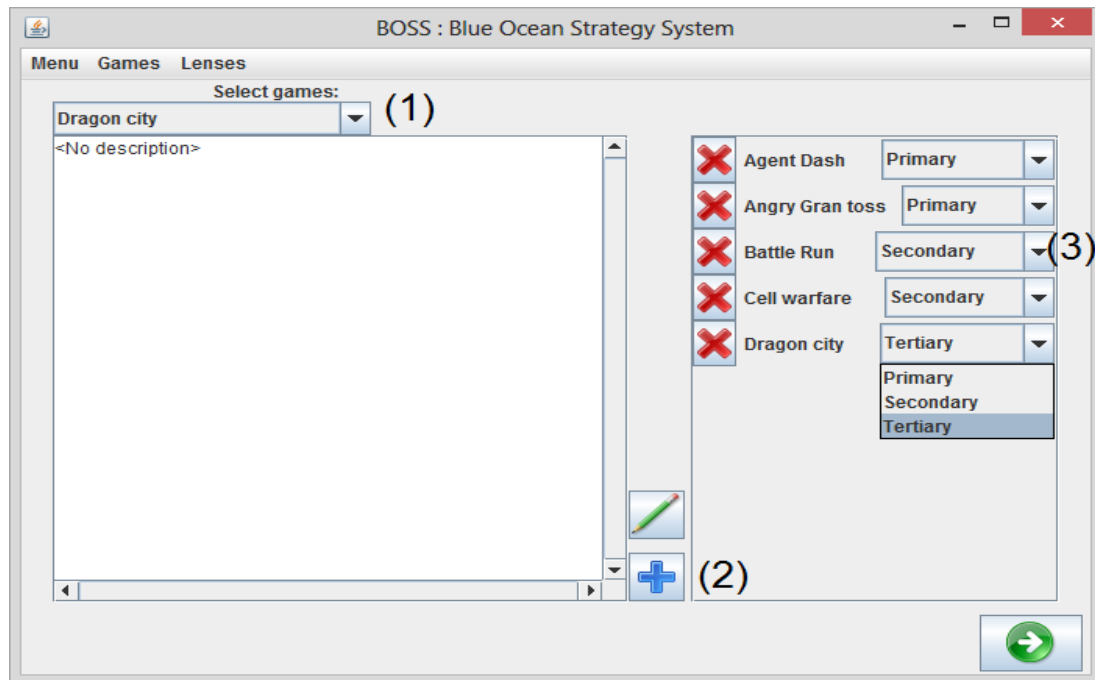


Figura 6.4: Tela de seleção de jogos.

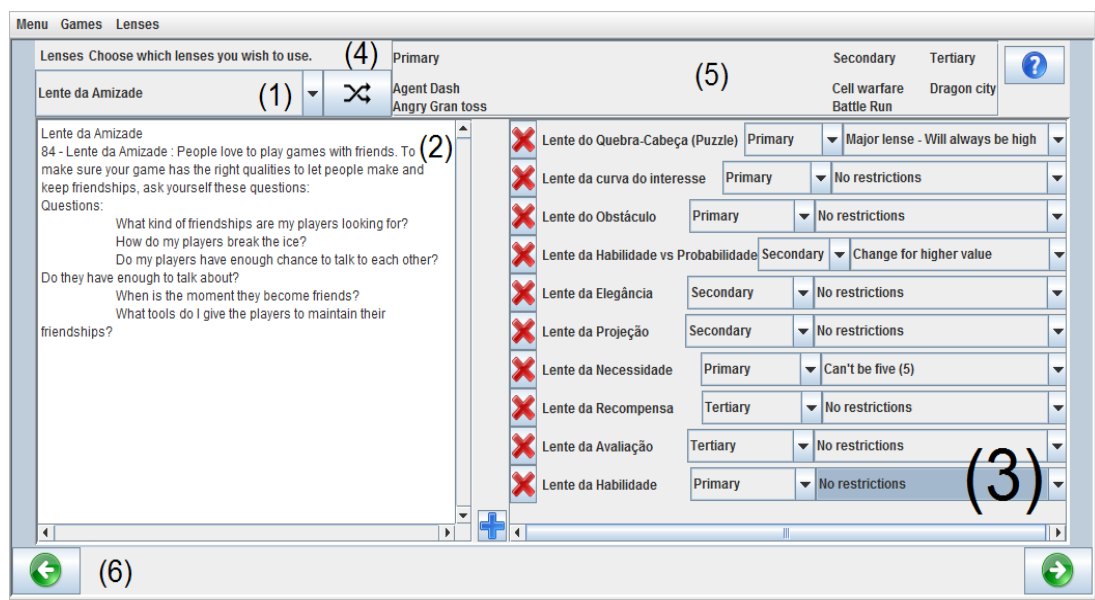


Figura 6.5: Tela de seleção de lentes.

A terceira etapa (Figura 6.6) consiste em verificar se todos os jogos estão com as lentes preenchidas. Caso algum jogo esteja com um valor de uma lente faltando, um aviso aparece. Se o usuário clicar nesse aviso, ele é encaminhado à tela de edição de jogo.

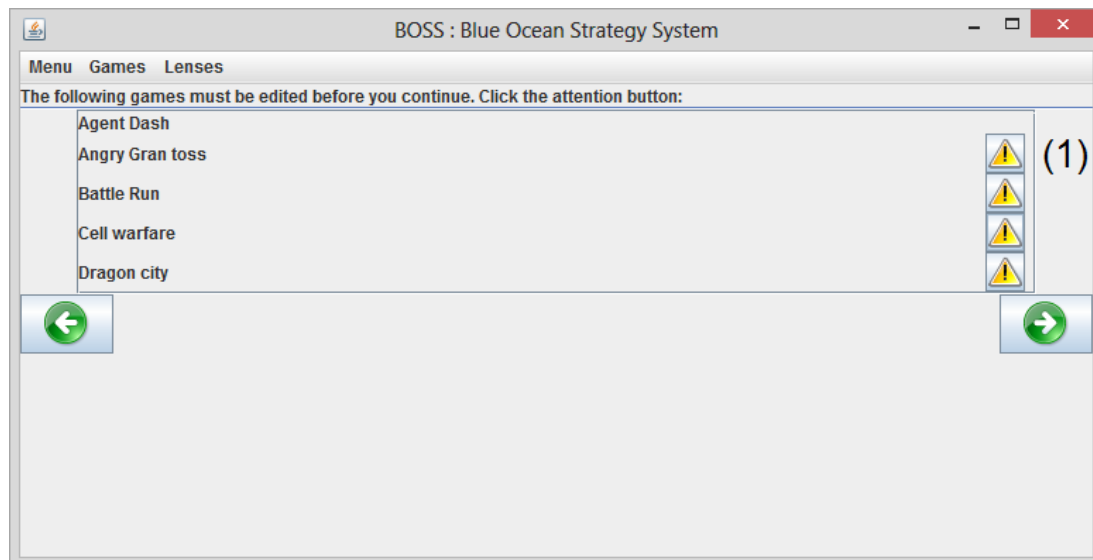


Figura 6.6: Tela que verifica se todos os jogos possuem valores para as lentes selecionadas.

A última etapa é a tela de geração de curvas, mostrada na Figura 6.7. À esquerda, ela apresenta um painel de estratégia (1) com as lentes e os jogos selecionados. Curvas geradas também são desenhadas automaticamente, mas uma opção (2) permite escondê-las para visualização seletiva. À direita existe a lista dos jogos selecionados e curvas geradas (3).

Nessa tela é possível criar uma nova curva (4), duplicar uma existente (5), salvar uma curva no sistema (6) ou exportar a visualização atual do painel de estratégia como uma imagem (7).

Inserir jogo

Permite inserir um jogo novo no sistema, e pode ser acessada pela opção: “Games” -> “Add new game”. Ela abre as telas de edição de jogos, mostradas nas Figuras Figura 6.8, Figura 6.9 e Figura 6.10.

Ver jogos cadastrados

Acessada pela opção “Games” -> “View all games”, apresenta todos os jogos cadastrados no sistema, permitindo que o usuário exclua-os ou edite-os. Figura 6.11 mostra essa tela.

Inserir lente e Ver lentes cadastradas

Similares a Inserir jogo e Ver jogos cadastrados, respectivamente.

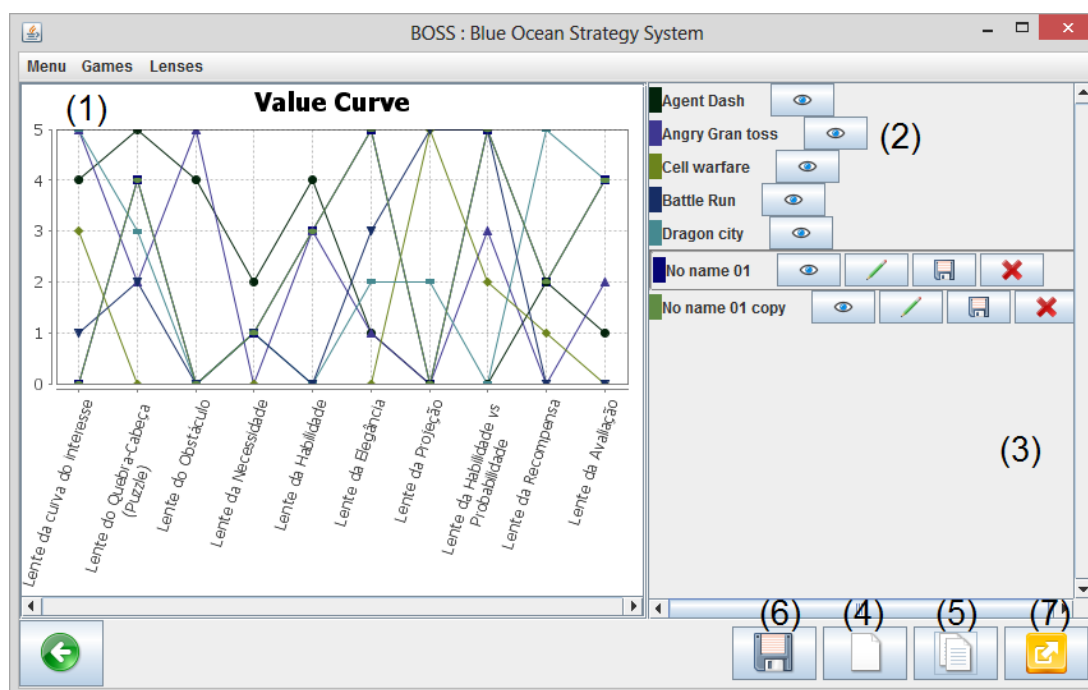


Figura 6.7: Tela de geração de curvas.

The screenshot displays the BOSS: Blue Ocean Strategy System interface, specifically the 'General information' form for inserting a new game. The top menu bar includes 'Menu', 'Games', and 'Lenses'. The main window is titled 'BOSS : Blue Ocean Strategy System'. The form has three tabs: 'General information', 'Lenses', and 'View canvas'. The 'General information' tab is active, showing fields for 'Name:' and 'Description:'. There are also 'Save' and 'Cancel' buttons at the bottom right.

Figura 6.8: Tela de inserção de um jogo novo (Parte 1): informações gerais.

BOSS : Blue Ocean Strategy System

Menu Games Lenses

General information Lenses View canvas

A-F G-L M-R S-Z

Lente da Amizade 0 1 2 3 4 5

Lente da Avaliação 0 1 2 3 4 5

Lente do Avatar 0 1 2 3 4 5

Lente da Ação 0 1 2 3 4 5

Lente da Beleza 0 1 2 3 4 5

Lente do Brinquedo 0 1 2 3 4 5

✓ ✗

Figura 6.9: Tela de inserção de um jogo novo (Parte 2): valores e comentários das lentes.

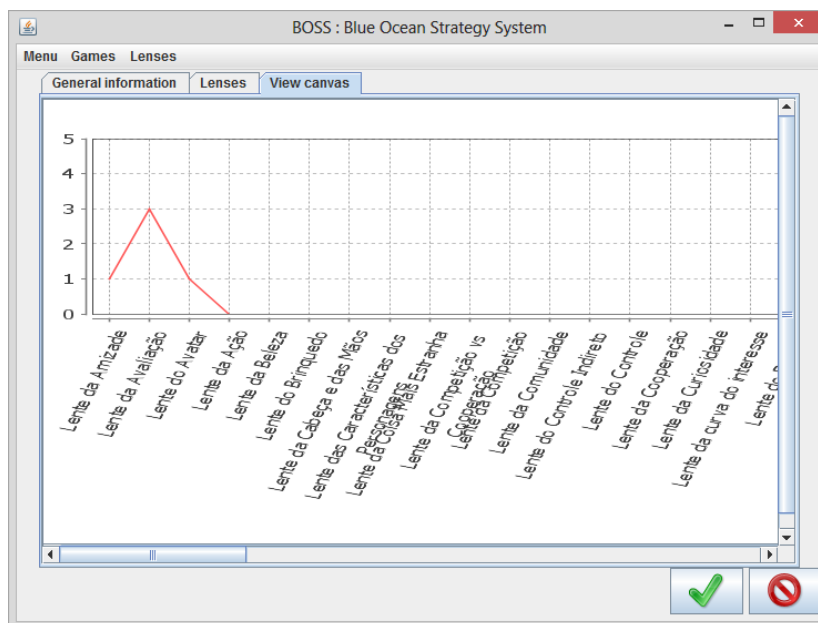


Figura 6.10: Tela de inserção de um jogo novo (Parte 3): painel de estratégia.

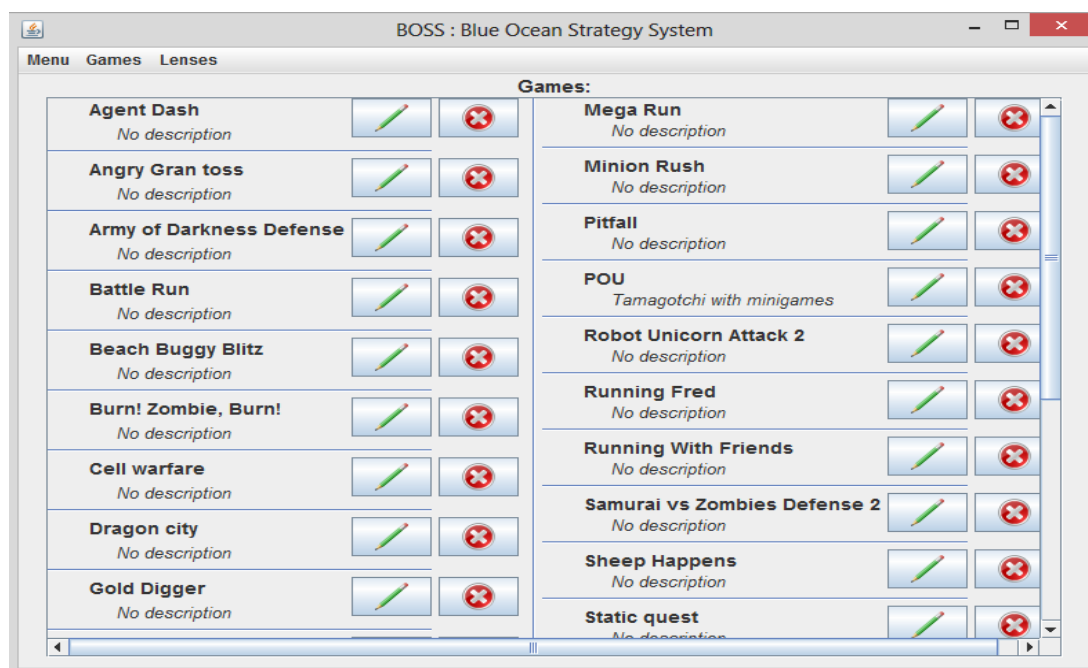


Figura 6.11: Tela de visualização de todos os jogos cadastrados.

6.3 Considerações Finais

Esse capítulo apresentou os aspectos de implementação do BOSS Tools, uma ferramenta de apoio ao processo BOSS. Essa ferramenta fornece funcionalidades que objetivam auxiliar e diminuir o tempo gasto no processo, melhorando a produtividade da equipe. A arquitetura cliente-servidor, na qual foi desenvolvida, permite que vários *designers* utilizem a aplicação simultaneamente, dividindo uma mesma base de dados, embora não a mesma atividade. O próximo capítulo apresenta os experimentos realizados para validar essa ferramenta. Nele também serão descritos e discutidos os resultados de tais experimentos.

7. Experimentos e Avaliação

Esse capítulo apresentará os experimentos realizados para validar a ferramenta BOSS Tools e o processo BOSS, propostos nesse trabalho, analisando a evolução do processo e se o uso da ferramenta pode realmente auxiliá-lo.

7.1 Definição

Essa seção apresenta os objetivos dos experimentos.

7.1.1 *Objetivos gerais*

O objetivo desses experimentos é (1) avaliar se os conceitos essenciais do processo se mantiveram após sua evolução, e (2) avaliar a ferramenta proposta nesse trabalho, de modo a descobrir: i) se ela impacta positivamente nas atividades do processo BOSS; ii) qual a qualidade percebida da ferramenta pelos usuários.

7.1.2 *Medidas de avaliação*

Os seguintes pontos foram utilizados para a avaliação desse trabalho:

- **Produtividade:** Engloba o tempo gasto com a atividade e o custo de realizá-la.
- **Diversidade:** Trata da quantidade de diferentes curvas possíveis de serem geradas, e nível de influência de ideias e gostos pessoais. Quanto menor a quantidade de ideias tendenciosas durante a geração, maior a diversidade. Quanto maior a quantidade de curvas diferentes, maior a diversidade.
- **Facilidade de uso:** Medida subjetiva, referente à percepção dos usuários em relação ao uso da ferramenta. Envolve a dificuldade em realizar tarefas e a clareza da ferramenta.
- **Essência do processo:** Avaliação sobre a evolução do processo, com o objetivo de verificar se, após as alterações, o processo permanece com os conceitos essenciais intactos.

7.1.3 Tipos de avaliação utilizados

O processo foi avaliado por meio de uma entrevista com alguns *designers* da Manifesto Games e BigHut Games. A ferramenta foi avaliada de duas maneiras: uma quantitativa e uma qualitativa. A quantitativa envolveu a análise temporal da *produtividade* e *diversidade* durante e após a utilização do processo, com e sem a ferramenta. A avaliação qualitativa foi obtida por meio de um questionário e realizada após a utilização da ferramenta.

7.2 Experimentos sobre o Processo BOSS

A versão inicial do processo descrito na Seção 5 foi aplicada e testada na indústria, conforme descreve a Seção 4.1. Acreditamos que essa validação de mercado seja suficiente para mostrar que: i) a aplicação do BOS ao processo de conceituação faz sentido; e ii) esse processo pode trazer bons resultados ao produto final, do ponto de vista de inovação.

Devido a limitações de tempo, não pudemos testar a evolução desse processo proposta na Seção 5 no mercado. Entretanto, acreditamos que as alterações no processo não afetaram seus conceitos essenciais, mantendo a base do mesmo. Com o objetivo de validar essa hipótese, realizamos uma entrevista com quatro (4) *designers* da Manifesto Games que utilizaram tanto a versão inicial quanto a atual do processo.

A entrevista consistiu em três perguntas:

1. *Na sua opinião, a evolução do processo manteve os conceitos essenciais do mesmo?*
2. *Você acha que o processo melhorou?*
3. *Como você classificaria a evolução do processo numa escala de 1 a 5, onde 1 equivale a “muito ruim” e 5 equivale a “muito boa”?*

7.3 Experimentos com a Ferramenta BOSS Tools

Os experimentos foram realizados nas empresas locais Manifesto Games e BigHut Games. Dez (10) pessoas participaram da parte quantitativa, sendo, em sua maioria, da área de *design*. Quatro (4) pessoas participaram da parte qualitativa, todas da área de *design*, com experiências variando entre seis meses e seis anos.

7.3.1 *Experimento Quantitativo*

A parte quantitativa do experimento foi realizada em duas iterações do processo, uma totalmente manual e outra com a ferramenta. A primeira etapa da atividade foi igual nas duas iterações. Iniciou-se com uma preparação para os testes, onde os *designers* criaram duas personas, e selecionaram cinco jogos e dez lentes. Em seguida, um *designer* apresentou esses dados, para fornecer explicações básicas sobre as lentes aos participantes que não estavam familiarizados com o processo. Após isso, os processos foram diferenciados.

Na iteração manual, os participantes foram separados em dois grupos. Cada grupo teve 40 minutos para gerar quantas curvas conseguisse. Ao término desse tempo, duas curvas eram selecionadas. Cada grupo teria, então, mais 30 minutos para gerar conceitos com cada curva.

Na iteração com a ferramenta, um participante interagia com o sistema, realizando a atividade de criar duas curvas utilizando as informações definidas na primeira etapa. Essa atividade foi realizada por dois usuários diferentes: um criou as curvas com dados já inseridos no sistema, o outro inseriu as informações durante a geração. Os participantes, então, foram divididos em dois grupos, e cada grupo recebeu uma das curvas. Foram dados 30 minutos para gerar propostas de conceitos para as curvas. Ao fim desse tempo, as curvas eram trocadas, e as equipes deveriam propor novos conceitos para a outra curva.

7.3.2 *Experimento Qualitativo*

No experimento qualitativo, a cada participante foi pedido que interagisse com o sistema, realizando um conjunto de tarefas, listado abaixo:

- *Inserir um jogo*
- *Inserir uma lente*
- *Abrir a tela de visualização dos jogos e editar um jogo após inserir a lente*
- *Repetir a inserção de jogos 4 vezes*
- *Criar uma curva.*

Ao término dessas tarefas, os usuários preencheram um questionário, que pode ser visualizado no Apêndice B.

7.4 Resultados dos Experimentos

As respostas das perguntas 1 e 3 da entrevista estão representadas na Figura 7.1.

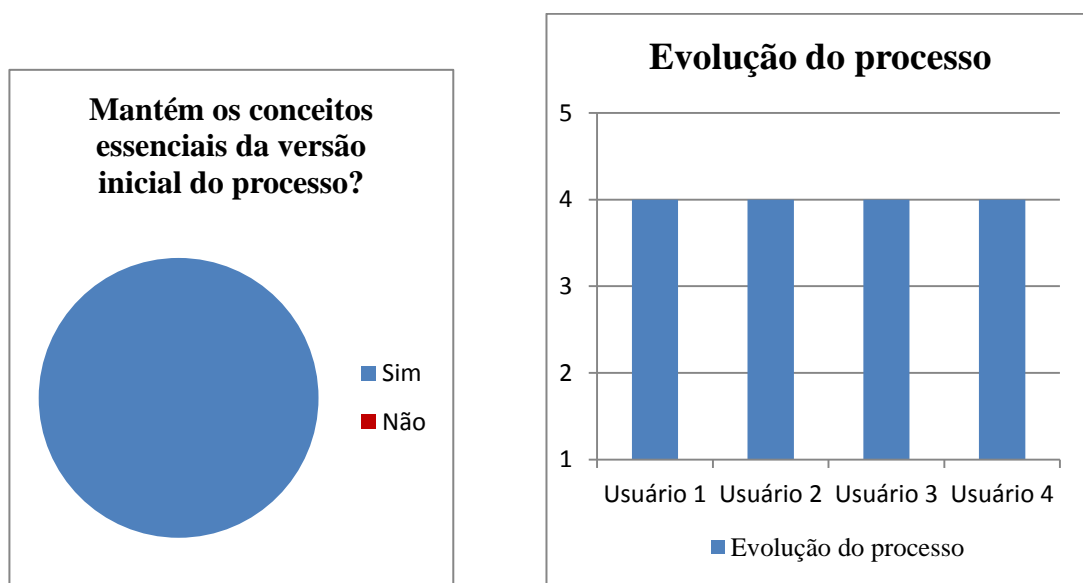


Figura 7.1: Porcentagem de respostas à pergunta sobre a essência do processo após sua evolução, e classificação da evolução do processo (1 = muito ruim, 5 = muito boa).

A Tabela 7.1 sumariza o tempo (T) gasto durante a fase de geração de curvas, com e sem a ferramenta. Com a ferramenta, temos os dados do tempo gasto com os dados inseridos anteriormente, e com os dados inseridos durante os testes. Tabela 7.2 compara o tempo gasto em cada etapa, utilizando a ferramenta, com a inserção dos dados antes (“Com dados”) e durante (“Sem dados”) os experimentos. Além disso, utilizando a ferramenta, foram propostos cinco conceitos de jogos diferentes.

Tabela 7.1: Comparação entre o tempo gasto com e sem a ferramenta

	Manual	BOSS Tool	
		Com dados	Sem dados
T (min) gasto na geração	40	8	32
T (min) gasto na geração, multiplicado pela quantidade de participantes necessários	200	8	32

Os resultados do questionário estão condensados nas figuras abaixo. Figura 7.2 apresenta a utilidade percebida pelos usuários das funcionalidades gerais da aplicação.

Figura 7.3 apresenta a utilidade das funcionalidades específicas secundárias, como armazenamento de lentes e inserção de jogos.

Tabela 7.2: Comparação do tempo gasto por etapa, utilizando a ferramenta, com ou sem dados armazenados previamente.

Tempo \ Etapa	Escolha dos jogos	Escolha das lentes	Criação	Total
T (min) sem dados	0	5	3	8
T (min) com dados	24	5	3	32

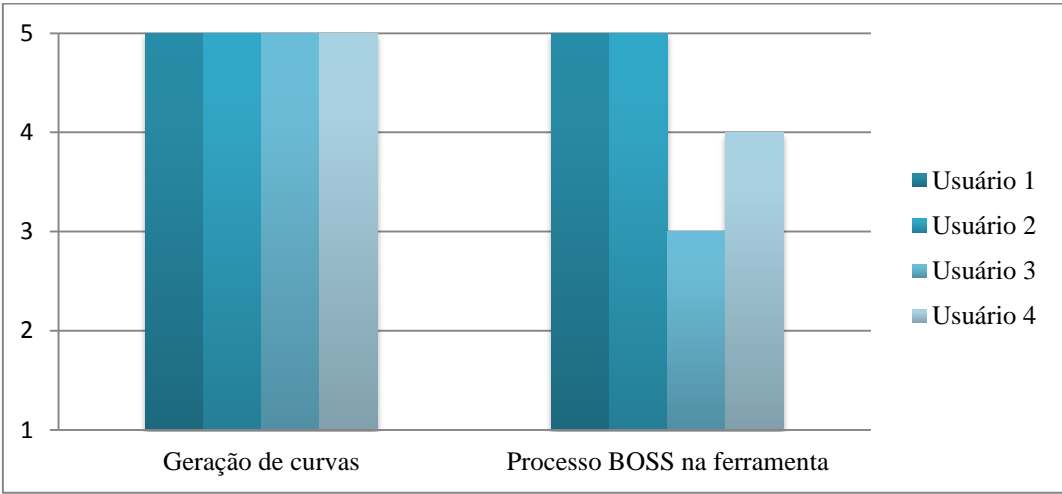


Figura 7.2: Utilidade das funcionalidades da aplicação (1 = inútil, 5 = muito útil) de geração (como um processo num todo) e somente da geração.

Figura 7.4 mostra como os usuários classificaram o sistema e a dificuldade em utilizar a ferramenta. Já a Figura 7.5 mostra a melhoria percebida do processo, em relação à versão inicial do mesmo, e na produtividade da equipe, em relação à versão manual do processo.

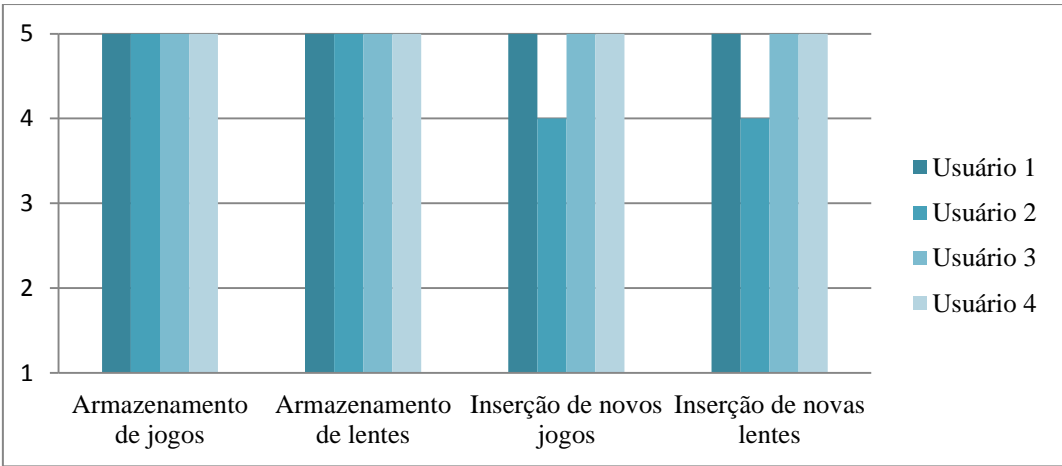


Figura 7.3: Utilidade das funcionalidades específicas da aplicação (1 = inútil, 5 = muito útil).

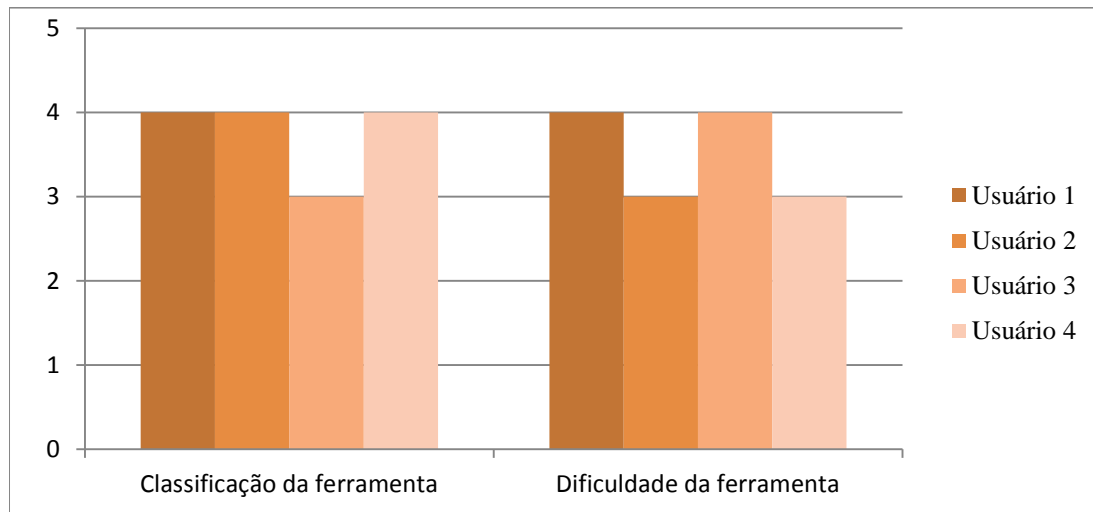


Figura 7.4: Classificação da ferramenta (1 = muito ruim, 5 = muito boa), Dificuldade da ferramenta (1 = muito fácil, 5 = muito complicada).

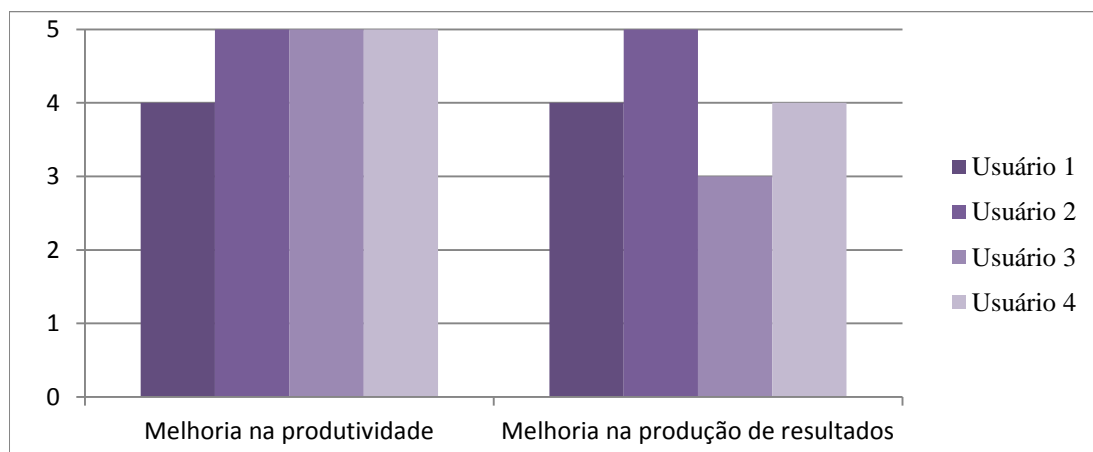


Figura 7.5: Melhoria percebida na produtividade e nos resultados produzidos (1 = Atrapalhou, 5 = Ajudou muito).

7.4.1 Discussão dos resultados

Dentre os resultados apresentados, discutiremos inicialmente as entrevistas. A opinião dos quatro *designers* que participaram dessa etapa foi unânime em relação à questão inicial da entrevista, sobre o BOSS manter os elementos essenciais da versão inicial do processo. A resposta foi que, sim, a evolução do processo ainda contém os elementos principais de sua versão anterior. Isso indica que as mudanças efetuadas não foram suficientes para o processo apresentar uma perda significativa de desempenho num caso de teste real, em comparação com seu antecessor.

Os entrevistados indicaram, ainda, que a evolução do processo foi positiva. Alguns dos comentários foram “(o processo) ficou mais claro” e “ficou melhor aplicável, em virtude do tempo economizado”.

A partir dos dados presentes na seção anterior, a utilização do BOSS Tools indicou resultado positivo. Em medida de produtividade, podemos constatar, através das Tabela 7.1 e Tabela 7.2, que, embora exista um gasto de tempo inexistente na versão manual do processo, que consiste na inserção e seleção dos dados no sistema, o tempo do processo é significativamente reduzido ao utilizar a ferramenta. No pior caso, que ocorre quando é necessário inserir os dados dos jogos e das lentes durante a execução do processo, o tempo gasto diminuiu em 20%. No melhor caso, a diminuição foi de 80%. Se levarmos em conta a quantidade de pessoas envolvidas no processo, a discrepância se torna ainda maior, com uma diminuição de 184% e 196% no pior e melhor caso, respectivamente.

Essa melhoria também foi percebida pelos participantes da parte qualitativa do experimento, onde 75% dos entrevistados responderam que a ferramenta ajudou muito em relação à produtividade (nota máxima), e o restante respondeu que ajudou consideravelmente. Dentre os comentários, os usuários indicaram que a aplicação ajudou a economizar bastante tempo, especialmente na parte da geração da curva. Segue um dos comentários dos participantes: “Antes perdia-se muito tempo trabalhando na curva e estabelecendo os pontos manualmente. Com o programa teve um ganho absurdo no processo.” O fato de permitir um armazenamento e reutilização de jogos e lentes também foi elogiado.

Com relação à medida de diversidade, o resultado, embora não mesurável, é inerente da dissociação dos participantes da etapa do processo que define os valores para a curva. Isso automaticamente reduz a possibilidade do participante tender a curva de valor para algo tendencioso, seja no sentido de gosto pessoal, seja como preferências gerais. A remoção completa da influência humana, entretanto, não pode ocorrer, pois isso acarretaria em uma quantidade exagerada de curvas sem sentido. O uso de restrições, portanto, permite um bom balanceamento entre os dois. Alguns dos comentários dos participantes com relação a esse ponto foram: “As propostas geradas foram um tanto diferentes entre si, o que é algo válido e interessante. O BOSS ajudou sim a gerar resultados diferentes” e “Ajudou a gerar opções bem diferentes de ideias”.

Finalmente, com relação à usabilidade da ferramenta, alguns pontos precisam ser discutidos. Primeiramente, embora a classificação da ferramenta tenha sido, de forma geral, positiva, os participantes levantaram alguns pontos de melhoria em relação à facilidade de uso e curva de aprendizagem da ferramenta, e sugeririam funcionalidades como busca de lentes por nome e formatos diferentes de exibição das curvas. Alguns comentários foram: *“A aplicação no geral não é difícil de se usar, mas acho que ela poderia ser mais intuitiva. Seria legal condensar alguns passos para que agente possa fazer o necessário o usando o menor número de cliques possível”* e *“Colocar um sistema de busca mais eficiente. Talvez opções para a curva gerada sair em um formato de apresentação”*. Todos os participantes responderam que a aplicação ajudou no processo e que a utilizariam novamente.

7.5 Considerações Finais

Esse capítulo apresentou os experimentos realizados para validar o BOSS e a BOSS Tools. Uma entrevista sobre o processo foi realizada, assim como duas avaliações da ferramenta, uma quantitativa e uma qualitativa. O processo em si não foi avaliado novamente, devido a limitações de tempo. Entretanto, verificou-se, com base na entrevista com *designers*, que os conceitos essenciais do processo não mudaram desde sua versão inicial. Assim, acreditamos que o resultado obtido na avaliação comercial do processo se repetirá para a nova versão do mesmo.

Os resultados dos experimentos com a ferramenta mostraram-se positivos, com indícios de que a BOSS Tools auxiliou a equipe durante o processo.

8. Conclusões

Nesse trabalho, propôs-se estruturar, formalizar e evoluir um processo iniciado por duas empresas locais, e desenvolver uma ferramenta de apoio ao mesmo. O processo, chamado *Blue Ocean Strategy System* (BOSS), tem o objetivo de aumentar a inovação utilizando estratégias de negócio na fase inicial do *game design*. A versão inicial do processo foi validada no mercado com o lançamento de dois jogos, ambos anteriores ao presente trabalho.

Uma ferramenta de apoio, BOSS Tools, foi desenvolvida para auxiliar o uso desse processo. Essa aplicação foi avaliada preliminarmente utilizando experimentos descritos nas seções anteriores. Os resultados mostraram uma economia significativa de tempo, indicaram diminuição de ideias tendenciosas por parte da equipe e apresentaram uma boa avaliação de uso pelos participantes. De forma geral, consideramos os resultados desse trabalho positivos.

Ficam, então, como contribuições a estruturação e formalização do processo BOSS e a ferramenta de apoio ao mesmo, BOSS Tools. Entre as limitações encontradas, podemos ressaltar a necessidade de melhorias na usabilidade do sistema, além da necessidade de estender esse trabalho para atender os demais problemas sugeridos na Seção 6.1.

Indireta e secundariamente, este trabalho também reforça o interesse que a indústria de jogos pode ter pelos métodos de inovação que já estão sendo usado em outros setores.

8.1 Trabalhos Futuros

Ficam como trabalhos futuros evoluir o processo novamente, e a ferramenta. Primeiramente, teremos que buscar solucionar alguns, se não todos, problemas levantados na Seção 6.1. Assim, esperasse tornar o processo mais automatizado, aumentando a produtividade da equipe e melhorando seus resultados.

Ainda sobre os problemas levantados na Seção 6.1, podemos sugerir as seguintes possíveis soluções para alguns dos problemas. No problema P1, sobre definição de

mercados, a busca por tendências de estilos pode ser atacada através de mineração de dados em redes sociais e sites de jogos famosos, especialmente sites de críticas e *vlogs* de jogadores. Já a descrição de jogos provavelmente pode ser resolvida com processamento de linguagem natural associado à mineração de dados ou técnicas de agrupamento, entretanto trata-se de um problema não trivial.

O problema de criação de personas ou perfis (P2) e o problema de escolha de jogos (P3) devem seguir a mesma linha de solução de P1. Entretanto, acreditamos que a solução P3 não deve ser completamente procedural, pois a escolha de que jogos selecionar, dado um conjunto de jogos de determinado estilo, pode depender de fatores subjetivos não programáveis.

O problema de escolha das lentes (P4) pode ser atacado utilizando técnicas de agrupamento, mineração de dados ou algoritmos de otimização.

Por fim, um estudo de usabilidade também deve ser feito, de modo a melhorar a interação com a BOSS Tools, e solucionar os problemas encontrados durante os experimentos realizados.

Referências

ABOUJAFARI, M. R. et al. Study of Blue Ocean Strategy Effect on the Market Value of Listed Companies in Tehran Stock Exchange Market. **Life Science**, v. 10, n. 6s, p. 61–70, 2013.

ADAMS, E. **Fundamentals of game design**. 2nd Editio ed. Berkeley: New Riders, 2010.

ARAÚJO, M.; ROQUE, L. Uma proposta metodológica para organizar o desenvolvimento de jogos originais. **Ann. of Videojogos 2009**, p. 219–231, 2009.

BARROS, G. A. B. et al. Applying Blue Ocean Strategy to Game Design : A Path to Innovation. **Proceedings of the 12th Brazilian Symposium on Computer Games - SBGAME**, p. 267–276, 2013.

BLOMQUIST, Å.; ARVOLA, M. Personas in action: ethnography in an interaction design team. **Proceedings of the 2nd Nordic Conference on Human-computer interaction**, p. 197–200, 2002.

COOPER, A. **The inmates are running the asylum**. 2nd Editio ed. [s.l.] Sams Publishing, 2004. p. 288

CRAWFORD, C. **The Art of Computer Game Design**. Vancouver: [s.n.].

CREDIDIO, D. DE C. **Metodologia de Design aplicada à concepção de jogos digitais**. [s.l.] Federal University of Pernambuco, Recife, Brazil, 2007.

DRACHEN, B. A.; CANOSSA, A.; EL-NASR, M. S. Intro to User Analytics Data for Analytics. **Game Developer Magazine**, v. 20, n. 5, p. 21–26, 2013.

FULLERTON, T. **Game design workshop: a playcentric approach to creating innovative games**. 2nd Editio ed. San Francisco: Morgan Kaufmann, 2008.

GABLER, K. et al. **How to Prototype a Game in Under 7 Days : Tips and Tricks from 4 Grad Students Who Made Over 50 Games in 1 Semester**. Disponível em: <http://www.gamasutra.com/view/feature/2438/how_to_prototype_a_game_in_under_7_.php>. Acesso em: 5 fev. 2014.

HAGEN, U. Where do Game Design Ideas Come From? Innovation and Recycling in Games Developed in Sweden. **Proceedings of DIGRA 2009**, p. 1–11, 2009.

KIM, W.; MAUBORGNE, R. Blue ocean strategy: from theory to practice. **California Management Review**, v. 43, n. 3, p. 105–121, 2005.

KULTIMA, A.; KÖÖNIKKÄ, J.; KARVINEN, J. The Four Different Innovation Philosophies Guiding the Game Development Processes. **Games and Innovation Research Seminar**, p. 34–40, 2011.

LAUREL, B. (ED.). **Design Research: Methods and Perspectives**. Cambridge, MA: MIT Press, 2003.

LUBAN, P. **The Right Decision at the Right Time Selecting the Right Features for a New Game Project**. Disponível em: <http://www.gamasutra.com/view/feature/131443/the_right_decision_at_the_right_.php>. Acesso em: 17 jan. 2014.

MORRIS, M.; SCHINDEHUTTE, M.; ALLEN, J. The Entrepreneur's Business Model: Toward a Unified Perspective. **Journal of Business Research**, v. 58, n. 6, p. 726–735, 2005.

MUELLER, R. M.; THORING, K. Design Thinking Vs Lean Startup: A Comparison of Two Userdriven Innovation Strategies. **Proceedings of 2012 International Design Management Research Conference**, p. 151–161, 2012.

OMG, O. M. G. **Software & Systems Process Engineering Meta-Model Specification V2.0**. Disponível em: <<http://www.omg.org/spec/SPEM/2.0/>>.

OSTERWALDER, A. et al. **Business Model Generation—a Handbook for Visionaires, Game Changers, and Challengers**. [s.l.] NewYerk Wiley, 2010.

PAAVILAINEN, J. et al. **GameSpace: Methods for Design and Evaluation for Casual Mobile Multiplayer**. [s.l.: s.n.]. Disponível em: <<http://tampub.uta.fi/infim/978-951-44-7730-0.pdf>>.

RIES, E. **The Lean Startup: How today's entrepreneurs use continuous innovation to create radically successful businesses**. New York: Crown Business, 2011.

ROUSE III, R. **Game design: Theory and practice**. 2nd Editio ed. Massachusetts: Jones & Bartlett Learning, 2004.

SALEN, K.; ZIMMERMAN, E. **Rules of play: Game design fundamentals**. Cambridge, MA: MIT Press, 2014.

SHELL, J. **The Art of Game Design: A book of lenses**. Burlington, MA: Morgan Kaufmann Publishers, 2008.

TSCHANG, F. T. When Does An Idea Become an Innovation? The Role of Individual and Group Creativity in Videogame Design. **Proceedings of Druid Summer 2003 Conference: Creating, Sharing and Transferring Knowledge**, 2003.

WESLEY, D.; BARCZAK, G. **Innovation and Marketing in the Video Game Industry: Avoiding the Performance Trap**. Boston: Gower Publishing, 2010.

ZIMMERMAN, E. Play as research: the iterative design process. In: LAUREL, B. (Ed.). **Design Research: methods and perspectives**. [s.l.] MIT Press, 2003. p. 174–196.

Apêndice A

Código da classe ValueCurveCreator

```
package boss.creator;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;

import boss.BOSS4Games;
import boss.core.Lense;
import boss.core.ValueCurve;

public class ValueCurveCreator {
    public ValueCurve create(ArrayList<Lense> p1, ArrayList<Lense> s1, ArrayList<Lense> t1,
        ArrayList<ValueCurve> gamesPrimary, ArrayList<ValueCurve> gamesSecondary,
        ArrayList<ValueCurve> gamesTertiary) {

        Integer quantPAcimaMetade = 0;
        Integer quantSAcimaMetade = 0;
        Integer quantTAcimaMetade = 0;
        Integer removeuP = 0;
        Integer removeuS = 0;

        ValueCurve result = new ValueCurve();
        ArrayList<Lense> primary = new ArrayList<>();
        primary.addAll(p1);

        ArrayList<Lense> secondary = new ArrayList<>();
        secondary.addAll(s1);

        ArrayList<Lense> tertiary = new ArrayList<>();
        tertiary.addAll(t1);

        HashMap<Lense, Integer> restriction = new HashMap<Lense, Integer>();
        restriction.putAll(BOSS4Games.getInstance().getStateRestrictions());

        Random r = new Random();
        int terco = (int) ((float)restriction.size() / 3);
        int partial = (int) Math.floor((2.0/10.0)*restriction.size());

        //Verifica todas as lentes que não podem mudar
        checkNoChange(gamesPrimary, result, primary, restriction, r, quantPAcimaMetade);
        checkNoChange(gamesSecondary, result, secondary, restriction, r,
            quantSAcimaMetade);
        checkNoChange(gamesTertiary, result, tertiary, restriction, r, quantTAcimaMetade);

        //Verifica todas as lentes determinantes
        checkDeterminant(gamesPrimary, result, primary, restriction, r, quantPAcimaMetade);
        checkDeterminant(gamesSecondary, result, secondary, restriction, r,
            quantSAcimaMetade);
        checkDeterminant(gamesTertiary, result, tertiary, restriction, r, quantTAcimaMetade);
    }
}
```

```

//Verifica todas as lentes que devem aumentar
checkOnlyHigher(gamesPrimary, result, primary, restriction, r, quantPAcimaMetade);
checkOnlyHigher(gamesSecondary, result, secondary, restriction, r,
quantSAcimaMetade);
checkOnlyHigher(gamesTertiary, result, tertiary, restriction, r, quantTAcimaMetade);

//Verifica todas as lentes que devem diminuir
checkOnlyLess(gamesPrimary, result, primary, restriction, r, quantPAcimaMetade,
removeuP);
checkOnlyLess(gamesSecondary, result, secondary, restriction, r,
quantSAcimaMetade, removeuS);
checkOnlyLess(gamesTertiary, result, tertiary, restriction, r, quantTAcimaMetade,
null);

//Verifica todas as lentes que não podem ser 5
checkCantBeFive(gamesPrimary, result, primary, restriction, r, quantPAcimaMetade,
removeuP);
checkCantBeFive(gamesSecondary, result, secondary, restriction, r,
quantSAcimaMetade, removeuS);
checkCantBeFive(gamesTertiary, result, tertiary, restriction, r, quantTAcimaMetade,
null);

//Verifica todas as lentes que não podem ser 0
checkCantBeZero(gamesPrimary, result, primary, restriction, r, quantPAcimaMetade);
checkCantBeZero(gamesSecondary, result, secondary, restriction, r,
quantSAcimaMetade);
checkCantBeZero(gamesTertiary, result, tertiary, restriction, r, quantTAcimaMetade);

//Demais lentes

//Soma a quantidade de lentes que já foram utilizadas
int soma = quantPAcimaMetade + quantSAcimaMetade+ quantTAcimaMetade;

//Remove valores do mercado primário e secundário
for(int i = 0; removeuP < partial && i < primary.size(); i++) {
    int next = r.nextInt(primary.size());
    result.addLense(primary.get(next), 0);
    primary.remove(next);
    removeuP++;
}

for(int i = 0; removeuS < partial && i < secondary.size(); i++) {
    int next = r.nextInt(secondary.size());
    result.addLense(secondary.get(next), 0);
    secondary.remove(next);
    removeuS++;
}

int missingAbove = (terco-soma > 0 ? r.nextInt(terco-soma) : 0);

//Aumenta valores do mercado primário
for(int i = 0; missingAbove > 0 && i < primary.size(); i++) {
    int next = r.nextInt(primary.size());
    int value = (r.nextInt(3)) + 3 ;
    soma++;
    quantPAcimaMetade++;
    result.addLense(primary.get(next), value);
    primary.remove(next);
}

```

```

missingAbove = (terco-soma);

//Aumenta valores do mercado secundário
for(int i = 0; missingAbove > 0 && i < secondary.size(); i++) {
    int next = r.nextInt(secondary.size());
    int value = (r.nextInt(3)) + 3;
    result.addLense(secondary.get(next), value);
    secondary.remove(next);
}

//Diminui do mercado primário e secundário
for(int i = 0; i < primary.size(); i++) {
    int value = (r.nextInt(3));
    result.addLense(primary.get(i), value);
}

for(int i = 0; i < secondary.size(); i++) {
    int value = (r.nextInt(3));
    result.addLense(secondary.get(i), value);
}

//Adiciona os valores do mercado terciário
for(int i = 0; i < tertiary.size(); i++) {
    int value = (r.nextInt(5)) + 1;
    result.addLense(tertiary.get(i), value);
}

return result;
}

private void checkDeterminant(ArrayList<ValueCurve> gamesPrimary,
    ValueCurve result, ArrayList<Lense> primary,
    HashMap<Lense, Integer> restriction, Random r, Integer qAbove) {
    for(int i = 0; i < primary.size() &&
restriction.containsValue(Restrictions.DETERMINANT); i++) {
        Lense lense = primary.get(i);
        Integer aux = restriction.get(lense);

        if(aux != null && aux == Restrictions.DETERMINANT) {
            int value = r.nextInt(2) + 4;
            qAbove++;
            result.addLense(lense, value);
            restriction.remove(primary.get(i));
            primary.remove(i);
        }
    }
}

private void checkCantBeZero(ArrayList<ValueCurve> gamesPrimary,
    ValueCurve result, ArrayList<Lense> primary,
    HashMap<Lense, Integer> restriction, Random r, Integer qAbove) {
    for(int i = 0; i < primary.size() &&
restriction.containsValue(Restrictions.CANT_BE_ZERO); i++) {
        Lense lense = primary.get(i);
        Integer aux = restriction.get(lense);

        if(aux != null && aux == Restrictions.CANT_BE_ZERO) {
            int value = r.nextInt(5) + 1;
            if(value > 3)
                qAbove++;
        }
    }
}

```

```

        result.addLense(lense, value);
        restriction.remove(primary.get(i));
        primary.remove(i);
    }
}

private void checkCantBeFive(ArrayList<ValueCurve> gamesPrimary,
    ValueCurve result, ArrayList<Lense> primary,
    HashMap<Lense, Integer> restriction, Random r, Integer qAbove, Integer
removeu) {
    for(int i = 0; i < primary.size() &&
restriction.containsValue(Restrictions.CANT_BE_FIVE); i++) {
        Lense lense = primary.get(i);
        Integer aux = restriction.get(lense);

        if(aux != null && aux == Restrictions.CANT_BE_FIVE) {
            int value = r.nextInt(5);
            if(value > 3)
                qAbove++;
            else if(value == 0 && removeu != null)
                removeu++;
            result.addLense(lense, value);
            restriction.remove(primary.get(i));
            primary.remove(i);
        }
    }
}

private void checkOnlyLess(ArrayList<ValueCurve> gamesPrimary,
    ValueCurve result, ArrayList<Lense> primary,
    HashMap<Lense, Integer> restriction, Random r, Integer qAbove, Integer
removeu) {
    for(int i = 0; i < primary.size() &&
restriction.containsValue(Restrictions.CHANGE_ONLY_DOWN); i++) {
        Lense lense = primary.get(i);
        Integer aux = restriction.get(lense);

        if(aux != null && aux == Restrictions.CHANGE_ONLY_DOWN) {
            final int id = r.nextInt(gamesPrimary.size());

            int value =
r.nextInt(gamesPrimary.get(id).getValue(lense).getValue() + 1);
            if(value > 3)
                qAbove++;
            else if(value == 0 && removeu != null)
                removeu++;
            result.addLense(lense, value);
            restriction.remove(primary.get(i));
            primary.remove(i);
        }
    }
}

private void checkOnlyHigher(ArrayList<ValueCurve> gamesPrimary,
    ValueCurve result, ArrayList<Lense> primary,
    HashMap<Lense, Integer> restriction, Random r, Integer qAbove) {
    for(int i = 0; i < primary.size() &&
restriction.containsValue(Restrictions.CHANGE_ONLY_UP); i++) {
        Lense lense = primary.get(i);

```

```

        Integer aux = restriction.get(lense);

        if(aux != null && aux == Restrictions.CHANGE_ONLY_UP) {
            final int id = r.nextInt(gamesPrimary.size());

            int value = r.nextInt(5) +
gamesPrimary.get(id).getValue(lense).getValue();
            value = Math.min(value, 5);
            if(value > 3)
                qAbove++;
            result.addLense(lense, value);
            restriction.remove(primary.get(i));
            primary.remove(i);
        }
    }
}

private void checkNoChange(ArrayList<ValueCurve> gamesPrimary,
    ValueCurve result, ArrayList<Lense> primary,
    HashMap<Lense, Integer> restriction, Random r, Integer qAbove) {
    for(int i = 0; i < primary.size() &&
restriction.containsValue(Restrictions.DONT_CHANGE); i++) {
        Integer aux = restriction.get(primary.get(i));

        if(aux != null && aux == Restrictions.DONT_CHANGE) {
            int value =
gamesPrimary.get(r.nextInt(gamesPrimary.size())).getValue(primary.get(i)).getValue();
            if(value > 3)
                qAbove++;
            result.addLense(primary.get(i), value);

            restriction.remove(primary.get(i));
            primary.remove(i);
        }
    }
}
}

```

Apêndice B

Questionário

Esse questionário visa captar a sua percepção em relação ao processo BOSS e a aplicação BOSS Tools, desenvolvida durante o mestrado em Ciência da Computação de Gabriella Barros na Universidade Federal de Pernambuco.

Obrigada pela sua ajuda! Sua cooperação é muito importante.

1. Em sua opinião, quão importante é o uso de um processo na concepção de jogos?

Não importa ☒ ☐ ☐ ☐ ☐ Muito importante

2. Como você classificaria a aplicação?

Muito ruim ☐ ☐ ☐ ☐ ☐ Muito boa

3. Em sua opinião, quão difícil/não intuitiva é a aplicação?

Muito difícil ☐ ☐ ☐ ☐ ☐ Muito simples

4. Em sua opinião, BOSS Tools ajudou a melhorar a produtividade da equipe? Como?

Não ajudou ☐ ☐ ☐ ☐ ☐ Ajudou muito

5. Em sua opinião, BOSS Tools ajudou a produzir melhores resultados? Por quê?

Não ajudou ☐ ☐ ☐ ☐ ☐ Ajudou muito

6. Em geral, você acha que o BOSS Tools ajudou o processo? Por quê?

☐ Sim ☐ Não

7. Você usaria a aplicação BOSS? Por quê?

☐ Sim ☐ Não

8. Como você avalia o uso do processo BOSS com a BOSS Tools?

Inútil  Muito útil

9. Como você avalia a funcionalidade de geração automática de curvas?

Inútil  Muito útil

10. Como você avalia a funcionalidade de armazenamento de jogos?

Inútil  Muito útil

11. Como você avalia a funcionalidade de armazenamento de lentes?

Inútil  Muito útil

12. Como você avalia a funcionalidade de inserção de novos jogos?

Inútil  Muito útil

13. Como você avalia a funcionalidade de inserção de novas lentes?

Inútil  Muito útil

14. Além dessas características, você achou alguma outra interessante? Qual? Por quê?

15. Você tem sugestões de melhoria sobre a BOSS Tools?

16. Você tem reclamações sobre a BOSS Tools?

17. Qual é o seu nome?

18. Qual é o seu e-mail?

19. Qual a sua profissão?

20. Há quanto tempo trabalha com isso?

21. Há quanto tempo utiliza/conhece esse processo de concepção de jogos?
