

# Um Processo para Geração Dinâmica de Personagens Não-Jogáveis Visualmente Distintos

Rebeca Haddad

*Instituto Universidade Virtual*

*Universidade Federal do Ceará (UFC) Universidade Federal do Ceará (UFC) Universidade Federal do Ceará (UFC)*

Fortaleza, Brasil

rebecaphcc@gmail.com

Thalisson Oliveira

*GREat*

Fortaleza, Brasil

thalissonoliveira@great.ufc.br

Rute Castro

*GREat*

Fortaleza, Brasil

rute@great.ufc.br

Rossana Andrade

*Departamento de Computação*

*Universidade Federal do Ceará (UFC)*

Fortaleza, Brasil

rossana@ufc.br

Ticianne Darin

*Instituto Universidade Virtual*

*Universidade Federal do Ceará (UFC)*

Fortaleza, Brasil

ticianne@virtual.ufc.br

**Resumo**—No design de jogos, os personagens tem um papel importante em estimular níveis de identificação entre jogadores e jogo, despertando diferentes tipos e intensidades de reações emocionais e desempenhando um papel essencial na regulação da experiência do jogador. Em jogos cuja diversidade e variedade de personagens não-jogadores (NPCs) tem um papel importante para a narrativa e rejogabilidade, a criação de personagens pode tomar um longo tempo do projeto. Nesse contexto, este artigo apresenta um processo de desenvolvimento de personagens simples e replicável, que decorreu empiricamente a partir das necessidades de um projeto real. Trata-se de uma estratégia para geração dinâmica de NPCs animados e visualmente distintos entre si, a partir da combinação e colorização procedural de grupos de partes modulares individualmente ilustradas.

**Palavras-chave**—jogos, geração de personagem, animação

## I. INTRODUÇÃO

Vários aspectos dão forma a um jogo, como o universo do jogo, a narrativa, a apresentação audiovisual e a jogabilidade em si. Apesar de não serem obrigatórios, um dos elementos mais importantes para o jogo são os personagens, pois desempenham papéis específicos na história que o jogo supostamente conta. Personagens são a interface por meio da qual os jogadores experimentam a paisagem física e social do mundo do jogo [1] e, dependendo de seus objetivos e design, podem proporcionar feedback visceral, imersão cognitiva, recursos sociais e recursos da fantasia [2].

Os personagens estimulam níveis de identificação entre jogadores e jogo, despertando diferentes tipos e intensidades de reações emocionais [1]. Por isso, personagens têm um papel essencial na regulação da experiência do jogador, de acordo com o nível de engajamento que este desenvolva com os personagens [1]. Os jogadores apresentam diferentes níveis de engajamento com os diversos tipos de personagens que podem aparecer em um jogo, além do personagem-jogador ou avatar.

Os personagens podem ser classificados em quatro tipos principais [3]: *personagens de palco*, que fazem parte de

um cenário e não têm personalidade nem motivo próprios; *personagens funcionais*, semelhantes aos personagens de fundo, mas com uma função geral; *personagens de elenco*, com diferentes graus de personalidade, possuindo uma função específica no jogo relacionada à sua história; e *personagens-jogadores*, que são manipulados pelos jogadores, mas as motivações dos personagens são decididas pela história do jogo. Existem poucos trabalhos que discutem em detalhes personagens diferentes do personagem-jogador [4]. Um deles é a análise de Dan Pinchbeck (2009) de personagens não-jogadores (NPCs), que identifica um NPC como um agente que tem alguma forma de individualidade e é reconhecível como um personagem separado da população de fundo [5]. Os NPCs são cruciais para a estrutura lúdica, pois são eles que criam objetivos, dando ao jogador missões e recompensando o jogador. Assim, o jogador é capaz de atribuir significado aos NPCs e se sentem recompensados por esse significado [5].

NPCs são personagens dentro do jogo que são controlados pelo computador, e não pelo jogador, sendo comuns em diversos tipos de jogos e vindo em muitas formas e tamanhos diferentes, variando do mundano ao fantástico e ao distintamente não-humano [6]. Os NPCs podem afetar a imersão baseada em desafios sendo muito habilidosos em sua tarefa ou não habilidosos o suficiente para representar um desafio satisfatório [7]. Assim, dependendo das características do jogo, NPCs podem ter bastante relevância e demandar atenção dos artistas em sua construção. Uma parte fundamental na construção de NPCs - e de outros personagens - é a sua caracterização, que pode ser definida como tudo que é observável sobre um personagem, incluindo como eles se parecem, soam, como se mexem, como se vestem, sua inteligência, atitude e assim por diante [8].

No contexto do processo de desenvolvimento de jogos, o artista conceitual que precisa lidar com a caracterização e criação de diversos personagens jogáveis e não-jogáveis pode

acabar lidando com um grande volume de atividades e estresse. É responsabilidade deste membro da equipe definir o conteúdo emocional do jogo por meio da arte, além de criar conceitos claros e precisos dos personagens e demais *assets* visuais, em tempo hábil para implementação e - frequentemente - produzindo também as ilustrações devido a limitação de tempo e recursos.

Assim, em jogos cuja diversidade e variedade de NPCs tem um papel importante para a narrativa e para a rejogabilidade, a criação destes personagens pode ter um alto custo para o projeto – o que é ainda mais complexo no contexto de jogos *indie* e casuais, que muitas vezes precisam ser desenvolvidos em um espaço de tempo relativamente curto, por uma equipe pequena [9].

Dessa forma, o objetivo deste artigo é apresentar uma estratégia para geração dinâmica de personagens não-jogáveis visualmente distintos, inspirada pela necessidade de aumentar a rejogabilidade por meio da variedade de NPCs com características diferentes, em um projeto real de um jogo casual. A estratégia apresentada decorreu empiricamente da evolução do processo de geração de personagens no projeto, por meio da colaboração entre ilustradores e programadores, e contempla a criação de NPCs animados a partir da combinação e colorização procedural de grupos de partes modulares individualmente ilustradas. Além disso, é discutida a experiência do time com diferentes ferramentas de animação e quanto à adaptação nas etapas do projeto.

## II. REFERENCIAL TEÓRICO

### A. Conceituação e Criação de Personagens de Jogos

Em jogos, enquanto a arte conceitual é destinada aos membros da equipe de desenvolvimento do jogo a ilustração é destinada aos usuários finais [10]. Nesse sentido, arte conceitual e ilustração fazem parte do mesmo fluxo de trabalho, onde arte conceitual é simplesmente o processo de design que trata de responder a questões visuais e pode se desviar substancialmente na forma técnica e na qualidade, enquanto a ilustração é o produto final desse processo [11]. No entanto, em equipes pequenas, é comum que uma mesma pessoa assuma ambos os papéis, já que espera-se que artistas conceituais tenham uma compreensão sólida dos fundamentos da ilustração, como o uso de cor, luz e composição, técnicas de pincelada e desenho, bem como a habilidade de transmitir uma história sem palavras e retratar personagens ou ambientes dinamicamente [10].

De acordo com a pesquisa baseada em entrevistas conduzida por Julia Rässa para definir quais métodos de design são usados por profissionais da indústria de jogos para criar arte conceitual de personagens, as principais etapas seguidas pelos profissionais da indústria são, resumidamente [11]: definir, a partir de uma história pré-existente, quais conceitos visuais devem ser utilizados como base; estabelecer uma visão geral do tipo de história que se deseja contar; pesquisar e coletar referências; definir o estilo do projeto e se a história será baseada no mundo real ou fantástica; fazer esboços em escala

de cinza; e aprimorar as ideias, tentando criar um conteúdo original, que não foi feito antes.

No caso de personagens, sua caracterização se refere a tudo aquilo que pode ser percebido nele, incluindo sua aparência, movimentação, atitudes, roupas, e assim por diante [8]. Mas, um personagem é um conceito que ultrapassa sua representação gráfica, a qual tem o papel de representar visualmente sua personalidade e comportamento, de forma convincente [12] e existem diversas recomendações gráficas e conceituais para a sua criação [13]. O conceito e a caracterização de personagens em jogos podem ser segmentados. Isso significa que os personagens podem – em caso de jogos que não necessitam de personagens complexos – focar em termos visuais já que a caracterização, nesses casos, é mais relevante para encantar o jogador do que sua história particular [13].

### B. Arquétipos, Cores e Formas na Caracterização de Personagens

A definição de arquétipos, cores e formas é uma base essencial para a caracterização de personagens, pois estes elementos afetam o inconsciente e ajudam a estabelecer padrões de significação na cultura humana. Essa discussão é feita por Nieminen, o qual se debruça no estudo da teoria de Carl Jung para justificar os arquétipos defendidos por Bryan Tillman, em seu livro *Creative Character Design* [14].

De forma resumida, Jung (2014) desenvolveu a hipótese de que o ser humano é dotado de uma estrutura psíquica que contém imagens primordiais, universais e atemporais, às quais ele denomina arquétipos. Arquétipos indicam a existência de determinadas formas na psique que estão presentes em todo tempo e em todo lugar. Os arquétipos primordiais de Jung foram *self*, *shadow*, *anima* e *persona*.

Em termos de personagens, segundo Nieminen enquanto o *self*, juntando inconsciente e consciente, encaixa-se como herói da história, *shadow* representa o seu anti-herói. Já *anima* e *persona* representam, respectivamente, o interesse amoroso e personagens cuja personalidade depende da perspectiva ou cena. Isso significa que, mesmo que Jung tenha definido quatro arquétipos principais, podem existir inúmeros outros a partir destes. Assim, os arquétipos são recriados em diferentes mídias, a fim de criar histórias que envolvem o público e evoquem significados de fácil assimilação.

Nieminen também evidencia a relevância da psicologia das cores para a criação de personagens, visto que o próprio Jung as define como a língua-mãe do inconsciente. O diferencial das cores está na reação instantânea que elas evocam. Analisando outros estudos, Nieminen discute que as preferências de cores entre usuários são definidas com base em suas personalidades, possibilitando a criação de paletas que evoquem significados específicos. Na caracterização de personagens, as cores podem estimular diferentes tipos de emoções em uma audiência [15], bem como eliciar diversos significados para o público. Por exemplo, o vermelho pode simbolizar desejo, paixão e agressividade, enquanto o azul pode remeter ao pensamento, lealdade e sonho [16].

Orquestrar de forma coesa os elementos estéticos de cores e formas é importante na formação do personagem e na sua associação com o arquétipo planejado para ele Nieminen. As formas do corpo, por exemplo, estão diretamente associadas a determinados tipos de personalidade. Esses significados variam através dos principais tipos de corpos: endomorfo, mesomorfo e ectomorfo. Enquanto o corpo endomorfo geralmente comunica gentileza e bom humor, o corpo mesomorfo comunica força e heroísmo. Já o corpo ectomorfo comunica fragilidade e proteção. Isso acontece por causa das formas associadas a eles, pois quadrados (mesomorfo) comunicam estabilidade enquanto círculos (endomorfo) dão a impressão de algo inofensivo. Dessa forma, a aparência do personagem é vital para o design de jogos, já que ela fornece informações essenciais para o *gameplay* que podem, por exemplo, diferenciar heróis de vilões.

### C. Rejogabilidade e Aleatorização em Jogos Casuais

Uma visão predominante sobre jogos casuais é que estes devem ser viciantes e altamente rejogáveis [17]. Isto é, embora as sessões de jogo possam ser curtas, o design permite ao jogador reiniciar o jogo diversas vezes, de forma que este continue prendendo seu interesse. Uma das soluções utilizadas para manter o jogo novo e interessante é continuar mudando, por meio da aleatorização de certas características [18]. Nessa técnica, enquanto o jogo geral permanece o mesmo, os designers adicionam segmentos de código que tornam certas variáveis aleatórias, de forma que, cada vez que o nível é reiniciado ou o cenário é repetido, o algoritmo aleatório é reiniciado e fornece uma nova experiência para o jogador - seja em aspectos pequenos ou grandes [18]. Nesse contexto, variáveis menos importantes afetam menos o jogo e, como resultado, fornecem menor variação na experiência, e vice-versa.

A randomização pode adicionar significativa rejogabilidade a um jogo devido à capacidade de jogar várias "versões" diferentes dentro de um jogo, dando ao jogador a possibilidade de se deparar com o novo, como afirmam os autores do livro *Rules of Play: Game Design Fundamentals*: "... é crucial em um jogo que os jogadores não saibam exatamente como ele vai se desenrolar" [19]. A randomização pode ser um método bem-sucedido para adicionar capacidade de reprodução aos jogos, mantendo o interesse do jogador, pois a experiência varia ligeiramente [18].

Em jogos casuais, a aleatorização torna a repetição de parte do jogo mais interessante e menos punitiva [20]. Juul (2010) analisa que, por exemplo, em *Bejeweled* repetir o mesmo nível continuamente envolve alguma variação, pois as peças que devem ser combinadas são embaralhadas [20]. Segundo o autor, a aleatorização é uma forma de manter o interesse do jogador, mesmo quando o jogador é forçado a reproduzir parte de um jogo - o que comumente ocorre em jogos casuais. Elementos que farão o jogo divertido e empolgante, como os personagens e a história, são os pontos principais que impactam os consumidores com relação à rejogabilidade, portanto, devem ser levados em consideração pelos designers

[18]. O uso de modelos que fornecem características diferentes para a geração aleatória ou semi-aleatória de personagens é uma forma de tornar os personagens gerados menos previsíveis [21].

Uma das formas de aleatorizar a produção de assets visuais para jogos é o uso da técnica *Procedural Content Generation* (PCG) que permite, devido à sua grande escalabilidade, um o ritmo de trabalho mais rápido, se feito de maneira logística [22]. Segundo Amato (2017), mesmo que a geração de jogos "relativamente infinitos" seja uma aposta interessante, deve-se levar em conta o quanto o jogo resultante pode ocupar da memória do dispositivo, visto que dependendo do seu nível, a criação procedural pode forçar jogadores a usarem mais memória do que usariam em jogos não procedurais [22]. O autor conclui que a geração procedural de conteúdo é uma solução viável para a falta de artistas em um projeto, além de ser, também, mais barata. Ele levanta, porém, o questionamento de se esses conteúdos podem atender e superar as expectativas dos jogadores em relação ao jogo, já que artistas procuram constantemente criar elementos que emocionam e tenham significado para o jogador. Por fim, o autor destaca que é necessário que haja pesquisa para determinar o quão afetado o usuário pode ser por esse tipo de aplicação, procurando sempre estabelecer variantes de mais qualidade.

### III. TRABALHOS RELACIONADOS

Com relação à geração de NPCs, Lebron (2013) criou a ferramenta *Personage Character Generator* que, por meio de código, produz NPCs de jogos digitais ou para RPGs de mesa. Esse projeto apresenta resultados satisfatórios e um método ágil de produção. Já Firmino et al (2016) atualizaram a *Personage*, levando-a para uma tecnologia mais atual, a *Unity* [23]. Usando *Unity*, propuseram um método ágil de criação de jogos, baseado na geração procedural de NPCs, por meio da filtragem de diversos bancos de dados e aplicação de informações a personagens instanciados aleatoriamente no jogo, com características únicas. Foi também utilizada a inteligência artificial *Corsário*, uma ferramenta da *Unity* que gera informações procedurais em NPCs. O modelo se mostrou eficiente, pois com ele houve enriquecimento de personagens de maneira ágil, desde que se atente ao enredo de cada jogo, segundo os autores. No entanto, o resultado de ambos os trabalhos foca em gerar informações textuais sobre os personagens - como dados para missões e características de personagens - enquanto o processo descrito neste artigo foca em criar um banco visual de imagens, que permite a geração dinâmica de personagens de maneira semi-aleatória.

Bourke et al (2013) implementaram um algoritmo para a geração de *assets* visuais por meio de formas aleatórias e independentes proceduralmente [24]. Assim como neste artigo, o trabalho de Bourke e Shier procura implementar uma resposta visual satisfatória, trabalhando com uma geração completamente procedural de formas visuais. O algoritmo usa a perspectiva do jogador para gerar diferentes tipos de imagens e os detalhes das texturas são gerados dependendo

da proximidade do jogador de cada objeto. Ou seja: além de criar novas formas, o algoritmo também preserva a memória do dispositivo poupando detalhes que, naquela perspectiva, são desnecessários para quem os vê. Tanto a geração de imagens diferentes como a preservação do nível de detalhes obtiveram resultados positivos ao final do projeto. A estratégia proposta, no entanto, não foca em personagens e nem no trabalho conjunto entre programadores e ilustradores.

No presente trabalho, o alvo é gerar NPCs não somente de forma mais ágil, mas produzir personagens coerentes com a proposta estética do jogo, sendo fisicamente distintos e expressando diferentes características de personalidade. Dessa forma, o processo proposto se diferencia de outros na literatura por unir a produção artística com a aleatorização procedural para a criação dinâmica de NPCs com características visuais diferentes entre si.

#### IV. CONTEXTO

A técnica relatada neste artigo foi desenvolvida no contexto de um projeto de pesquisa, desenvolvimento e inovação (P&D&I), em uma parceria entre indústria e academia. O projeto descrito neste artigo ocorreu em parceria com uma empresa multinacional de eletrônicos que possui mais de 100 subsidiárias locais em todo o mundo, por meio de um projeto de Lei da Informática <sup>1</sup>

No decorrer do projeto, um dos objetivos definidos era o desenvolvimento de jogos casuais e hipercasuais, para incorporar aos celulares desenvolvidos pela empresa parceira. Dessa forma, uma pesquisadora em IHC & Jogos ficou responsável por montar os processos e orientar a produção dos jogos em equipes multidisciplinares, de acordo com as restrições da empresa, trazendo desafios técnico-científicos que pudessem ser abordados. Para o jogo em questão neste artigo, a equipe de desenvolvimento era composta por quatro pessoas nas seguintes funções: Game Designer, programador, designer de interfaces/interação e artista/ilustradora.

Essa equipe propôs e desenvolveu um jogo móvel 2D de culinária mágica, onde o jogador assume o papel de um ajudante, que mistura ingredientes e poções em seu caldeirão para preparar receitas para os clientes: monstros famintos, que apresentam diferentes valências de emoção ao chegar. Trata-se de um jogo de combinação, destreza manual e administração de tempo. Enquanto a paciência dos monstros vai caindo, o jogador precisa ser rápido em misturar os ingredientes corretos e estourar as bolhas no caldeirão que faz as misturas de ingredientes, para determinar a qualidade do prato servido e, assim, modificar o humor dos monstros.

A progressão do jogo é dividida em dias. O objetivo principal do jogador é progredir o máximo de dias possíveis, enquanto produzem pratos com a melhor qualidade possível para agradar aos diferentes monstros. Nesse contexto, a diversidade de monstros é uma chave importante para

umentar a rejogabilidade e o engajamento no decorrer dos dias representados no jogo. No entanto, no caso de personagens, esse desenvolvimento pode tomar um longo tempo do projeto. No caso deste projeto especificamente, a primeira versão do jogo deveria ser lançada em até seis meses e o projeto tinha orçamento para apenas uma pessoa que atuasse como artista e ilustradora.

Além disso, um ponto importante era que o jogo, tendo uma temática mágica, não deveria somente aproveitar-se dos estereótipos referentes a essa temática, para que os personagens não fossem uma mera reprodução cultural, mas denotassem a identidade única do produto. Esse contexto inspirou a necessidade de desenvolver uma estratégia para a geração dinâmica de personagens não-jogáveis animados e visualmente distintos entre si, a partir da combinação procedural de grupos de partes do corpo individualmente ilustradas.

Devido a uma cláusula de confidencialidade e o fato de o projeto ainda estar em andamento, não podemos divulgar os nomes reais ou mais detalhes sobre a empresa, o projeto e os jogos desenvolvidos.

#### V. METODOLOGIA

##### A. Visão geral do processo de desenvolvimento

No contexto do projeto, propusemos cinco etapas para a implementação do jogo, como uma adaptação do processo de design de jogos casuais proposto por Tefry (2010) [25], para o contexto do projeto: geração de ideias, conceituação, prototipação, playtesting e pré-lançamento. De forma resumida, na primeira etapa, eram executadas técnicas de geração de ideias como *brainstorming*, *brainwriting* e gerados documentos descrevendo uma versão simplificada de Game Design Canva de diferentes ideias, as quais eram apresentadas para a empresa parceira.

Após a definição de qual ideia seria desenvolvida, seguia-se a segunda etapa, em que o game design canva era ajustado e detalhado. Além disso, era desenvolvida a backstory do jogo, definidos e listados os possíveis personagens e coletáveis, junto com seus primeiros rascunhos e criado o plano de *level design*. Mais uma vez, todos estes itens eram apresentados à empresa parceira para aprovação. Após estarem acertados os aspectos de *gameplay* e progressão, era criado um cronograma de prototipação, que guiaria a etapa seguinte.

Na terceira etapa, eram desenvolvidos iterativamente protótipos da mecânica principal, que gradativamente eram acrescidos de aspectos adicionais do *gameplay*, interface e aspectos visuais. Também nesta etapa, eram finalizadas as *concept arts* dos personagens, itens e cenários e realizados ajustes e balanceamentos.

A etapa de playtesting ocorria simultaneamente com a de prototipação, em ciclos iterativos, permitindo a evolução dos protótipos, a identificação de problemas e de oportunidades para enriquecer o *gameplay* e a geração de insumos para ajustes de jogabilidade e interface. Finalmente, a etapa de pré-lançamento focava na divulgação e preparação de material para lançamento do jogo na Google Play, incluindo a criação e

<sup>1</sup>A lei da Informática está descrita no Decreto 5.906/06, Lei nº 10.176/01, Lei nº 13.674/18 e Lei nº 13.969/19 e é uma lei que concede incentivos fiscais para empresas do setor de tecnologia (áreas de hardware e automação), que tenham por prática investir em Pesquisa e Desenvolvimento.

alimentação da página do jogo na loja e a produção de banners e vídeos de divulgação.

### B. Evolução do Processo de Criação de Personagens

No contexto do processo de desenvolvimento, a criação dos personagens não-jogáveis (i.e., monstros que fariam pedidos de comidas mágicas) acontecia nas etapas de conceituação e prototipação/playtest. O processo proposto neste artigo para a geração de personagens decorreu empiricamente das fases do processo de criação de personagens. À medida que se evoluía no processo e se analisava a viabilidade, o impacto e a jogabilidade dos protótipos nos *playtests*, o processo de geração de personagens foi definido.

Durante todas as etapas da evolução da abordagem, sempre se considerou a necessidade de que a combinação de cores e formas remetesse a diferentes arquétipos e emoções na criação dos personagens. Nesse sentido, inicialmente, os personagens foram individualmente produzidos, para que fossem únicos e marcantes em suas personalidades e visual. No decorrer do processo, detalhado a seguir, as características de história e informações sobre os personagens se tornaram irrelevantes para o *gameplay*, enquanto suas características físicas deviam denotar variedade de identidade. Os princípios do descritos no livro *Creative Character Design* de Bryan Tillman foi utilizado como direcionamento na criação dos novos personagens e em como passar a mensagem desejada com eles.

Assim, as etapas seguiram-se visando a produção ágil de personagens por meio da combinação de elementos conceituais, como arquétipos, e estéticos, como a cor e as formas, para a criação de um banco de partes do corpo dos personagens.

1) *Fase 1: Uso de animação quadro-a-quadro*: Tendo sido definido que os personagens seriam um gatinho mágico cozinheiro e um conjunto de monstros clientes com diferentes personalidades, foram esboçadas ideias, nomes e conceitos. Inicialmente, o objetivo era que os monstros fossem conhecidos do gato e que cada um tivesse uma personalidade única. Os primeiros monstros pensados (Nox e Fredo) tinham identidades particulares e eram bastante diferentes entre si: Nox deveria remeter à tecnologia e natureza, com uma postura séria e misteriosa, enquanto Fredo deveria ter um tom humorístico e atrapalhado, contrapondo a seriedade de Nox. Após a definição da paleta de cor - feita com base em uma pesquisa de usabilidade e acessibilidade - os primeiros *sprites* para estes personagens foram produzidos, visando uma animação quadro-a-quadro (Fig. 1). Essa técnica foi definida, principalmente, devido à experiência prévia da artista. Nesse ponto, os *playtests* demonstravam ótima aceitação dos personagens e os monstros estavam de acordo com o Game Design Canva, recebendo aprovação dos *Product Owners*.

No entanto, durante o ciclo de produção se identificou que a animação dos personagens seria problemática, devido à estética escolhida – *stencil*, uma técnica na qual existe um espaço entre a *lineart* e a cor das imagens – a qual demandaria tempo e trabalho redobrado. Isso se deveu ao



Fig. 1. Primeiros sprites dos personagens, inicialmente com personalidades únicas e ilustração individual

fato de que seria necessário animar tanto a camada de traçado do desenho quanto a camada de pintura intercalando, a cada quadro, de uma camada para a outra. Quando a estética de *stencil* foi aprovada, não havia sido identificado o trabalho duplo que traria para as animações, o que era particularmente problemático dado que havia apenas uma estagiária na equipe para trabalhar com animação.

Outra situação que demandou mudança no processo de animação foi afetada, também, ao trabalho remoto devido à pandemia da Covid-19 durante os anos de 2020 e 2021. Apesar de a artista utilizar em sua casa um computador que atendia a todos os requisitos do Adobe Animate, a ferramenta não suportava a utilização do traçado escolhido para a arte (traço estilo carvão, que era material próprio da Adobe). Mesmo com as tentativas remotas de resolver o problema, devido aos constantes *bugs*, travamentos e a lentidão do programa ao usar os modelos criados, o Adobe Animate foi descartado como ferramenta neste momento – o que também afetou o processo de animação.

2) *Fase 2: Mudança para animação Skeletal*: Aproveitando o momento de mudanças, se decidiu mudar também a técnica de animação, para modularizar os personagens, priorizando a produção de uma variedade de monstros de forma mais eficiente. A técnica quadro-a-quadro havia sido selecionada porque o jogo tinha uma característica de organicidade em sua estética. Assim, o alvo era ter mais liberdade em relação à deformação das formas e dos movimentos dos personagens, controlando a fluidez da animação com base na perspectiva que o público teria. No entanto, para conferir mais rapidez ao processo de animação dos personagens atuais e os próximos que seriam criados, a equipe optou por utilizar a técnica de *“skeletal animation”*. Esta técnica consiste na construção de um esqueleto articulável para um personagem e o uso de uma hierarquia de partes interconectadas, popularmente chamadas de *“bones”*. Para dividir o corpo do personagem e obter os resultados mais naturais e satisfatórios possíveis, foi utilizado como base o livro *Figure Drawing For All It's Worth* de Loomis, para fundamentar as definições de articulações e anatomia.

Em seguida, foi feita uma análise de ferramentas que dessem suporte a esta técnica e foi selecionado o Character Animator, porque facilitaria a transposição das artes feitas no

Illustrator facilmente para o esqueleto do personagem que seria montado no programa, por meio da importação do personagem com reconhecimento das camadas e hierarquização de partes apropriada.

O processo de criação dos personagens passou a ser, resumidamente: (i) definir as emoções que o personagem teria, (ii) no Illustrator, criar bocas e olhos para cada emoção, dividindo-o em camadas coerentes, (iii) definir e criar N instâncias para olhos, como “piscar”, “arregalar” (Fig. 2), “franzir”; e o mesmo para as bocas (Fig. 3), (iv) dividir as articulações do corpo, como braço superior, braço inferior, torso, etc. Após separar em camadas detalhando os nomes das partes (e.g.: boca, olho esquerdo, olho direito, nariz) as partes eram separadas em subcamadas, com os diferentes estados que a boca e os olhos poderiam assumir. Após completar a definição do personagem, este era exportado do Illustrator para o Character Animator, que interpreta cada nome de camada como instâncias e separa as partes do corpo em um esqueleto padrão. Depois disso, cabia ao ilustrador alterar esse esqueleto e fazer animações e movimentos como ele.

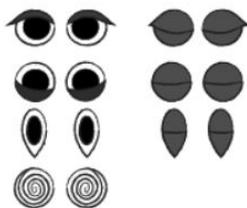


Fig. 2. Exemplo de modularização de olhos e suas instâncias



Fig. 3. Exemplo de modularização de bocas e suas instâncias

### 3) Fase 3: Animação procedural das partes modulares:

Apesar de as mudanças introduzidas até então, para animar os monstros ainda seria necessário criar um esqueleto para cada e, posteriormente, repetir esse processo sempre que um novo monstro fosse criado. Já que cada monstro tinha um corpo específico, com diferentes bocas e olhos, além de diferentes formas de falar e se portar, em termos práticos seria preciso um arquivo de marionete de cada monstro com instâncias diferentes. Assim, para viabilizar a produção rápida e modular dos monstros, mantendo suas identidades particulares, era necessário mais eficiência não somente na produção dos personagens, mas também geração das animações.

Tomando por base a técnica de animação *Skeletal*, decidiu-se por criar previamente o mínimo de esqueletos possíveis e ter um banco de imagens constituído pelas partes modulares dos personagens, que servisse como base

para a montagem e animação destes por meio da Unity. Dessa forma, a pessoa responsável pela ilustração ficava responsável por criar uma quantidade pré-definida de corpos e desenhar diversas feições e características que pudessem ser semi-aleatorizadas proceduralmente, incluindo: bocas, narizes, olhos, cabelos, chifres e acessórios, como roupas e chapéus. As partes do rosto e suas variações eram criadas tendo em vista diferentes expressões e arquétipos.

Em termos de ferramentas, fez-se necessário apenas o Adobe Illustrator para desenhar as partes modulares dos personagens e a plataforma Unity para montá-los, colorí-los e animá-los. Para manter a coerência dos personagens, bem como a identidade única que era requisito do jogo, foram criadas regras lógicas restringindo as combinações de certas partes do corpo com determinadas roupas e cores, criando padrões com base em arquétipos gerais que o monstro poderia incorporar. Exemplos de arquétipos utilizados como base foram: herói, rebelde, sábio e homem comum.

Finalmente, o processo desenvolvido – o qual é apresentado na seção VI – dava suporte à geração dinâmica de personagens não-jogáveis por meio da combinação semi-aleatória de grupos de partes do corpo proceduralmente animadas, como exemplifica a Fig. 4.

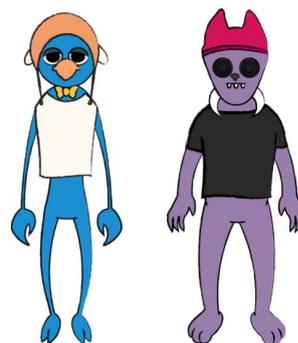


Fig. 4. Exemplo de personagens gerados dinamicamente, cujos elementos gráficos remetem, respectivamente, ao arquétipo de homem comum e rebelde. Olhos, bocas, narizes, roupas e acessórios são partes modulares e foram combinadas de forma semi-aleatória e posicionadas proceduralmente sobre os corpos utilizando Unity. Após montado o personagem, cada elemento foi colorido seguindo regras definidas por meio de código.

## VI. PROCESSO PARA GERAÇÃO DINÂMICA DE PERSONAGENS NÃO-JOGÁVEIS

Os personagens não-jogáveis gerados por este processo são formados por partes modulares individualmente ilustradas. No caso do projeto em que foi aplicado, tratavam-se de: cabeças, cabelos, olhos, narizes, bocas, orelhas, chifres, corpos, asas e acessórios/roupas, como camisas, gravatas, chapéus e óculos. Cada parte modular é sorteada e montada proceduralmente em um personagem antes de ele aparecer em cena. Algumas partes podem influenciar na seleção de outras, por exemplo, um cabelo, que exige que o personagem não use chapéu, ou uma boca que exige que não haja nariz. As partes modulares são coloridas proceduralmente de acordo com uma paleta de cores

pré-definida, após serem montadas na posição em que devem aparecer para formar o personagem. Todos os personagens possuem animações em comum, a animação de entrada, a animação de saída e a animação de ações específicas.

#### A. Etapas do Processo

Após o refinamento da abordagem de produção de personagens refinada ao longo do projeto, o processo final obtido permitiu a geração de uma grande quantidade de personagens distintos, proporcional à combinação de partes modulares criadas. O processo proposto foi aplicado em um projeto real, conforme descrito na Seção IV, sendo utilizado para a geração de todos os personagens do jogo, porém foi pensado de forma a permitir o reuso em diferentes contextos e tipos de jogos, de acordo com as etapas duas etapas listadas a seguir.

##### 1) Criação de Partes Modulares

- a) Definir os arquétipos de personagens que serão utilizados no jogo, com base no Game Design Canva (ex.: sábio, mago, explorador, criador, herói, rebelde, amante);
- b) Definir a paleta de cores e estilo estético a ser seguido no jogo como um todo e, particularmente, para personagens;
- c) Definir quais partes modulares comporão os personagens (ex.: olho, boca, chifres, asas, camisa, arma);
- d) Definir que instâncias as partes modulares terão quando animadas (ex.: para cada olho, uma instância dele piscando; para cada boca, instâncias expressando felicidade e raiva);
- e) Definir estados que os personagens assumirão durante a animação (ex.: idle, andando, pulando);
- f) Criar esboços das partes modulares independentes e suas instâncias e variações, de acordo com todas as definições anteriores, para discussão e validação com a equipe;
- g) Testar tamanhos e combinações das partes entre si;
- h) Ajustar e arte-finalizar as partes modulares selecionadas (sem a aplicação de cores).

##### 2) Geração de Personagens

- a) Importar, organizar e hierarquizar as partes modulares para a Unity;
- b) Definir as posições que cada parte modular pode assumir na hierarquia (esqueleto), por exemplo: cada torso pode assumir uma posição diferente em cena, dependendo da cabeça sorteada;
- c) Implementar a função de criação semi-aleatória de personagens, incluindo:
  - regras para seleção e randomização das partes modulares
  - regras sobre que partes devem ser combinadas entre si, em quais estados
  - regras para coloração das partes
  - regras para posicionamento das partes

- regras para ocorrência de animações

Nesse processo, os princípios apresentados por Bryan Tillman [14] para o design criativo de personagens foram utilizados para direcionar a criação de novos personagens e a forma de passar a mensagem desejada por meio deles. Para que os personagens não possam ser rotulados por uma única característica e que, ao mesmo tempo, não sejam formados por partes que não “conversem” entre si, é necessário definir regras após a criação das ilustrações. Dessa forma, mesmo que diferentes, cada característica do personagem é harmônica com as demais.

Durante a criação do banco de partes modulares para personagens, para que o processo seja fluido, é essencial que os seguintes cuidados sejam observados:

- Cada prancheta deve representar uma classe da hierarquia e suas instâncias;
- As camadas devem ser nomeadas de forma que possam ser facilmente reconhecidas pelo programador;
- Ilustradores e programadores devem compartilhar um repositório com controle de versão;
- Ilustradores e programadores devem testar conjuntamente as regras definidas para a geração e posicionamento das partes modulares, de forma que problemas possam ser identificados e corrigidos imediatamente.

A qualquer momento, é possível acrescentar novas partes modulares para compor os personagens, sem retrabalho. Assim que a fase é iniciada, o código cria um personagem com uma combinação semi-aleatória de partes organizadas conjuntamente por artistas e programadores para fazer sentido visualmente. Nesse cenário, o programador assume o papel de animador dos personagens por meio da própria Unity.

#### B. Colorização

As regras de colorização utilizadas no projeto descrito neste artigo são apresentadas a seguir, para ilustrar que tipos de decisões são necessárias para colorir personagens gerados por meio da aplicação do processo proposto. No entanto, é importante ressaltar que cabe à equipe analisar se estas regras são adequadas ao jogo em desenvolvimento, ao utilizar este processo.

Após definir a paleta de cores do jogo, é importante identificar as tonalidades das matizes frias e quentes que serão utilizadas nos personagens, sempre acompanhadas de cores de alto contraste. A partir daí, as regras de combinação de cores definidas devem levar em consideração a comunicação de ideias e características. Existe uma grande flexibilidade ao utilizá-las e inúmeras combinações baseadas na psicologia das cores, casado com a escolha de formas e personalidades. De forma simplificada, por exemplo, um personagem com traço raivoso pode utilizar formas angulares e cor de pele ou aura avermelhada ou amarelada; enquanto um que se pareça tímido e dócil pode utilizar peças de roupa e acessórios com predominância de tons pastéis azuis e rosados e formas mais arredondas, e assim por diante.

Após ser definida a paleta de cores do jogo e personagens, para cada personagem a ser gerado:

1) **Definir a cor base:** Selecionar uma dentre as cores definidas para a pele na paleta do jogo. Esta será a cor base do torso, membros e cabeça do personagem. No jogo em que o processo foi desenvolvido, inicialmente, foram utilizadas apenas cores frias como cor base, como mostra o exemplo da Fig. 5-A, em que foi selecionado o azul. Posteriormente, identificou-se que tanto cores quentes quanto frias poderiam ser escolhidas sem prejuízo para o contraste entre o personagem, cenário e interface. Recomenda-se que o programador faça com que tons adequados de azul e vermelho sejam mais comuns na definição da cor base, pois costumam fazer parte das harmonias favoritas das pessoas.

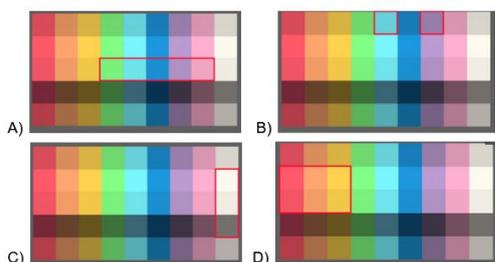


Fig. 5. Exemplo de escolha de cores para a colorização das partes modulares dos personagens, de acordo com as regras definidas

- 2) **Definir a cor dos detalhes interiores:** Selecionar as cores harmônicas à cor base que ofereçam bom contraste para as partes do corpo que serão sobrepostas ao corpo e cabeça. No jogo em questão, inicialmente, eram selecionadas as cores de tonalidade escura análogas à cor base (Fig. 5-B). No jogo em questão, posteriormente, foi identificado que a melhor combinação para os detalhes interiores era a segunda cor mais escura da paleta, na mesma matiz da cor base, a qual pode ser aplicada em detalhes como pálpebras, e opcionalmente, uma cor complementar à cor base para outros detalhes, como a boca.
- 3) **Definir a cor dos detalhes exteriores:** Selecionar uma cor de alto contraste para colorir detalhes do personagem que não estarão sobrepostos à cor base, como chifres e asas. Na Fig. 5-C, ilustra-se a escolha de preto e branco para este passo.
- 4) **Definir a cor dos acessórios e roupas:** Selecionar uma cor complementar à cor base para o(s) acessório(s) ou roupa(s) principal(is) e, opcionalmente, uma cor análoga para os demais acessórios e/ou roupas. Se necessário, podem ser aplicadas ainda variações na tonalidade das cores selecionadas, caso haja muitos acessórios e peças de roupa. Inicialmente, no jogo em questão, uma vez que a cor de base era sempre tons frios, acessórios e roupas seriam coloridos com quaisquer das cores quentes da paleta (Fig. 5-D). Posteriormente, os testes demonstraram que a escolha por cores complementares e análogas conferiam maior variedade e harmonia para

os personagens.

### C. Exemplo de Implementação do Processo

A Unity permite a criação de cenas, nas quais podem ser adicionados, posicionados e configurados objetos que compõem o jogo, os *game objects*. Alguns exemplos de *game objects* são formas geométricas, imagens, objetos 3D e textos. Além de posicionar e configurar objetos, a Unity permite alterar as propriedades desses objetos através de *scripts* que são escritos usando a linguagem de programação C#.

De forma resumida, para a geração dinâmica de personagens, é criado um *game object* para o personagem, composto por um conjunto de outros *game objects* que representavam as suas partes modulares. Para permitir que a cada início de partida fosse renderizado um personagem diferente, foi criado um *script* que recebia como parâmetro conjuntos de imagens para cada parte modularizada e, posteriormente, realizava a seleção semi-aleatória de um exemplar do conjunto para cada parte, segundo os sub-conjuntos de *game objects* que podiam ser combinados entre si. A Fig. 6 apresenta um trecho do código C# para a seleção aleatória de uma das partes do monstro, enquanto a Fig. 7 demonstra o produto final da produção dos personagens utilizando o processo e algoritmo descritos nesta seção, em um trecho da interface final do jogo.

```

83 // Script de Unity | 13 referências
84 public class AssembleParts : MonoBehaviour
85 {
86     13 referências
87     public static GameObject InstantiatePart(PackOfParts partList, RectTransform rt)
88     {
89         GameObject part = null;
90         if (partList != null && partList.allParts.Length > 0)
91         {
92             GameObject randomPart = partList.allParts[Random.Range(0, partList.allParts.Length)];
93             if (rt.transform.childCount > 0)
94             {
95                 foreach (RectTransform rect in rt)
96                 {
97                     Destroy(rect.gameObject);
98                 }
99             }
100             part = Instantiate(randomPart, rt);
101         }
102         return part;
103     }

```

Fig. 6. Fragmento do código de seleção aleatória de uma parte do monstro



Fig. 7. Exemplo de personagens gerados, coloridos e posicionados na interface do jogo, utilizando a implementação do processo descrito.

Na Fig. 6, o método `InstantiatePart(PackOfParts partList, RectTransform rt)` possui como primeiro parâmetro o `PackOfParts partList`, que é um conjunto com as diferentes versões de uma parte do personagem, por exemplo, os olhos. Já o segundo parâmetro, o `RectTransform rt`, representa o objeto, com sua respectiva posição, dentro do qual vai ser inserida a parte selecionada. A linha 90 do código na figura mostra que é selecionado apenas um elemento de dentro de `partList`, de maneira aleatória, o que é garantido pela execução da instrução `Random.Range`.

Devido a uma cláusula de confidencialidade, não podemos divulgar o algoritmo completo da solução.

## VII. DISCUSSÃO

### A. Ferramentas testadas durante a evolução do processo

Como descrito na seção V-B, durante a evolução do processo de geração de personagens, à medida que se decidida que técnica seria adequada para concepção e animação dos personagens, diversas ferramentas foram testadas pela equipe, a cada fase. Nesta seção, são resumidas as percepções obtidas das ferramentas que foram testadas, de acordo com o contexto anteriormente descrito.

As primeiras ferramentas testadas se referiam à técnica de animação quadro-a-quadro. Essa técnica é útil quando se deseja aplicar de forma mais orgânica os princípios básicos da animação. Algumas ferramentas comumente utilizadas para essa técnica são Adobe Animate, OpenToonz, ToonBoom Harmony. Já a técnica de animação *skeletal* tem boa compatibilidade com o processo do mercado de jogos casuais, visto que você pode reutilizar o mesmo esqueleto em inúmeras cenas e contextos diferentes e pode ser criada com ferramentas como DragonBones, Character Animator e ToonBoom Harmony. A equipe teve a oportunidade de examinar, mais detalhadamente, a Adobe Animate, a Character Animator e a DragonBones contando, além da estagiária do projeto, com a perspectiva de colaboradores mais experientes com animação que foram consultados durante o processo.

A Adobe Animate foi considerada intuitiva e fácil de usar, exigindo pouco tempo de adaptação. O processo de animação apoiado por esta ferramenta foi simples, mas sua integração com outros programas Adobe apresentou *bugs*, até mesmo em máquinas compatíveis. Por exemplo, o traçado de “carvão estreitado” utilizado na arte, que veio disponível com o Adobe Illustrator, no conjunto de pincéis extra, travava a Animate sempre que era aplicado aos sprites. Nos piores casos, o programa fechava repentinamente e era necessário reiniciá-lo, após perder qualquer progresso feito. Animadores experientes podem sentir falta de uma perspectiva mais sistemática, com mais opções de funcionalidades para configurar as mais diversas características da animação.

A Adobe Character Animator é especificamente voltada para a criação de *skeletal animations* e também permite integração com o pacote Adobe, seus pincéis e formatos. Apesar da interface enxuta e voltada para seu objetivo, foi percebida como relativamente complexa de usar, por alguém com pouca experiência. Obter bons resultados com a ferramenta exigiu

alguns testes, visto que ela oferece muitas funcionalidades e é preciso se ater aos pequenos detalhes de cada opção. Um dos aspectos mais destacados da Character Animator foi a funcionalidade de *webcam*, que permite o controle do rosto do personagem com base no rosto do ilustrador, servindo como molde para cada expressão do personagem. Character Animator foi considerado o mais agradável, facilitando a transposição das artes feitas no Illustrator facilmente para o esqueleto do personagem que seria montado no programa.

A Dragon Bones é voltada para a criação e animação de *sprites* para jogos e oferece uma interface sucinta e objetiva, mas não foi considerada intuitiva, exigindo um processo de ambientação e treino. Já a plataforma Unity, apesar de não ser própria para arte e animação, mas para programação e prototipação de jogos, oferece diversos recursos e plugins que permite animar personagens e outros *assets* do jogo. A plataforma oferece recursos diversos para controle da animação, mas deixa a animação mais a cargo do programador do que do artista, exigindo que ambos trabalhem conjuntamente para melhores resultados.

### B. Necessidade de flexibilização do processo para o contexto e realidade do time

No contexto relatado neste artigo, a temática definida nas fases de geração de ideias e conceituação foi, desde o início, uma preocupação do ponto de vista artístico. Seria um desafio reforçar a temática de monstros em uma cozinha mágica de maneira satisfatória, visto toda a significação estética que existe no senso comum acerca desse tema. Além disso, o objetivo de desenvolver um jogo casual em cerca de seis meses, com métricas e mecânicas simples e tendo apenas uma estagiária artista representavam desafios importantes nesse cenário.

Ora, no jogo desejamos transpor uma nova realidade, proporcionar uma vivência exterior ao indivíduo. Em termos artísticos, o desafio enfrentado foi casar a estética dos personagens com as regras desse mundo – ao mesmo tempo em que considerando as limitações do projeto, a audiência que o produto procurava atingir, o objetivo desses personagens, como esses personagens reforçam as funcionalidades do jogo e o que esses personagens podem significar para o jogador.

Dessa forma, a logística foi um ponto-chave para a evolução do processo de criação de personagens. A cada etapa, a equipe analisava o que era necessário fazer, o que era possível podemos fazer e a melhor forma de fazê-lo. Por isso, mesmo que a animação de personagens articulados fosse uma solução interessante para substituir o processo quadro-a-quadro do Adobe Animate, ainda havia problemas, como a falta de experiência da ilustradora com os programas identificados, bem como com *skeletal animation* em si. Assim, o processo proposto decorreu de uma série de decisões de design que, em termos de logística, se mostraram a melhor estratégia para este caso, possibilitando prazos e métricas realistas, e a criação de jogos com uma equipe pequena e uma ilustradora pouco experiente. Dessa forma, chegamos a um processo passível de ser replicado para o desenvolvimento de personagens dos

próximos jogos do projeto e que conferia à equipe estabilidade e conforto.

A consciência e adaptação de acordo com o contexto e necessidade do projeto foram priorizados pela equipe, visando evitar um processo de desenvolvimento que não atendesse às expectativas do cliente e ainda trouxesse prejuízos à saúde mental de seus criadores, artistas e programadores. A equipe constantemente discutia suas limitações entre todos, o que manteve a equipe focada, unida e otimista em busca de soluções melhores. A decisão de que o programador seria responsável por criar as animações na Unity, enquanto a ilustradora desenvolveria um banco de partes de corpo que poderia ser alimentado ilimitadamente por futuros artistas do projeto e que ambos colaborariam com a pesquisadora do projeto na definição das regras foi fruto da flexibilização no processo. Essa flexibilização e adaptação permitiu atender aos requisitos do cliente, enquanto a equipe procurava a maneira mais fluida de resolver os problemas que surgiam.

Assim, o argumento apresentado neste artigo não é que o processo proposto seja necessariamente mais eficiente ou mais adequado do que outros. Mas, que se trata de um processo de desenvolvimento de personagens simples e replicável, que foi utilizado com sucesso em uma equipe pequena, com uma artista pouco experiente, para o desenvolvimento de jogos casuais com uma grande quantidade de NPCs, aumentando sua rejogabilidade.

### VIII. CONCLUSÃO

Personagens são importantes para reforçar a temática de um jogo, carregando em si parte da significação estética da ambientação que se deseja que um jogo comunique, seu espírito e originalidade. Por outro lado, nem sempre os projetos dispõem dos recursos de tempo e pessoas para produzir conceitos, ilustrações e modelos individualizados para cada personagem, especialmente NPCs. A abordagem apresentada neste artigo permite que equipes pequenas possam produzir NPCs visualmente diferentes, a partir de um trabalho conjunto entre ilustrador e programador. Os resultados obtidos indicam que o processo tem o potencial de gerar, de forma simples e em um curto período de tempo, uma grande quantidade de NPCs visualmente satisfatórios e distintos. Os trabalhos futuros incluem o refinamento das regras e o teste do processo em outros tipos de personagens. A abordagem é uma ideia útil para outros times de desenvolvimento de jogos que podem aplicá-la, evolui-la e testar sua adequabilidade com diferentes contextos e requisitos.

### AGRADECIMENTOS

Agradecemos o financiamento do projeto LGE Tools & CAS com administração da FCPC. Rossana Andrade é bolsista de produtividade CNPq DT-2.

### REFERÊNCIAS

- [1] P. Lankoski, "Player character engagement in computer games," *Games and Culture*, vol. 6, no. 4, pp. 291–311, 2011.
- [2] K. Isbister, *Better game characters by design: A psychological approach*. Elsevier/Morgan Kaufmann, 2006.
- [3] J. H. Smith, S. P. Tosca, and S. Egenfeldt-Nielsen, *Understanding video games: the essential introduction*. Routledge, 2008.
- [4] J. Blom, "The dynamic game character: Definition, construction, and challenges in character ecology," Ph.D. dissertation, IT University of Copenhagen, Department of Digital Design, 2020.
- [5] D. Pinchbeck, "An analysis of persistent non-player characters in the first-person gaming genre 1998-2007: a case for the fusion of mechanics and diegetics," *Eludamos: Journal of Computer Game Cultures*, vol. 3, no. 2, pp. 261–279, 2009.
- [6] H. Warpefelt, "The non-player character: Exploring the believability of npc presentation and behavior," Ph.D. dissertation, Department of Computer and Systems Sciences, Stockholm University, 2016.
- [7] L. Ermi and F. Mäyrä, "Fundamental components of the gameplay experience: Analysing immersion," 2005.
- [8] S. Meretzky, "Building character: An analysis of character creation," *Gamasutra*. URL= [http://www.gamasutra.com/resource\\_guide/20011119/meretz\\_ky\\_01.htm](http://www.gamasutra.com/resource_guide/20011119/meretz_ky_01.htm), 2001.
- [9] T. Darin and N. Carneiro, "Desafios na avaliação de fatores humanos na interação com jogos digitais," in *XIX Simpósio Brasileiro de Games e Entretenimento Digital (SBGames)*, 2020.
- [10] E. Lilly, *The Big Bad World of Concept Art for Video Games: An Insider's Guide for Students*, 1st ed. Design Studio Press, 2015.
- [11] J. Rässa, "Concept art creation methodologies: Visual development of "rock boy"," 2018.
- [12] P. Blair, *Cartoon Animation*. Walter Foster Publishing, 1995.
- [13] A. Rollings and E. Adams, *Andrew Rollings and Ernest Adams on game design*. New Riders, 2003.
- [14] B. Tillman, *Creative character design*. Routledge, 2012.
- [15] E. Joosten, G. v. Lankveld, and P. Spronck, "Colors and emotions in video games," in *11th International Conference on Intelligent Games and Simulation GAME-ON*. sn, 2010, pp. 61–65.
- [16] M. Medeiros Filho, A. Karlota, V. F. Martins, P. R. L. Pinheiro, and A. M. M. Neves, "Games, cores e personagens: Uma análise da relação cromática em jogos digitais clássicos," in *XIV Simpósio Brasileiro de Games e Entretenimento Digital (SBGames)*, 2015.
- [17] J. Tams, "Online casual games q&a," *Casual Connect Magazine*, 2006.
- [18] T. Frattesi, D. Griesbach, J. Leith, T. Shaffer, and J. DeWinter, "Replayability of video games," *IQP, Worcester Polytechnic Institute, Worcester*, 2011.
- [19] K. Salen, K. S. Tekinbaş, and E. Zimmerman, *Rules of play: Game design fundamentals*. MIT press, 2004.
- [20] J. Juul, "In search of lost time: on game goals and failure costs," in *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, 2010, pp. 86–91.
- [21] J. Andersson, L. Bergman, E. Hildinge, J. Iversen, and D. Johansson, "A video game using procedural animation," B.S. thesis, 2017.
- [22] A. Amato, "Procedural content generation in the game industry," in *Game Dynamics*. Springer, 2017, pp. 15–25.
- [23] E. Firmino, J. Pagini, and M. Henrique, "Geração procedural de histórias: o uso de npcs procedurais para complementar o jogo," in *XV Simpósio Brasileiro de Games e Entretenimento Digital SBGames*, 2016.
- [24] P. Bourke and J. Shier, "Space filling: A new algorithm for procedural creation of game assets," in *Proceedings of the 5th annual international conference on computer games multimedia & allied technology*. CGAT In Press, 2013.
- [25] G. Trefry, *Casual game design: designing play for the gamer in all of us*. CRC Press, 2010.