

Game Design Document (GDD) - Gala Of The Hidden (GotH)

Versão: 0.1 (MVP Draft) **Data:** 09 de Julho de 2025 **Autor:** João Alfredo (ExperienceFactory)

1. Visão Geral Executiva

1.1. Título do Projeto: Gala Of The Hidden (GotH)

1.2. Gênero: MMORPG Sandbox Gótico de Horror Pessoal/Social

****1.3. Plataformas:** PC (Windows) como plataforma inicial, com potencial para expansão futura.

Adendo Estratégico para o MVP: Para a fase de Produto Mínimo Viável (MVP), o jogo será desenvolvido como uma aplicação web de alta performance com visualização estilizada em ASCII/texto, renderizada em uma grade 2.5D (com suporte a múltiplos níveis de altura, similar a Dwarf Fortress). O frontend será construído com Nuxt.js (Vue 3) e a biblioteca de componentes PrimeVue, garantindo máxima acessibilidade via navegador, iteração de desenvolvimento rápida e foco total na validação dos sistemas de simulação. A transição para uma engine de jogo completa como a Unity 3D está planejada para fases pós-MVP, utilizando os sistemas de backend e a lógica de jogo já validados..

1.4. Engine: Unity 3D, utilizando o Universal Render Pipeline (URP) para otimização e flexibilidade visual.

1.5. Backend: SpacetimeDB, uma plataforma de banco de dados e servidor de jogo em tempo real, otimizada para jogos multiplayer massivos.

1.6. Frase de Impacto (Elevator Pitch): "Gala Of The Hidden é um MMORPG sandbox gótico onde a linha entre jogador e NPC se dissolve. Mergulhe em uma Sibiu de 1940, um mundo vivo e reativo onde suas ações moldam uma narrativa emergente de intriga

sobrenatural, horror pessoal e sobrevivência. Aqui, o roleplay não é uma escolha, é a chave para a sua existência."

1.7. Público-Alvo: * Jogadores de MMORPG que buscam uma experiência mais profunda e menos "grind-oriented". * Fãs de jogos sandbox e de simulação emergente (ex: Dwarf Fortress, RimWorld, Space Station 13). * Entusiastas do universo New World of Darkness e de temas góticos/sobrenaturais. * Jogadores que valorizam a narrativa emergente e a liberdade de escolha com consequências significativas. * Pessoas que apreciam um estilo de arte único e otimizado para hardware variado.

1.8. Proposta de Valor Única: GotH se diferencia por sua abordagem radicalmente sistêmica ao MMORPG. Em vez de uma história linear, o jogo oferece um mundo simulado onde a narrativa emerge das interações complexas entre jogadores e NPCs, impulsionadas por um backend de ponta (SpacetimeDB) que garante fluidez e persistência. O "roleplay sistêmico" força os jogadores a viverem seus papéis, tornando cada decisão crucial para a sobrevivência e progressão em um universo gótico de horror pessoal. A otimização para hardware variado e o modelo de monetização justo garantem acessibilidade e sustentabilidade a longo prazo.

1.9. Modelo de Monetização: Buy-to-Play (B2P) para o jogo base, complementado por Expansões de Conteúdo pagas e uma Loja de Cosméticos. Este modelo de "Games as a Service" (GaaS) visa a sustentabilidade do desenvolvimento contínuo sem a necessidade de uma assinatura mensal obrigatória ou mecânicas "pay-to-win".

2. Filosofia de Design e Pilares

2.1. Simulador de Feeling: GotH não é apenas um jogo, é uma experiência imersiva projetada para evocar a sensação visceral de ser uma criatura sobrenatural em um mundo secreto e perigoso. A imersão será alcançada através de: * **Feedback Sensorial:** Sons ambientes ricos, iluminação dramática, efeitos visuais sutis e impactantes que comunicam o estado do mundo e do personagem. * **Feedback Sistêmico:** As consequências das ações do jogador são comunicadas de forma orgânica pelo próprio mundo (reações de NPCs, mudanças no ambiente, eventos emergentes), minimizando a necessidade de interfaces de usuário intrusivas. * **Ausência de UI Excessiva:** A interface será minimalista, priorizando a comunicação visual e sonora para manter o jogador dentro da experiência.

2.2. Narrativa Emergente: A história de GotH não é pré-escrita; ela é um produto orgânico da simulação do mundo e das interações dos jogadores. * **Sementes de Plot:** O desenvolvimento injetará "sementes de plot" (eventos, novos NPCs, artefatos) no mundo, que servirão como catalisadores para a simulação de NPC. * **Reação do Mundo:** NPCs com agendas e memórias reagirão a essas sementes e às ações dos jogadores, gerando cadeias de eventos e tramas complexas. * **Histórias Pessoais:** Cada jogador terá uma história única, moldada por suas escolhas, suas interações com o mundo e as consequências emergentes.

2.3. Roleplay Sistêmico: Em GotH, o roleplay não é uma opção, mas uma necessidade estratégica para a sobrevivência e o sucesso. * Consequências Inevitáveis: Cada ação do jogador tem uma consequência lógica e sistêmica. Quebrar as regras do mundo (ex: a Máscara) resulta em reações severas e escalonadas do ambiente e de seus habitantes. * Incentivo ao Comportamento Coerente: O jogo recompensa a descrição, a manipulação social, a inteligência e a adaptação às regras do mundo sobrenatural. * Transformação do "Griefing": Ações disruptivas não "quebram" o jogo, mas transformam o jogador em um alvo de alto valor para os sistemas de controle do mundo, levando a um caminho de jogo desafiador e, muitas vezes, fatal. * Habilidade do Jogador como Diferencial:** A filosofia central de GotH é que a ficha de personagem fornece as ferramentas, mas a habilidade do jogador em aplicá-las em tempo real é o fator decisivo para o sucesso. O poder não é medido por números absolutos, mas pela capacidade do jogador de tomar decisões estratégicas, aplicar o conhecimento do mundo, gerenciar riscos e executar ações com o timing correto. Isso garante uma jogabilidade fundamentalmente horizontal, onde a inteligência e a astúcia do jogador superam a força bruta, recompensando o mérito e a imersão.

2.4. Otimização e Acessibilidade: A fluidez da experiência e a capacidade de rodar em uma ampla gama de hardware são pilares fundamentais. * **Estilo de Arte Estilizado:** O estilo Neo-Gothic Expressionist Comic Art é inerentemente mais otimizado que o fotorrealismo. * **Tecnologias de Otimização:** Uso extensivo de LODs, Mesh Combine, Occlusion Culling e materiais eficientes na Unity. * **Backend Eficiente:** O SpacetimeDB é projetado para alta performance e baixa latência, garantindo que a simulação do mundo não seja um gargalo.

2.5. Mundo Vivo e Reativo: Sibiu em 1940 não é um pano de fundo estático, mas um ecossistema dinâmico que respira e reage. * **Simulação de NPC Profunda:** NPCs com agendas, rotinas, relacionamentos e memórias que os tornam agentes autônomos no mundo. * **Ciclo Dinâmico:** O ciclo dia/noite e a passagem do tempo afetam

diretamente a jogabilidade e o comportamento de NPCs e jogadores. * **Sistemas Interconectados:** As ações em um sistema (ex: quebra da Máscara) reverberam em outros (ex: reputação, geração de Caçadores, eventos de pânico).

2.6. Metagame como Conhecimento Onírico Emergente (O Sonhar/Realidade Alternativa): GotH integra o conhecimento do jogador (metagame) na lore e nas mecânicas de forma orgânica, concebendo-o como uma manifestação do **Sonhar (The Dreaming)** ou de uma **Realidade Alternativa** que permeia o mundo físico. Em vez de uma mecânica de "pontos", o acesso a esse conhecimento onírico é uma **habilidade passiva e emergente** ligada à inteligência, sabedoria e conexão com o sobrenatural do personagem. O "tempo de inatividade" do jogador (quando o personagem está "dormindo" ou em torpor) se torna um período ativo de exploração onírica, onde Magos e Changelings, em particular, podem acessar informações sutis ou até mesmo influenciar o mundo de formas veladas, transformando o metagame em uma extensão natural de suas habilidades inatas e da simulação de NPC.

3. Tecnologia e Arquitetura

3.1. Visão Geral da Arquitetura: GotH adota uma arquitetura de jogo distribuída e orientada a dados, onde o estado do mundo é mantido de forma centralizada e consistente pelo SpacetimeDB, e a lógica do cliente (Unity) se concentra na renderização e na interface do usuário. Esta abordagem minimiza a complexidade do backend tradicional de MMORPGs, permitindo um foco maior na simulação de mundo e na experiência do jogador.

3.2. Game Engine: Unity 3D

- **Versão:** Será utilizada a versão LTS (Long Term Support) mais recente da Unity 3D para garantir estabilidade e suporte a longo prazo.
- **Render Pipeline: Universal Render Pipeline (URP):**
 - **Escolha:** Selecionado por sua flexibilidade, escalabilidade e desempenho otimizado para uma ampla gama de plataformas, alinhando-se com o pilar de "Otimização e Acessibilidade".
 - **Benefícios Técnicos:**
 - **Single-Pass Forward Rendering:** Método de renderização eficiente que reduz o número de draw calls e otimiza o processamento de luzes, crucial para cenas complexas de MMORPG.

- **Customização de Shaders:** Permite a criação de shaders personalizados para alcançar o estilo "Neo-Gothic Expressionist Comic Art" (ex: cel-shading, contornos, efeitos de pós-processamento estilizados) sem comprometer o desempenho.
 - **Otimização Integrada:** Ferramentas e fluxos de trabalho otimizados para LODs (Levels of Detail), Occlusion Culling e instancing de GPU.
- **Linguagem de Scripting:** C# (para lógica de cliente, UI, integração com SDK do SpacetimeDB).
- **Ferramentas Essenciais da Unity:**
 - **ProBuilder:** Para prototipagem rápida de ambientes e estruturas.
 - **Shader Graph:** Para criação visual de shaders complexos e estilizados.
 - **VFX Graph:** Para criação de efeitos visuais otimizados e performáticos (poderes, impactos, destruição).
 - **Post Processing Stack:** Para controle fino da atmosfera visual e efeitos de câmera.

3.3. Backend: SpacetimeDB

- **Conceito:** SpacetimeDB é a espinha dorsal do backend de GotH, atuando como um banco de dados transacional em memória, um servidor de jogo e um sistema de sincronização de estado em tempo real. Ele implementa uma arquitetura de "Database-Oriented Design".
- **Linguagem de Lógica de Jogo:** Módulos de lógica de jogo (reducers) serão escritos em **Rust** (preferencialmente, devido ao desempenho e segurança de memória) ou C# (se a equipe tiver mais familiaridade inicial).
- **Vantagens Técnicas e Operacionais Chave:**
 - **Zero Downtime Updates:**
 - **Mecanismo:** A arquitetura do SpacetimeDB permite a atualização da lógica do jogo (módulos) e até mesmo do esquema do banco de dados sem a necessidade de parar o servidor. Isso é alcançado através de um sistema de hot-swapping de módulos e um log de transações robusto que garante a consistência dos dados durante a transição.
 - **Impacto:** Elimina interrupções de serviço para os jogadores, garantindo disponibilidade contínua e permitindo iterações rápidas no desenvolvimento e balanceamento do jogo.

- **Sincronização de Estado em Tempo Real (Built-in):**
 - **Mecanismo:** Clientes Unity se conectam diretamente ao SpacetimeDB e "assinam" queries (consultas) para os dados do mundo que lhes interessam (ex: NPCs na área, estado de objetos). O SpacetimeDB gerencia automaticamente a replicação e o envio de atualizações delta para os clientes, mantendo um cache local consistente.
 - **Impacto:** Reduz drasticamente a complexidade do código de rede no cliente e no servidor, garantindo que o estado do mundo seja sempre consistente e atualizado para todos os jogadores com latência mínima. Essencial para a fluidez e a reatividade da simulação de NPC.
- **Eficiência e Escalabilidade:**
 - **In-Memory Database:** O estado do jogo é mantido primariamente em memória RAM para acesso de leitura/escrita ultrarrápido (latência de microssegundos por transação). A durabilidade é garantida por um Write-Ahead Log (WAL) persistente.
 - **ECS-Inspired Design:** A estrutura de dados do SpacetimeDB é inspirada no padrão Entity-Component-System, o que o torna altamente eficiente para modelar e processar dados de jogos, permitindo simulações complexas com muitos agentes.
 - **Alto Throughput:** Capaz de processar milhões de transações por segundo, suportando um grande número de jogadores e interações simultâneas.
- **Consistência Forte (ACID):**
 - **Mecanismo:** Todas as operações de escrita (reducers) são transações ACID (Atomicidade, Consistência, Isolamento, Durabilidade), garantindo que o estado do jogo nunca seja corrompido ou inconsistente, mesmo em ambientes distribuídos.
 - **Impacto:** Crucial para a integridade de um MMORPG, onde a confiança na persistência das ações do jogador é fundamental.
- **Simplificação da Arquitetura de Backend:**
 - **Unificação:** Elimina a necessidade de gerenciar múltiplos serviços (servidor de jogo, banco de dados, fila de mensagens, etc.). A lógica do jogo reside diretamente no banco de dados.

- **Impacto:** Reduz a sobrecarga de DevOps e permite que a equipe de desenvolvimento se concentre na lógica do jogo e na experiência do usuário, em vez de na infraestrutura complexa.
- **SDK de Integração:** Será utilizado o SDK oficial do SpacetimeDB para Unity (C#) para a comunicação cliente-servidor. Este SDK abstrai a complexidade da rede, fornecendo APIs para assinar dados, enviar reducers e receber callbacks de atualização.

3.4. Ferramentas de Desenvolvimento e Otimização:

- **Advanced Tools Mega Pack (Unity Asset Store):**
 - **Funcionalidades:** Essencial para otimização de assets e implementação de ambientes destrutíveis. Inclui ferramentas para:
 - **Fracturing & Destruction:** Criação de malhas pré-fraturadas para objetos e ambientes destrutíveis.
 - **Automatic LOD:** Geração e gerenciamento automático de Levels of Detail para otimização de renderização.
 - **Mesh Simplify/Decimate:** Redução da contagem de polígonos em modelos 3D.
 - **Mesh Combiner:** Combinação de malhas estáticas para reduzir draw calls.
 - **Impacto:** Permite a criação de um mundo visualmente rico e interativo que ainda roda fluidamente em hardware menos potente.
 - **Versionamento de Código:** Git (com LFS para assets grandes) para controle de versão e colaboração.
 - **Ferramentas de Modelagem 3D:** Blender (ou similar) para criação de assets 3D.
 - **Ferramentas de Texturização:** Substance Painter/Designer (ou similar) para criação de texturas estilizadas.
-

4. Game Design Document (GDD) - Foco em Sistemas

Esta seção detalha os sistemas de jogo que compõem a experiência de GotH, com ênfase em como eles contribuem para a narrativa emergente, o roleplay sistêmico e a simulação de mundo.

4.1. Core Gameplay Loop

O ciclo de gameplay principal em GotH é impulsionado pelas necessidades do personagem, as oportunidades do mundo simulado e as consequências das ações do jogador.

1. **Identificar Necessidade/Objetivo:** O jogador percebe uma necessidade imediata do personagem (ex: Fome da Fera, baixa Sanidade, necessidade de recursos) ou um objetivo de longo prazo (ex: ganhar reputação com uma facção, investigar um mistério, adquirir um artefato).
2. **Planejar Ação:** Com base nas informações disponíveis (conhecimento do mundo, reputação, estado de NPCs), o jogador formula um plano para atingir seu objetivo, considerando os riscos e as potenciais consequências.
3. **Executar Ação:** O jogador interage com o mundo e seus habitantes (NPCs e outros jogadores) para realizar o plano. Isso pode envolver combate, interação social, exploração, investigação, uso de poderes sobrenaturais, etc.
4. **Lidar com Consequências:** As ações do jogador reverberam através dos sistemas do jogo. O mundo reage (NPCs mudam de comportamento, reputação é alterada, eventos emergentes são disparados, a Máscara é afetada). O jogador deve então reagir a essas consequências.
5. **Adaptar e Repetir:** Com base nas novas informações e no estado do mundo, o jogador adapta seus objetivos e planos, reiniciando o ciclo.

Este loop é contínuo e emergente, sem um fim pré-definido, incentivando a exploração, a experimentação e a adaptação.

4.2. Ambientação e Mundo

- **Local:** Sibiu, Romênia (Transilvânia).
 - **Descrição:** Uma cidade histórica com forte influência medieval e barroca, caracterizada por ruas de paralelepípedos, praças amplas, edifícios antigos e uma atmosfera gótica. A cidade é cercada por paisagens naturais, incluindo florestas densas e as montanhas dos Cárpatos.
 - **Contexto Histórico (1940s):** O jogo se passa durante a Segunda Guerra Mundial, com a Romênia sob influência ou ocupação. Este período oferece um pano de fundo de tensão social, escassez de recursos e um ambiente de

incerteza que complementa o horror pessoal. A arquitetura e o estilo de vida refletem a era pré-computação, com elementos Art Deco e góticos.

- **Lore do "Milagre":** A "milagrosa" preservação de Sibiu durante a guerra é um ponto central da lore emergente. Isso será atribuído a intervenções sobrenaturais (pactos vampíricos, rituais mágicos, proteção de matilhas de lobisomens, ou a presença de um artefato/entidade poderosa), que os jogadores podem descobrir e influenciar.
- **Ciclo Dia/Noite:**
 - **Duração:** Um ciclo completo de 24 horas de jogo será condensado para aproximadamente 2 horas de tempo real (12x aceleração).
 - **Impacto na Jogabilidade:**
 - **Dia:** Predominância de NPCs humanos em suas rotinas diurnas. Visibilidade alta. Perigoso para a maioria dos *splats* sobrenaturais (especialmente vampiros).
 - **Noite:** Predominância de NPCs sobrenaturais e atividades ocultas. Visibilidade reduzida. Ambiente mais propício para atividades sobrenaturais.
 - **SpacetimeDB Implementation:**
 - `TABLE world_state { current_game_time: u64, day_night_cycle_speed_multiplier: f32, is_day: bool }`
 - Um módulo no SpacetimeDB (`time_controller`) será responsável por avançar `current_game_time` e atualizar `is_day` com base no `day_night_cycle_speed_multiplier` .
 - Outros módulos e sistemas (NPC AI, poderes de *splat*) assinarão `world_state` para ajustar seu comportamento e efeitos.

4.3. Sistemas de Personagem

- **Splats (Raça/Classe):**
 - **Conceito:** Cada *splat* representa uma facção sobrenatural principal ou um tipo de mortal com habilidades únicas. Ao criar um personagem, o jogador escolhe um *splat* e recebe um background narrativo inicial que o posiciona como um recém-transformado ou um indivíduo com habilidades latentes.

Isso garante uma imersão imediata e um ponto de partida para a progressão horizontal.

- **Atributos e Habilidades:**

- **Atributos:** Força, Destreza, Vigor, Carisma, Manipulação, Aparência, Percepção, Inteligência, Raciocínio. Estes são a base para testes de perícia e combate.
- **Habilidades:** Conhecimentos e proficiências (ex: Briga, Armas Brancas, Furtividade, Persuasão, Ocultismo, Investigação).
- **Progressão:** A progressão é horizontal, focada no crescimento de conhecimento e influência, não em números absolutos de poder. Habilidades são aprimoradas através da prática e do treinamento com NPCs (desbloqueados por reputação).

- **Poderes Sobrenaturais:**

- **Conceito:** Habilidades únicas e temáticas para cada *splat*, com custos (recursos) e consequências (quebra da Máscara, Paradoxo, Fúria).
- **Implementação:** Ativação via atalhos ou menus contextuais. Feedback visual e sonoro claro para o uso de poderes.
- **SpacetimeDB Implementation:**
 - `TABLE character_attributes { id: u64, strength: u8, dexterity: u8, ... }`
 - `TABLE character_skills { id: u64, skill_name: string, level: u8 }`
 - `TABLE character_powers { id: u64, power_id: u16, power_level: u8, current_cooldown: u64 }`
 - Reducers para `use_power(character_id, power_id, target_id, location_id)` que verificam custos, aplicam efeitos e atualizam o estado do mundo e do personagem.

- **Splats Jogáveis (MVP):**

- **Vampiro:**

- **Background:** Recém-abraçado, lutando para entender sua nova existência e as regras da sociedade vampírica de Sibiu.
 - **Mecânicas Principais:**

- **Fome da Fera:** Necessidade constante de sangue.
 - **Torpor Diurno:** Inatividade forçada durante o dia.
 - **Poderes Iniciais:** Celeridade (velocidade), Potência (força), Presença (influência social).
 - **Vantagens/Desvantagens do Ciclo:** x3 de bônus em atributos/habilidades no ciclo noturno.
 - **SpacetimeDB:** `TABLE character_splat_vampire { id: u64, blood_points: u8, torpor_until: u64 }`
- **Hunter (Mortal):**
 - **Background:** Um mortal que teve um encontro traumático com o sobrenatural e agora busca entender e combater as criaturas da noite.
 - **Mecânicas Principais:**
 - **Foco em Investigação:** Habilidades e equipamentos para rastrear e descobrir segredos.
 - **Aura de Debuff:** Passivamente impõe debuffs em *splats* sobrenaturais próximos (ex: -1 em atributos, -10% em velocidade). Este efeito é sutil e não visível para o sobrenatural, mas sentido.
 - **Vantagens/Desvantagens do Ciclo:** x1 em todos os ciclos, mas com debuffs que fazem outros *splats* se tornarem x1 próximos dele.
 - **SpacetimeDB:** `TABLE character_splat_hunter { id: u64, investigation_points: u8, active_debuff_aura: bool }`

- **Splats Jogáveis (Objetivos Futuros):**

- **Lobisomem:** Fúria, Gnose, formas lupinas, territórios, caerns. Vantagens x2 no ciclo noturno.
- **Mago:** Mana, Esferas, Paradoxo, rituais. Regras especiais para o Paradoxo baseado em LOD de região.
- **Changeling:** Glamour, ilusões, atração de Caçadores escalonada com uso de poder em áreas de alto paradoxo. x1 em todos os ciclos.
- **Random Mortals:** Personagens permadeath com arquétipos aleatórios (bruxa, psíquico, ghoul, dhampyr, wolf-blooded, proximi, alchemist, fae-

touched, slasher, ghost). Regras especiais de criação e permadeath.

4.4. Sistemas de Sobrevivência e Horror Pessoal

Estes sistemas são projetados para impulsionar o roleplay sistêmico, forçando os jogadores a lidar com as realidades sombrias de sua existência sobrenatural.

- **Máscara/Humanidade:**

- **Conceito:** Representa o segredo da existência sobrenatural e a conexão do personagem com sua humanidade. A manutenção da Máscara é vital para a sobrevivência da sociedade sobrenatural.
- **Medidor:** `mask_score` (0-100) e `humanity_score` (0-100).
- **Quebra da Máscara:** Ações que diminuem a Máscara incluem:
 - Uso visível de poderes sobrenaturais perto de mortais.
 - Drenar sangue/energia de mortais de forma descuidada.
 - Expor a existência de outros sobrenaturais.
 - Cometer atos de violência extrema ou desumanidade.
- **Consequências Sistêmicas:**
 - **Reações de NPCs:** Mortais que testemunham quebras da Máscara podem entrar em pânico, denunciar às autoridades, ou se tornar investigadores. NPCs sobrenaturais podem reagir com desaprovação, raiva ou até mesmo hostilidade.
 - **Geração de Caçadores:** Ações de quebra da Máscara (especialmente em áreas de alto tráfego mortal) podem disparar a geração de NPCs Caçadores de diferentes níveis de ameaça, que investigarão e perseguirão o jogador.
 - **Perda de Status/Excomunhão:** Facções sobrenaturais podem impor penalidades, desde perda de reputação até excomunhão, cortando o acesso a recursos e proteção.
 - **Eventos de Pânico/Caos:** Quebras da Máscara em larga escala ou frequentes podem levar a eventos de pânico na cidade (ex: toque de recolher, aumento da presença policial, histeria em massa), tornando a vida mais difícil para todos os sobrenaturais.
- **SpacetimeDB Implementation:**

- `TABLE character_identity { id: u64, public_name: string, hidden_name: string, current_mask_score: u8, current_humanity_score: u8 }`
- `TABLE npc_knowledge { npc_id: u64, character_id: u64, knows_supernatural: bool, suspicion_level: u8 }`
- `TABLE world_events { event_id: u64, event_type: string, location: u64, severity: u8, triggered_by: u64 }`
- **Reducers:** `expose_supernatural_act(character_id, npc_id, act_type, severity)` que atualiza `current_mask_score`, `suspicion_level` e pode disparar `world_events` ou a geração de `hunter_npc`.

- **Fome da Fera (Vampiros, Lobisomens):**

- **Conceito:** Uma necessidade primordial e crescente que, se não for satisfeita, leva à perda de controle e à manifestação da natureza bestial do personagem.
- **Medidor:** `hunger_of_beast` (0-100), aumenta com o tempo e o uso de poderes.
- **Consequências:**
 - **Debuffs:** Diminuição de atributos, dificuldade em testes sociais.
 - **Comportamentos Compulsivos:** O personagem pode ser forçado a buscar alimento de forma irracional, arriscando a Máscara.
 - **Frenesi (Vampiros):** Perda total de controle, atacando indiscriminadamente.
 - **Histeria (Lobisomens):** Transformação incontrolável e violência.
- **SpacetimeDB Implementation:**
 - `TABLE character_survival_stats { id: u64, hunger_of_beast: u8, sanity: u8, humanity: u8 }`
 - **Reducers:** `update_hunger(character_id, delta)` (chamado periodicamente ou por uso de poder), `satiating_hunger(character_id, amount)` (chamado ao consumir alimento). Módulos de IA de NPC assinam `hunger_of_beast` para disparar comportamentos de frenesi/histeria.

- **Insanidade (Magos, Mortais, Changelings):**

- **Conceito:** A mente do personagem é corroída pela exposição ao sobrenatural, ao uso de magia, ou a eventos traumáticos.
- **Medidor:** `sanity` (0-100), diminui com eventos traumáticos, uso de magia (Paradoxo), ou exposição prolongada a fenômenos sobrenaturais.
- **Consequências:**
 - **Debuffs:** Penalidades em testes mentais, dificuldade de concentração.
 - **Alucinações:** Visuais e auditivas que distorcem a percepção do mundo.
 - **Comportamentos Irracionais:** O personagem pode agir de forma errática, paranoica ou violenta.
 - **Quebra Mental:** Perda total de controle, levando a um estado de "permadeath" ou ban temporário.
- **SpacetimeDB Implementation:**
 - `TABLE character_survival_stats { id: u64, hunger_of_beast: u8, sanity: u8, humanity: u8 }`
 - `TABLE character_active_effects { id: u64, effect_type: string, duration: u64, source: u64 }`
 - **Reducers:** `update_sanity(character_id, delta),`
`apply_insanity_effect(character_id, effect_type).`

- **Permadeath/Ban Temporário:**

- **Conceito:** A morte ou a falha catastrófica em GotH tem consequências severas, reforçando o peso das decisões do jogador.
- **Condições de Ativação:**
 - **Morte Física:** HP zerado em combate.
 - **Quebra Catastrófica da Máscara:** Exposição em massa, resultando em caçada implacável por Caçadores de alto nível.
 - **Perda Total de Controle:** Sucumbir completamente à Fome da Fera ou à Insanidade.
- **Tipos de Consequência:**

- **Ban Temporário:** Para a maioria das "mortes" ou falhas. O personagem é removido do jogo por um período (ex: 24h, 48h, 1 semana de tempo real). Isso simula o personagem se escondendo, se recuperando de ferimentos graves, ou sendo "removido" da cena por sua facção. Durante o ban, o jogador não pode acessar o personagem.
- **Permadeath Real:** Para condições extremas e raras (ex: ser capturado e executado por uma facção rival, falha crítica em um ritual de alto risco, ou ser caçado por um NPC de nível lendário). O personagem é permanentemente removido, e o jogador deve criar um novo.
- **SpacetimeDB Implementation:**
 - `TABLE character_state { id: u64, status: string, ban_until: u64, last_death_reason: string }`
 - **Reducers:** `set_character_status(character_id, new_status, ban_duration, death_reason)` que atualiza o status do personagem e o tempo de ban. Módulos de IA de NPC assinam `character_state` para reagir a personagens "mortos" ou "banidos".

4.5. Sistema de Conhecimento Onírico / O Sonhar

- **Conceito:** Este sistema representa o acesso do personagem a uma camada de informação e influência que transcende a realidade física, manifestando-se como intuições, sonhos ou ações realizadas em estado de inatividade. O "Sonhar" é um domínio onde o conhecimento metagame do jogador pode ser contextualizado e onde *splats* com forte conexão com o mental e o sobrenatural (especialmente Magos e Changelings) podem operar de forma única.
- **Acesso ao Conhecimento Onírico (Passivo e Emergente):**
 - **Inteligência e Sabedoria:** Personagens com altos atributos de Inteligência e Raciocínio, e habilidades relacionadas a Conhecimento (ex: Ocultismo, Psicologia, Investigação), terão uma maior propensão a receber "intuições" ou "insights" que se alinham com o conhecimento metagame do jogador.
 - **Mecanismo:** O sistema de simulação de NPC (gerenciado pelo SpacetimeDB) pode, de forma sutil, "vazar" informações para o jogador através de:
 - **Diálogos Contextuais:** NPCs podem mencionar algo de forma vaga que se alinha com o conhecimento metagame do jogador.

- **Eventos Ambientais:** Pequenos detalhes visuais ou sonoros que o personagem "percebe" e que o jogador reconhece como uma pista.
 - **Sonhos/Visões:** Quando o personagem está em estado de inatividade (dormindo, em torpor), o jogador pode receber mensagens textuais ou visuais que representam sonhos ou visões, contendo informações sobre:
 - **Relacionamentos de NPC:** Como um NPC se sente em relação a outro NPC ou a um jogador.
 - **Agendas de NPC:** O que um NPC planeja fazer, onde trabalha, seus hábitos.
 - **Segredos Sutis:** Pistas sobre fraquezas de inimigos, localização de itens, ou agendas ocultas de facções.
 - **Natureza do Conhecimento:** Este conhecimento é sempre sutil, interpretativo e nunca direto. O jogador ainda precisa "juntar as peças", mas o sistema fornece as peças que seu personagem "intuitivamente" deveria ter.
- **Atividades Oníricas Durante a Inatividade (Dormindo/Torpor):**
 - **Conceito:** O tempo em que o personagem está inativo (dormindo para mortais, em torpor para vampiros, etc.) não é tempo "perdido", mas sim uma oportunidade para atividades no Sonhar.
 - **Mecanismo:** Quando o jogador desloga ou seu personagem entra em estado de inatividade, o sistema de simulação do SpacetimeDB pode continuar a processar certas ações para o personagem no "Sonhar".
 - **Magos (Sabedoria Alta, Habilidades Mentais):**
 - **Sonambulismo/Busca de Informação:** Magos com alta Sabedoria e habilidades mentais (ex: Ocultismo, Consciência) podem ser configurados para, durante o sono/inatividade, realizar ações de "sonambulismo" ou "busca de informação onírica".
 - **Ações:** Isso pode incluir:
 - **Observar Sonhos de NPCs:** Obter informações sobre o estado emocional de NPCs, seus medos, desejos, e como se sentem em relação a outros.
 - **Rastrear Conexões Mentais:** Entender as relações de poder e influência entre NPCs e facções.

- **Estudar Ocultismo Onírico:** Aprofundar o conhecimento em certas áreas arcanas, talvez até "descobrir" novas magias ou rituais.
 - **Feedback:** O jogador receberia um "relatório de sonho" ao logar novamente, detalhando as informações coletadas.
- **Changelings (Conexão Direta com a Hedge/Sonhar):**
 - **Viagem para a Hedge:** Changelings podem, durante a inatividade, "viajar" para a Hedge (o reino entre o Sonhar e a realidade), um lugar de alto risco e alta recompensa.
 - **Atividades na Hedge:**
 - **Coleta de Glamour:** Recurso vital para Changelings.
 - **Busca por Segredos:** A Hedge pode conter informações sobre segredos do mundo físico ou do Sonhar.
 - **Influência Sutil em NPCs:** Changelings podem tentar "alterar" sonhos de NPCs, influenciando sutilmente seus sentimentos, medos ou até mesmo suas agendas (com risco de Paradoxo ou de atrair caçadores da Hedge).
 - **Riscos:** A Hedge é perigosa. O personagem pode ser caçado por criaturas da Hedge, ou o uso excessivo de influência pode atrair Caçadores para o mundo físico. O jogador receberia um relatório de sucesso/falha e os riscos encontrados.
- **SpacetimeDB Implementation:**
 - `TABLE character_dream_activity { id: u64, last_dream_report: string, dream_activity_type: string, dream_activity_start_time: u64, dream_activity_duration: u64 }`
 - `TABLE npc_dream_state { npc_id: u64, current_dream_content: string, influenced_by_character: u64 }`
 - **Reducers:** `start_dream_activity(character_id, activity_type)`, `process_dream_activity(character_id, activity_type, duration)` (chamado pelo sistema de simulação de NPC durante a inatividade do jogador).
 - Módulos de IA de NPC seriam atualizados para ter `dream_content` e `influenced_by_character` como atributos que afetam seu comportamento.

4.6. Sistemas de Interação Social e Combate Social

Estes sistemas são cruciais para a experiência de MMORPG social de GotH, permitindo que os jogadores naveguem pela complexa teia de relacionamentos e poder no mundo sobrenatural e mortal de Sibiu. O combate social é tão vital quanto o físico, e muitas vezes mais perigoso, com suas consequências reverberando através do ambiente e da percepção dos observadores.

- **Reputação:**

- **Conceito:** A reputação é a principal forma de progressão horizontal em GotH, refletindo a influência e o status de um personagem dentro de diferentes grupos e facções. Não é um medidor de "poder de combate", mas de poder social e acesso.
- **Tipos de Reputação:**
 - **Reputação de Facção:** Mede o status do jogador com facções específicas (ex: Vampiros de Sibiu, Ordem dos Caçadores, Máfia Local, Polícia, Círculos Mágicos).
 - **Reputação Individual (Relacionamentos):** Mede a relação do jogador com NPCs específicos.
- **Ganhando Reputação:**
 - Completando missões ou objetivos para a facção/NPC.
 - Agindo de forma alinhada com os valores da facção (ex: manter a Máscara para vampiros, caçar sobrenaturais para Hunters).
 - Interações sociais bem-sucedidas (persuasão, negociação).
 - Fornecendo informações valiosas ou recursos.
- **Perdendo Reputação:**
 - Agindo contra os interesses da facção/NPC.
 - Quebrando a Máscara (para facções que a prezam).
 - Falhando em missões ou traindo confiança.
 - Comportamento hostil ou anti-social.
- **Consequências da Reputação:**
 - **Desbloqueio de Conteúdo:** Acesso a novas missões, áreas, treinadores de habilidades, recursos e itens exclusivos.

- **Interações de NPC:** NPCs reagem de forma diferente ao jogador (ajuda, desconfiança, hostilidade, respeito).
- **Poder Social:** Capacidade de influenciar eventos, mediar conflitos, formar alianças.
- **SpacetimeDB Implementation:**
 - `TABLE character_reputation { character_id: u64, faction_id: u64, reputation_score: i16 }`
 - `TABLE npc_relationships { npc_id_a: u64, npc_id_b: u64, relationship_score: i16, last_interaction_time: u64 }`
 - **Reducers:** `update_reputation(character_id, target_id/faction_id, delta_score, reason)` (chamado após missões, interações sociais, eventos).

- **Interação Social e Diálogo:**

- **Conceito:** O diálogo é uma ferramenta poderosa para o jogador, permitindo coletar informações, persuadir, enganar, intimidar e negociar.
- **Mecânica:**
 - **Sistema de Diálogo Contextual:** Opções de diálogo que aparecem com base nos atributos do personagem (Carisma, Manipulação, Aparência), habilidades (Persuasão, Intimidação, Enganação), reputação e conhecimento (informações coletadas).
 - **Testes de Habilidade:** Muitas opções de diálogo desencadearão testes de habilidade (ex: Persuasão vs. Vontade do NPC, Enganação vs. Percepção do NPC). O sucesso ou falha afeta a reputação e o fluxo da conversa.
 - **Consequências do Diálogo:** O diálogo pode levar a:
 - Ganhos/perdas de reputação.
 - Descoberta de novas informações (segredos, missões).
 - Mudanças na agenda de NPCs.
 - Início de combate físico ou social.
 - SpacetimeDB Implementation:
 - `TABLE dialogue_options { option_id: u64, text: string, skill_check_type: string,`

```
skill_check_difficulty: u8, success_effect: json,  
failure_effect: json }
```

- **Reducers:** `initiate_dialogue(character_id, npc_id)`, `select_dialogue_option(character_id, npc_id, option_id)` que processa o teste de habilidade e aplica os efeitos.

- **Combate Social e de Vontade:**

- **Conceito:** Conflitos que não envolvem violência física, mas sim a manipulação, o domínio mental, a humilhação ou a destruição da reputação de um oponente. Estes conflitos são dinâmicos e profundamente integrados ao ambiente e à simulação de NPC.
- **Mecânica:**
 - **Recursos:** Utiliza atributos sociais (Carisma, Manipulação, Aparência), habilidades sociais (Persuasão, Intimidação, Enganação) e o recurso de Vontade (Willpower).
 - **Minigames/Testes de Confronto:**
 - **Fase de Abertura:** Jogadores e NPCs usam habilidades sociais para "sondar" o oponente, buscando fraquezas ou pontos de pressão.
 - **Fase de Ataque:** Uso de habilidades sociais para "atacar" a Vontade do oponente. Sucesso causa "dano de Vontade".
 - **Fase de Defesa:** Uso de habilidades sociais ou atributos mentais para "defender" a Vontade.
 - **Poderes Sociais:** Alguns poderes sobrenaturais (ex: Presença de Vampiros, certas Esferas de Magos) podem ser usados como "ataques" ou "defesas" sociais.
 - **Integração com o Ambiente:**
 - **Contexto Local:** O ambiente físico onde o conflito ocorre pode fornecer bônus ou penalidades. Ex: um combate social em um salão de baile elegante pode favorecer personagens com alta Aparência e Etiqueta; um conflito em um beco escuro pode favorecer Intimidação.

- **Uso de Objetos:** Personagens podem usar objetos do ambiente para influenciar o conflito (ex: quebrar um copo para chamar atenção, apontar para uma evidência).
- **Impacto na "Plateia" (NPCs e Outros Jogadores):**
 - **Percepção:** NPCs e outros jogadores presentes no local do conflito social atuarão como "plateia". Sua percepção do conflito (quem está "ganhando", quem está agindo de forma apropriada/inapropriada) será influenciada pelo sucesso/falha dos participantes e pelo contexto.
 - **Reações Emergentes:** A plateia pode reagir de diversas formas:
 - **Aumento/Diminuição de Reputação:** O desempenho no conflito social pode alterar a reputação dos envolvidos com os observadores.
 - **Intervenção:** NPCs ou outros jogadores podem decidir intervir no conflito, seja para apoiar um lado, para acalmar a situação, ou para aproveitar o caos.
 - **Disparo de Eventos:** Um conflito social intenso pode atrair a atenção de guardas, da polícia, ou de facções interessadas, gerando novos eventos.
 - **Transição para Combate Físico:** Se a tensão escalar e a Vontade de um dos lados for completamente quebrada, ou se houver uma quebra da Máscara, o conflito social pode degenerar em combate físico.
- **Consequências do Dano de Vontade:**
 - **Perda de Vontade:** Reduz a capacidade do oponente de resistir a novas influências ou de usar certas habilidades.
 - **Quebra de Vontade:** O oponente é forçado a ceder, aceitar uma condição, revelar um segredo, ou sofrer uma penalidade de reputação/status.
 - **Humilhação Pública:** Pode levar a uma perda massiva de reputação com a plateia.
 - **SpacetimeDB Implementation:**
 - `TABLE character_willpower { id: u64, current_willpower: u8, max_willpower: u8 }`

- `TABLE social_conflict_state { conflict_id: u64, attacker_id: u64, defender_id: u64, current_phase: string, attacker_willpower_damage: u8, defender_willpower_damage: u8, location_id: u64, observers: array<u64> }`
- **Reducers:** `initiate_social_conflict(attacker_id, defender_id, location_id), perform_social_attack(conflict_id, character_id, skill_used), perform_social_defense(conflict_id, character_id, skill_used)`. Módulos de IA de NPC assinam `social_conflict_state` para simular a reação da plateia e a possível intervenção.

4.7. Simulação de Mundo e NPC

Este sistema é o motor da narrativa emergente em GotH, transformando Sibiú em um ecossistema dinâmico onde NPCs são agentes autônomos com suas próprias vidas, agendas e reações. A complexidade emerge da interação desses agentes com o ambiente e com os jogadores.

- **NPCs como Agentes Autônomos:**

- **Conceito:** Cada NPC não é um mero "dummy" passivo, mas uma entidade com um conjunto de atributos, habilidades, personalidade, memórias e uma hierarquia de objetivos e agendas. Eles operam com base em IA baseada em regras, máquinas de estado e árvores de comportamento, processadas no backend do SpacetimeDB.
- **Atributos de NPC:** Incluem características físicas, sociais, mentais, além de traços de personalidade (ex: "corajoso", "desconfiado", "ambicioso", "leal").
- **Rotinas e Comportamentos:** NPCs seguem rotinas diárias/noturnas (trabalho, casa, socialização, patrulha, reuniões secretas). Seus comportamentos são reativos a eventos e ao estado do mundo.
- **SpacetimeDB Implementation:**
 - `TABLE npc_data { id: u64, type: string, current_location: u64, current_agenda_id: u64, current_target_id: u64, health: u8, sanity: u8, personality_traits: array<string> }`

- `TABLE npc_routine { npc_id: u64, time_slot: u8, activity_type: string, location: u64 }`
- **Módulos de IA de NPC:** Funções em Rust/C# no SpacetimeDB que, a cada tick de simulação, avaliam o estado do NPC e do mundo para determinar a próxima ação.

- **Agendas e Objetivos de NPC:**

- **Conceito:** Cada NPC possui uma hierarquia de objetivos que impulsionam suas ações. Estes objetivos podem ser de curto prazo (ex: "ir para o trabalho", "comer") ou de longo prazo (ex: "ganhar influência na facção", "descobrir um segredo", "proteger um local").
- **Priorização Dinâmica:** A prioridade das agendas pode mudar com base em eventos, no estado do mundo, nas interações com jogadores e outros NPCs.
- **Exemplos:**
 - Um NPC vampiro pode ter a agenda "Manter a Máscara" como prioridade máxima, mas se sua "Fome da Fera" atingir um nível crítico, a agenda "Caçar Sangue" pode se tornar dominante.
 - Um NPC detetive humano pode ter a agenda "Patrulhar a Cidade", mas se um evento de "Quebra da Máscara" ocorrer em sua área, sua agenda muda para "Investigar Atividade Sobrenatural".
- **SpacetimeDB Implementation:**
 - `TABLE npc_agenda { agenda_id: u64, npc_id: u64, objective_type: string, target_id: u64, priority: u8, status: string }`
 - **Reducers:** `update_npc_agenda(npc_id, new_agenda_id, new_priority)` (chamado por eventos ou por módulos de IA de NPC).

- **Conhecimento e Memória de NPC:**

- **Conceito:** NPCs "lembram" de eventos, interações e informações relevantes. Este conhecimento influencia suas decisões e reações futuras.
- **Tipos de Conhecimento:**
 - **Fatos Conhecidos:** Informações específicas (ex: "Jogador X é um vampiro", "Corpo encontrado na Rua Y", "Facção Z está planejando um ataque").

- **Relacionamentos:** Como se sentem em relação a outros NPCs e jogadores (amizade, rivalidade, desconfiança).
 - **Percepção:** O que testemunharam (ex: um jogador usando um poder sobrenatural, um NPC cometendo um crime).
- **Disseminação de Conhecimento:** NPCs podem compartilhar informações entre si, ou com facções, propagando o conhecimento pelo mundo.
- **SpacetimeDB Implementation:**
 - `TABLE npc_knowledge { npc_id: u64, fact_id: u64, certainty: u8, source_id: u64, timestamp: u64 }`
 - `TABLE npc_relationships { npc_id_a: u64, npc_id_b: u64, relationship_score: i16, last_interaction_time: u64 }`
 - **Reducers:** `add_knowledge(npc_id, fact_id, certainty, source_id)`, `update_relationship(npc_id_a, npc_id_b, delta_score)`. Módulos de IA de NPC usam `npc_knowledge` e `npc_relationships` para tomar decisões.
- **Eventos Emergentes e "Sementes de Plot":**
 - **Conceito:** A narrativa de GotH é impulsionada por eventos que emergem da simulação do mundo. Estes eventos podem ser disparados por ações de jogadores, interações de NPC, ou por "sementes de plot" injetadas pelos desenvolvedores.
 - **Disparo de Eventos:**
 - **Ações de Jogadores:** Quebra da Máscara, sucesso/falha em missões, assassinato de NPCs importantes.
 - **Interações de NPC:** Conflitos sociais, descobertas, traições, alianças.
 - **"Sementes de Plot" (Atualizações):** Os desenvolvedores podem injetar novos fatos, NPCs, objetos ou agendas no SpacetimeDB.
 - **Exemplo:** A injeção de um `fact_id` como "Artefato Antigo Descoberto nos Esgotos" para alguns NPCs arqueólogos ou magos. Isso fará com que esses NPCs alterem suas agendas para "Investigar Artefato", levando-os a se mover, interagir e potencialmente entrar em conflito com outros NPCs ou jogadores que também buscam o artefato.

- **Cadeias de Eventos:** Um evento pode disparar uma série de outros eventos e reações em cascata, criando tramas complexas e imprevisíveis.
- **SpacetimeDB Implementation:**
 - `TABLE world_events { event_id: u64, event_type: string, location: u64, severity: u8, triggered_by: u64, timestamp: u64, status: string }`
 - **Event Dispatcher:** Um módulo central no SpacetimeDB que monitora o estado do mundo e as ações, disparando novos `world_events` e notificando os módulos de IA de NPC relevantes.
- **Mecanismos de Auto-Correção do Mundo:**
 - **Conceito:** Para evitar que o mundo se torne um caos incontrolável ou seja "quebrado" por jogadores que abusam da simulação, o próprio mundo possui mecanismos de auto-regulação que reagem a comportamentos disruptivos.
 - **"Cleanup Crews":** NPCs especializados (ex: Caçadores, agentes de facções sobrenaturais, polícia secreta) que são gerados ou ativados para investigar e "limpar" as consequências de quebras da Máscara ou de grandes perturbações. Eles rastrearão e perseguirão os responsáveis.
 - **"Wanted" System (Sistema de Procurado):** Jogadores que consistentemente quebram a Máscara ou cometem crimes graves se tornam "procurados" por facções específicas. Isso pode levar a:
 - NPCs hostis à vista.
 - Recompensas (bounties) para outros jogadores que os "neutralizem".
 - Restrições de acesso a áreas ou serviços.
 - **Regeneração de Recursos/Populações:** Se um recurso (ex: população humana para vampiros, território de caça para lobisomens) for excessivamente explorado, ele se esgotará temporariamente, forçando os jogadores a buscar alternativas ou a esperar pela recuperação.
 - **SpacetimeDB Implementation:**
 - `TABLE character_wanted_status { character_id: u64, faction_id: u64, wanted_level: u8, bounty_value: u64 }`
 - `TABLE resource_node_status { node_id: u64, resource_type: string, current_amount: u32, regeneration_rate: u32 }`

- **Módulos de Controle:** Módulos no SpacetimeDB que monitoram o estado do mundo e disparam esses mecanismos de auto-correção quando os limites são excedidos.

4.8. Combate Físico (Action)

O combate físico em GotH é um sistema de ação em tempo real, projetado para ser fluido, tático e visceral, com ênfase na **habilidade do jogador**, no feedback visual e nas consequências de dano localizado. Embora o combate social e a intriga sejam centrais, a violência física é uma ferramenta brutal e com consequências reais, que pode emergir de falhas em interações sociais ou da necessidade de sobrevivência.

- **Sistema de Combate em Tempo Real:**

- **Conceito:** Combate baseado em ação, onde o posicionamento, o timing, a direção do ataque/defesa e o uso estratégico de habilidades e poderes são cruciais. Inspirado na profundidade tática de jogos como Chivalry e Mordhau.
- **Mecânica:**
 - **Ataques Direcionais:** O jogador controla a direção do ataque (ex: de cima para baixo, da esquerda para a direita, da direita para a esquerda) através do movimento do mouse ou input direcional. Isso determina o arco do golpe e a parte do corpo que será atingida.
 - **Defesa Direcional (Parry/Block):** A defesa também é direcional. O jogador deve alinhar sua defesa com a direção do ataque recebido para bloquear ou aparar efetivamente.
 - **Habilidades Ativas:** Poderes sobrenaturais e habilidades de combate que consomem recursos (sangue, mana, fúria, estamina). Podem ser integrados aos ataques direcionais (ex: um ataque de Celeridade que é um golpe rápido de cima para baixo).
 - **Feedback Visual e Sonoro:** Animações fluidas, efeitos de partículas para impactos, sons de golpes e reações de dor, tremores de câmera sutis para golpes fortes.
- **Atributos como Modificadores Orgânicos:**
 - **Conceito:** Os atributos do personagem (Força, Destreza, Vigor, etc.) não determinam diretamente o sucesso ou falha de um ataque/defesa

(isso é habilidade do jogador), mas influenciam a **qualidade e a facilidade** da execução.

- **Mecanismo:**

- **Precisão e Velocidade:** Atributos mais altos podem tornar o "timing" para um ataque perfeito mais perdoável, ou aumentar a velocidade do golpe.
- **Equilíbrio e Recuperação:** Vigor pode influenciar a recuperação de equilíbrio após um golpe bloqueado ou uma esquivas.
- **Impacto:** Força pode aumentar o "stagger" (desequilíbrio) causado no oponente.
- **Fluidez:** O jogador ainda tem controle total, mas o personagem com atributos mais altos se sentirá mais "fluido" e "preciso" em suas ações, refletindo a capacidade inata do personagem.

- **SpacetimeDB Implementation:**

- `TABLE character_combat_state { id: u64, current_hp: u16, max_hp: u16, current_stamina: u8, is_blocking: bool, is_dodging: bool, target_id: u64, last_attack_direction: string, last_defense_direction: string }`
- `TABLE equipped_items { character_id: u64, item_id: u64, slot: string }`
- **Reducers:** `perform_attack(attacker_id, target_id, attack_type, weapon_id, attack_direction), perform_defense(defender_id, defense_direction)`. A lógica do SpacetimeDB calculará o impacto do ataque/defesa com base na direção, nos atributos do personagem e na habilidade do jogador.

- **Dano Localizado e Consequências:**

- **Conceito:** Ataques podem atingir partes específicas do corpo, resultando em diferentes tipos de dano e efeitos colaterais, aumentando o realismo e a profundidade tática.
- **Mecânica:**
 - **Hitboxes:** Modelos de personagens e NPCs terão hitboxes distintas para cabeça, tronco, braços (esquerdo/direito) e pernas

(esquerda/direita). A direção do ataque do jogador influenciará qual hitbox é mais provável de ser atingida.

- **Dano Variável:** O dano base de um ataque pode ser multiplicado ou modificado dependendo da parte do corpo atingida (ex: dano crítico na cabeça, dano reduzido em membros).

- **Efeitos Colaterais Específicos:**

- **Cabeça:** Confusão, desorientação, perda de consciência.
- **Braços:** Dificuldade em usar armas, redução da precisão, incapacidade de bloquear.
- **Pernas:** Lentidão, mancar, dificuldade em esquivar.
- **Impacto:** Força pode aumentar o "stagger" (desequilíbrio) causado no oponente.
- **Tronco:** Penalidades em vigor, sangramento.

- **Visuais de Dano:** Modelos de personagens podem mostrar visuais de dano (ex: roupas rasgadas, feridas visíveis, postura de dor).

- **SpacetimeDB Implementation:**

- `TABLE character_body_parts { id: u64, body_part_name: string, current_hp: u8, max_hp: u8, status_effects: array<string> }`
- **Reducers:** `apply_damage_to_body_part(character_id, body_part_name, damage_amount, damage_type)` que atualiza o HP da parte do corpo e aplica `status_effects`. Clientes assinam `character_body_parts` para ajustar o comportamento e a renderização.

- **Destruição de Ambiente (Integrada ao Combate):**

- **Conceito:** Certos elementos do ambiente podem ser danificados ou destruídos durante o combate, criando oportunidades táticas e alterando o cenário.
- **Mecânica:**
 - **Objetos Destrutíveis:** Paredes, móveis, objetos decorativos podem ter um HP e um estado de destruição.

- **Fratura Pré-Calculada:** Utiliza malhas pré-fraturadas (geradas com o "Advanced Tools Mega Pack") que são trocadas por modelos destruídos quando o HP do objeto chega a zero.
- **Consequências:** A destruição pode abrir novas rotas, criar obstáculos, ou revelar segredos.
- **SpacetimeDB Implementation:**
 - `TABLE destructible_object { id: u66, object_type: string, current_hp: u16, max_hp: u16, current_state: u8, position_x: f32, position_y: f32, position_z: f32 }`
 - **Reducers:** `apply_damage_to_object(object_id, damage_amount)` que atualiza o HP e `current_state`. Clientes assinam `destructible_object` para renderizar o estado correto.

4.9. Progressão e Economia

Estes sistemas definem como os jogadores crescem em influência e capacidade dentro do mundo de GoTH, e como os recursos são adquiridos, gerenciados e trocados. A progressão é intrinsecamente ligada à reputação e ao conhecimento, não a um aumento linear de poder.

- **Progressão Horizontal:**
 - **Conceito:** Em GoTH, a progressão não é medida por "níveis" tradicionais que aumentam o poder numérico do personagem. Em vez disso, o foco é na expansão das capacidades do personagem, no acesso a novas oportunidades e no aumento de sua influência no mundo.
 - **Pilares da Progressão:**
 - **Reputação:** Conforme detalhado na seção 4.5, a reputação com facções e NPCs individuais é o principal motor da progressão. Alta reputação desbloqueia acesso a áreas restritas, missões mais complexas, informações privilegiadas e treinadores de habilidades.
 - **Habilidades:** O personagem aprimora suas habilidades (ex: Persuasão, Investigação, Armas Brancas) através da prática e do treinamento com NPCs especializados. O aumento de uma habilidade abre novas opções de diálogo, combate ou interação com o ambiente.

- **Conhecimento:** A descoberta de segredos, a compreensão da lore emergente e o acesso a informações ocultas (incluindo através do Sistema de Conhecimento Onírico) são formas cruciais de progressão. Conhecimento pode desbloquear novas estratégias, poderes ou missões.
 - **Influência Social:** A capacidade de manipular NPCs, formar alianças, mediar conflitos e moldar eventos no mundo é a verdadeira medida do "poder" de um personagem.
 - **Recursos e Equipamento:** Adquirir recursos raros, artefatos e equipamentos especializados que se alinham com o estilo de jogo do personagem.
- **SpacetimeDB Implementation:**
 - `TABLE character_skills { id: u64, skill_name: string, level: u8, xp_to_next_level: u32 }`
 - `TABLE character_knowledge { id: u64, knowledge_fact_id: u64, discovered_timestamp: u64 }`
 - **Reducers:** `gain_skill_xp(character_id, skill_name, amount), discover_knowledge(character_id, knowledge_fact_id)`.
- **Recursos e Economia:**
 - **Conceito:** Os recursos em GotH vão além do dinheiro, incluindo elementos vitais para a sobrevivência sobrenatural, informações e favores. A economia é impulsionada pela simulação de NPC e pelas necessidades dos jogadores.
 - **Tipos de Recursos:**
 - **Moeda:** Dinheiro (ex: Lei Romena, moedas de ouro antigas) para transações básicas com mortais.
 - **Recursos Sobrenaturais:**
 - **Sangue (Vampiros):** Essencial para a sobrevivência e uso de poderes. Adquirido de mortais (com risco de Máscara) ou de fontes alternativas.
 - **Gnose/Fúria (Lobisomens):** Recursos para poderes e transformações. Adquirido de locais sagrados (caerns) ou através de rituais.

- **Mana (Magos):** Recurso para lançar feitiços. Adquirido através de meditação, estudo ou locais de poder.
- **Glamour (Changelings):** Recurso para poderes e ilusões. Adquirido através de emoções humanas ou do Sonhar.
- **Informações:** Segredos, agendas de NPCs, fraquezas de inimigos, localização de artefatos. Pode ser comprada, trocada ou obtida através de investigação.
- **Favores:** Compromissos sociais ou políticos que podem ser cobrados de NPCs ou outros jogadores.
- **Artefatos:** Itens raros com propriedades sobrenaturais, que podem ser usados, trocados ou vendidos.
- **Aquisição de Recursos:**
 - **Trabalhos/Atividades:** Jogadores podem realizar trabalhos que se encaixam em seus *splats* e reputação (ex: vampiros manipulando mercados financeiros, magos criando itens arcanos, lobisomens caçando em territórios específicos, hunters realizando investigações privadas).
 - **Exploração e Descoberta:** Encontrar recursos em locais ocultos ou através de investigação.
 - **Interação Social:** Negociação, persuasão, intimidação para obter recursos ou favores de NPCs.
 - **Caça/Drenagem:** Para recursos sobrenaturais (com as devidas consequências de Máscara/Fome da Fera).
- **Mercado Negro e Trocas:**
 - **NPCs Comerciantes:** NPCs especializados (humanos e sobrenaturais) que compram e vendem itens e informações. Os preços podem flutuar com base na oferta/demanda e na reputação do jogador com o NPC.
 - **Trocas entre Jogadores (Sistema de Balança de Valor):**
 - **Conceito:** A troca direta entre jogadores não é uma simples permuta de itens, mas uma negociação de valor que considera não apenas o valor intrínseco dos bens, mas também a influência social e a hierarquia entre os jogadores. Inspirado na mecânica de troca de Dwarf Fortress, onde o valor total dos bens trocados deve ser equilibrado.

- **Mecanismo:**

- **Valor Intrínseco do Item:** Cada item tem um valor base (que pode ser influenciado pela raridade, demanda e contexto do mundo).
- **Balança de Valor:** Durante uma proposta de troca, o sistema calcula a diferença de valor entre o que cada jogador está oferecendo.
- **Influência Social como Modificador:** A diferença de influência social (baseada na reputação geral, status de facção, ou hierarquia de *splat*) entre os dois jogadores pode alterar a "balança justa".
 - Um jogador com **maior influência** pode exigir uma troca desequilibrada a seu favor, e o jogador de menor influência pode ser forçado a aceitar para evitar penalidades sociais ou para ganhar um favor.
 - Um jogador com **menor influência** terá que oferecer mais para equilibrar a balança ou para compensar a diferença de status.
- **Favores e Dívidas:** Trocas podem incluir "favores" ou "dívidas" que não são itens físicos, mas compromissos sociais que podem ser cobrados no futuro.
- **Consequências:** Trocas muito desequilibradas (mesmo que aceitas) podem gerar ressentimento, perda de reputação com observadores, ou até mesmo futuras retaliações por parte do jogador desfavorecido.

- **SpacetimeDB Implementation:**

- `TABLE character_inventory { character_id: u64, item_id: u64, quantity: u32 }`
- `TABLE item_data { item_id: u64, item_name: string, item_type: string, base_value: u64, properties: json }`
- `TABLE character_social_status { character_id: u64, overall_influence_score: u16, splat_hierarchy_rank: u8 }`

- **Reducers:** `propose_trade(proposer_id, receiver_id, items_offered_proposer, items_offered_receiver), accept_trade(trade_id)`. A lógica do SpacetimeDB calculará o valor da troca, aplicando modificadores de influência social, e executará a transferência de itens e o registro de favores/dívidas.
-

5. Regras de Negócio

Esta seção descreve o modelo de monetização de GotH e a estratégia de "Games as a Service" (GaaS), visando a sustentabilidade do desenvolvimento contínuo, a acessibilidade para os jogadores e a manutenção da integridade do jogo.

5.1. Modelo de Monetização:

GotH adota um modelo híbrido que combina a compra única do jogo com vendas de conteúdo adicional e itens cosméticos, evitando a necessidade de uma assinatura mensal obrigatória e práticas "pay-to-win".

- **Buy-to-Play (B2P) - Jogo Base:**

- **Mecanismo:** Os jogadores adquirem o jogo base através de uma compra única. Esta compra concede acesso completo ao cliente do jogo e a todo o conteúdo fundamental (MVP e atualizações gratuitas).
- **Propósito:** Financiar o desenvolvimento inicial do jogo e cobrir os custos operacionais básicos dos servidores.
- **Preço:** Será definido para ser competitivo no mercado de MMORPGs B2P, refletindo o valor de um jogo com sistemas profundos e uma experiência única.

- **Expansões de Conteúdo (Pagos):**

- **Mecanismo:** Pacotes de conteúdo substanciais que introduzem grandes adições ao jogo, vendidos separadamente.
- **Propósito:** Gerar picos de receita significativos para financiar o desenvolvimento de longo prazo, a expansão da equipe e a criação de grandes blocos de conteúdo.
- **Conteúdo Típico:**

- Novas regiões geográficas (ex: outras cidades na Romênia, áreas rurais extensas).
 - Novos *splats* jogáveis principais (ex: Lobisomens, Magos, Changelings, com suas mecânicas e lore aprofundadas).
 - Grandes sistemas de jogo (ex: sistema de moradia de jogador complexo, sistema de criação de itens avançado, sistema de política de facção aprofundado).
 - Grandes eventos de lore que alteram fundamentalmente o estado do mundo simulado.
- **Frequência:** Planejadas para serem lançadas a cada 12-24 meses, dependendo do escopo e da capacidade de desenvolvimento.
- **Loja de Cosméticos (Pagos):**
 - **Mecanismo:** Uma loja in-game que oferece itens puramente estéticos para personalização de personagens e, futuramente, de esconderijos/propriedades.
 - **Propósito:** Gerar uma receita contínua e estável, complementando as vendas do jogo base e das expansões.
 - **Filosofia "Não Pay-to-Win":** Todos os itens vendidos na loja serão estéticos e não conferirão qualquer vantagem de gameplay, poder ou progressão. Isso garante um campo de jogo justo e mantém a integridade da experiência.
 - **Conteúdo Típico:**
 - Roupas, acessórios e estilos de cabelo temáticos (góticos, de época, sobrenaturais).
 - Aparências alternativas para poderes sobrenaturais (efeitos visuais).
 - Itens de decoração para esconderijos/casas.
 - Emotes, títulos e ícones de perfil.
 - **Frequência:** Novos itens cosméticos serão adicionados regularmente (semanal ou mensalmente) para manter a loja atualizada e atraente.

5.2. Estratégia "Games as a Service" (GaaS):

GotH será desenvolvido e mantido como um serviço contínuo, evoluindo e crescendo ao longo do tempo sem a necessidade de lançar sequências numeradas.

- **Atualizações Gratuitas Contínuas:**

- **Mecanismo:** Lançamento regular de patches e atualizações que não exigem compra adicional.
- **Conteúdo Típico:**
 - **Qualidade de Vida (QoL):** Melhorias na interface do usuário, otimizações de desempenho, correções de bugs, ajustes de balanceamento.
 - **Conteúdo Menor/Médio:** Novas "sementes de plot" para a simulação de NPC, pequenos eventos emergentes, expansões incrementais de áreas existentes (ex: novos interiores de edifícios, becos), novos itens ou habilidades menores.
 - **Melhorias de Engine/IA:** As melhorias na Unity e na simulação de IA de NPC (incluindo a futura IA avançada) serão entregues como parte dessas atualizações gratuitas.
- **Propósito:** Manter o jogo fresco, engajar a comunidade e demonstrar o compromisso com o desenvolvimento de longo prazo.

- **Zero Downtime:**

- **Mecanismo:** Utilização da arquitetura do SpacetimeDB para permitir atualizações de lógica de jogo e esquema de banco de dados sem a necessidade de tirar os servidores do ar.
- **Propósito:** Maximizar a disponibilidade do jogo para os jogadores, minimizando a frustração de interrupções de serviço e permitindo um ciclo de desenvolvimento e implantação mais ágil.

- **Comunicação Transparente com a Comunidade:**

- **Mecanismo:** Manutenção de um roadmap de desenvolvimento público, blogs de desenvolvedores, sessões de Q&A, e canais de feedback ativos (fóruns, Discord).
- **Propósito:** Construir uma comunidade engajada e leal, que se sinta parte do processo de desenvolvimento e entenda a direção do jogo.

5.3. Custos Operacionais (Estimativa Inicial):

- **Servidores:** O principal custo operacional contínuo será a infraestrutura de servidores para o SpacetimeDB e os serviços de rede associados. O SpacetimeDB é projetado para ser eficiente, mas a escala de um MMORPG exigirá recursos significativos.
 - **Estimativa:** Será necessário um planejamento detalhado de capacidade e custos de provedores de nuvem (AWS, Google Cloud, Azure) à medida que o projeto avança para o MVP e além.
 - **Licenças de Software:** Custos associados a licenças da Unity Pro (se necessário), ferramentas de arte e outros softwares de desenvolvimento.
 - **Manutenção e Suporte:** Custos de equipe para monitoramento de servidores, suporte ao cliente e manutenção geral do jogo.
-

6. Roadmap (MVP e Futuro)

Este roadmap delinea as fases de desenvolvimento de Gala Of The Hidden, começando com o Produto Mínimo Viável (MVP) e projetando os objetivos de crescimento e expansão do jogo.

6.1. Fase 1: Produto Mínimo Viável (MVP)

O MVP de GotH será uma prova de conceito jogável que demonstra o "feeling" central do jogo, a eficácia da simulação de mundo e a viabilidade da arquitetura tecnológica. O objetivo é criar uma experiência coesa e imersiva, mesmo que em escala limitada, para validar a visão e atrair futuros colaboradores e investidores.

- **6.1.1. Área Jogável:**
 - **Localização:** Um quarteirão ou distrito de Sibiu (ex: a Praça Grande e ruas adjacentes) com arquitetura gótica e Art Deco.
 - **Edifícios:** 3-5 edifícios com interiores acessíveis (ex: uma mansão, um bar/clube, uma loja, um esconderijo básico).
 - **Natureza:** Uma pequena área natural adjacente (ex: um cemitério, um parque isolado) para transição entre urbano e natural.
- **6.1.2. Splats Jogáveis (2):**
 - **Vampiro:**

- **Mecânicas:** Fome da Fera (necessidade de sangue), Ciclo Dia/Noite com Torpor diurno (inativado), 3-4 poderes básicos (ex: Celeridade, Potência, Presença).
 - **Progressão:** Ganhos de reputação com uma facção vampírica local.
- **Hunter (Mortal):**
 - **Mecânicas:** Habilidades de investigação básica, Aura de debuff para sobrenaturais próximos (conceito inicial).
 - **Progressão:** Ganhos de reputação com uma organização de caçadores local.
- **6.1.3. Sistemas de Sobrevivência e Horror Pessoal (Básico):**
 - **Máscara:** Implementação inicial da `mask_score` e `humanity_score`. Quebras da Máscara (uso visível de poder sobrenatural perto de mortais) resultam em perda de pontos e reação de NPCs humanos (pânico, denúncia).
 - **Fome da Fera:** Medidor funcional para Vampiros, com debuffs e risco de frenesi.
 - **Permadeath/Ban Temporário:** Implementação do ban temporário para "morte" ou falha catastrófica.
- **6.1.4. Simulação de NPC (Básica):**
 - **NPCs Humanos:** Rotinas diurnas/noturnas simples (trabalho, casa). Reações básicas a eventos (pânico, denúncia).
 - **NPCs Sobrenaturais:** Representantes das facções de Vampiros e Hunters com agendas básicas (patrulhar, coletar informações).
 - **Geração de Caçadores:** NPCs Caçadores de baixo nível são gerados para investigar quebras da Máscara.
 - **Conhecimento e Memória:** NPCs "lembram" de quebras da Máscara e de interações diretas com o jogador.
- **6.1.5. Interação Social e Combate Social (Básico):**
 - **Reputação:** Sistema funcional de reputação com as facções de Vampiros e Hunters.
 - **Diálogo Contextual:** Opções de diálogo básicas com NPCs, com testes de habilidade simples.

- **Combate Social:** Implementação inicial do dano de Vontade e testes de confronto, sem a complexidade da "plateia" ou ambiente ainda.
- **6.1.6. Combate Físico (Básico):**
 - **Sistema:** Combate em tempo real com ataques direcionais e defesa.
 - **Dano Localizado:** Hitboxes básicas (cabeça, tronco, membros) com dano variável.
 - **Destruição de Ambiente:** Alguns objetos destrutíveis (ex: caixas, barris) com fratura pré-calculada.
- **6.1.7. Progressão e Economia (Básica):**
 - **Progressão Horizontal:** Reputação como principal forma de "nível". Desbloqueio de algumas habilidades ou acesso a áreas/NPCs.
 - **Recursos:** Sangue (para Vampiros), dinheiro.
 - **Aquisição:** Drenagem de sangue (com consequências), obtenção de dinheiro de NPCs.
 - **Trocas:** Sistema básico de troca entre jogadores (sem a balança de valor complexa ainda).
- **6.1.8. Tecnologia:**
 - **Unity URP:** Configuração completa do projeto.
 - **SpacetimeDB:** Backend funcional com módulos Rust/C# para gerenciar o estado do mundo, NPCs e mecânicas básicas.
 - **SDK SpacetimeDB para Unity:** Integração cliente-servidor funcional.
 - **Ferramentas de Otimização:** Uso do "Advanced Tools Mega Pack" para LODs e otimização de malhas.

6.2. Fase 2: Expansão Pós-MVP (Primeiras Grandes Atualizações e Expansões)

Após o lançamento do MVP e a validação da visão, o desenvolvimento se concentrará em expandir o conteúdo e aprofundar os sistemas.

- **6.2.1. Expansão de Splats:**
 - Implementação completa de **Lobisomens** (Fúria, Gnose, formas lupinas, territórios, caerns).
 - Implementação completa de **Magos** (Mana, Esferas, Paradoxo, rituais, sistema de LOD de Paradoxo).

- Implementação completa de **Changelings** (Glamour, ilusões, atração de Caçadores escalonada).
- Implementação de **Random Mortals** (com regras de permadeath e arquétipos).
- **6.2.2. Aprofundamento da Simulação de NPC:**
 - Agendas e objetivos mais complexos para NPCs, com priorização dinâmica.
 - Sistema de relacionamentos e memória de NPC mais robusto, incluindo a disseminação de conhecimento entre NPCs.
 - Geração de missões emergentes baseadas nas necessidades e conflitos dos NPCs e facções.
 - Implementação completa de "Cleanup Crews" e "Wanted System".
- **6.2.3. Mecânicas de Sobrevivência e Horror Pessoal (Avançado):**
 - Fome da Fera e Insanidade mais detalhadas, com consequências visuais e de gameplay mais variadas.
 - Sistema de Permadeath/Ban Temporário mais refinado, com diferentes níveis de "morte" e recuperação.
- **6.2.4. Interação Social e Combate Social (Avançado):**
 - Implementação completa da "plateia" e do impacto do ambiente no combate social.
 - Consequências mais profundas para o dano de Vontade.
 - Minigames de diálogo e testes de confronto mais complexos.
- **6.2.5. Expansão do Mundo:**
 - Abertura de mais distritos de Sibiu (ex: bairros industriais, áreas residenciais, submundo de esgotos e catacumbas).
 - Implementação de áreas naturais mais extensas e perigosas ao redor de Sibiu.
 - Introdução de pontos de interesse icônicos da Transilvânia (ex: castelos, ruínas).
- **6.2.6. Economia e Trocas (Avançado):**
 - Implementação completa do Sistema de Balança de Valor para trocas entre jogadores, incluindo a influência social.
 - Mercado negro dinâmico com flutuações de preços baseadas na simulação.

- **6.2.7. Sistema de Conhecimento Onírico / O Sonhar:**
 - Implementação completa do sistema de Conhecimento Onírico, com atividades oníricas durante a inatividade para Magos e Changelings.
 - Feedback de sonhos/visões para o jogador.

6.3. Fase 3: Objetivos de Longo Prazo

- **Expansão Geográfica:** Abrir novas cidades e regiões da Romênia ou Europa Oriental.
 - **IA Avançada:** Pesquisa e implementação de IA mais sofisticada (ex: integração com modelos de linguagem para diálogos mais dinâmicos, uso de embeddings para memória semântica de NPC), visando tornar os NPCs indistinguíveis de jogadores em termos de comportamento e interação.
 - **Novos Splats/Facções:** Introdução de novas facções sobrenaturais ou tipos de personagens jogáveis.
 - **Ferramentas de Moderação:** Desenvolvimento de ferramentas robustas para GMs e admins para monitorar e intervir no mundo.
 - **Expansão de Plataformas:** Lançamento para outras plataformas (macOS, Linux, Consoles).
-

7. Equipe e Funções

Esta seção descreve a estrutura atual da equipe e os perfis de profissionais que a ExperienceFactory busca para desenvolver "Gala Of The Hidden", alinhados com a visão e o roadmap do projeto.

7.1. Estrutura Atual da Equipe:

- **João Alfredo:**
 - **Função Principal:** Idealizador, Game Designer Principal, Programador Core (Backend/Engine).
 - **Responsabilidades:** Definição da visão do jogo, arquitetura de sistemas, desenvolvimento do core do backend (SpacetimeDB) e da integração com a Unity, prototipagem de mecânicas chave.

7.2. Perfis Buscados (Fase MVP e Crescimento Inicial):

A ExperienceFactory busca indivíduos apaixonados por jogos inovadores, com experiência relevante e alinhados com a filosofia de design de GotH.

- **Programadores (Unity / C#):**

- **Experiência Necessária:** Sólida experiência em desenvolvimento de jogos com Unity e C#.
- **Conhecimentos Desejáveis:** Familiaridade com URP, otimização de performance em Unity, sistemas de animação, UI/UX. Experiência com desenvolvimento multiplayer ou com o SDK do SpacetimeDB é um grande diferencial.
- **Responsabilidades:** Implementação de mecânicas de gameplay no cliente (Unity), desenvolvimento de UI, integração de assets, otimização de performance no cliente.

- **Programadores (Backend / Rust / C#):**

- **Experiência Necessária:** Proficiência em Rust ou C#.
- **Conhecimentos Desejáveis:** Experiência com SpacetimeDB, bancos de dados em tempo real, sistemas distribuídos, arquitetura de jogos multiplayer, desenvolvimento de IA baseada em regras.
- **Responsabilidades:** Desenvolvimento de módulos de lógica de jogo (reducers) no SpacetimeDB, aprofundamento da simulação de NPC, otimização do backend.

- **Artistas 2D/3D (Estilizado / Expressionista):**

- **Experiência Necessária:** Portfólio demonstrando proficiência em arte 2D e/ou 3D com foco em estilos estilizados, neo-góticos e expressionistas (similar a Don't Starve, Vault Comics).
- **Conhecimentos Desejáveis:** Modelagem 3D otimizada para jogos (low-poly com texturas ricas), texturização (Substance Painter/Designer), rigging e animação, criação de VFX estilizados, iluminação e composição de cena.
- **Responsabilidades:** Criação de personagens, ambientes, objetos, efeitos visuais e elementos de UI que se alinhem com a direção de arte única de GotH.

- **Game Designers (Sistemas / Conteúdo Emergente):**

- **Experiência Necessária:** Forte compreensão de design de sistemas, com foco em jogos sandbox, simulação e narrativa emergente.
 - **Conhecimentos Desejáveis:** Experiência com design de MMORPGs, sistemas de economia, progressão horizontal, IA de NPC, design de missões emergentes. Familiaridade com o universo New World of Darkness é um bônus.
 - **Responsabilidades:** Refinar e expandir os sistemas de gameplay, balanceamento, design de mecânicas de interação social, criação de "sementes de plot" e eventos emergentes.
- **Escritores / Narradores (Lore Emergente):**
 - **Experiência Necessária:** Habilidade em criar lore rica e coerente, com foco em horror pessoal, intriga e mistério.
 - **Conhecimentos Desejáveis:** Experiência com escrita para jogos (diálogos, descrições de itens, eventos), familiaridade com o universo New World of Darkness, capacidade de criar narrativas que emergem de sistemas.
 - **Responsabilidades:** Desenvolver a lore de Sibiu e do mundo sobrenatural, criar backgrounds para NPCs e *splats*, escrever diálogos contextuais e descrições que aprofundem a imersão.

7.3. Cultura da Equipe:

A ExperienceFactory busca construir uma equipe colaborativa, apaixonada e inovadora, que valorize: * **Autonomia e Responsabilidade:** Cada membro é encorajado a ter propriedade sobre suas áreas. * **Experimentação e Iteração:** Abraçar a natureza experimental do desenvolvimento de jogos sandbox. * **Comunicação Aberta:** Transparência e feedback construtivo. * **Paixão por Jogos:** Um amor genuíno por criar experiências de jogo únicas e impactantes.

8. Considerações Técnicas Adicionais

Esta seção detalha aspectos técnicos específicos para a equipe de desenvolvimento, com foco em otimização, direção de arte e ferramentas, garantindo que a visão de "Gala Of The Hidden" seja tecnicamente viável e performática.

8.1. Otimização de Performance (Unity Client):

A fluidez e a capacidade de rodar em hardware mais modesto são pilares fundamentais. As seguintes técnicas serão empregadas:

- **Levels of Detail (LODs):**

- **Mecanismo:** Cada modelo 3D terá múltiplas versões com diferentes níveis de detalhe (contagem de polígonos e complexidade de textura). O motor renderiza a versão mais detalhada para objetos próximos e versões progressivamente mais simples para objetos distantes.
- **Implementação:** Utilização do sistema de LOD da Unity, com auxílio de ferramentas como o "Advanced Tools Mega Pack" para geração automática de LODs.
- **Impacto:** Reduz significativamente a carga de processamento da GPU e da CPU, especialmente em cenas com muitos objetos.

- **Occlusion Culling:**

- **Mecanismo:** O motor não renderiza objetos que estão ocluídos (escondidos) por outros objetos (ex: edifícios, paredes).
- **Implementação:** Configuração e bake do Occlusion Culling na Unity para ambientes estáticos.
- **Impacto:** Reduz o número de draw calls e a complexidade da cena a ser renderizada em tempo real.

- **Frustum Culling:**

- **Mecanismo:** Apenas objetos dentro do campo de visão da câmera (frustum) são renderizados.
- **Implementação:** Nativo da Unity, mas a organização da cena e a hierarquia de objetos podem otimizar seu desempenho.

- **Mesh Combine / Static Batching:**

- **Mecanismo:** Combina múltiplos objetos estáticos (que não se movem) em uma única malha, reduzindo o número de draw calls.
- **Implementação:** Utilização da funcionalidade de Static Batching da Unity e/ou ferramentas como o "Advanced Tools Mega Pack" para combinar malhas.
- **Impacto:** Melhora o desempenho da CPU ao reduzir a comunicação com a GPU.

- **Otimização de Materiais e Shaders:**
 - **Mecanismo:** Uso de shaders eficientes e materiais que minimizam cálculos complexos (ex: evitar múltiplos passes de renderização, reduzir o número de texturas por material).
 - **Implementação:** Desenvolvimento de shaders otimizados no Shader Graph do URP, com foco no estilo "Neo-Gothic Expressionist Comic Art" que naturalmente requer menos complexidade fotorrealista.
 - **Impacto:** Reduz a carga de processamento da GPU.
- **Otimização de Partículas e VFX:**
 - **Mecanismo:** Uso do VFX Graph da Unity para criar efeitos visuais performáticos, com controle sobre a taxa de spawn, tamanho e complexidade das partículas.
 - **Impacto:** Garante que os poderes sobrenaturais e os efeitos de combate sejam visualmente impactantes sem causar quedas de frame rate.

8.2. Direção de Arte e Implementação Técnica:

O estilo "Neo-Gothic Expressionist Comic Art" é a espinha dorsal visual de GotH, combinando a atmosfera sombria com uma estética estilizada e otimizada.

- **Estilo Visual:**
 - **Neo-Gothic:** Arquitetura inspirada em Sibiu (1940s), com elementos góticos, barrocos e Art Deco. Paleta de cores predominantemente escura, com tons dessaturados e contrastes dramáticos.
 - **Expressionist:** Uso de iluminação e sombra para criar drama e distorção, enfatizando o horror pessoal e a atmosfera de mistério. Ângulos de câmera dinâmicos e composições que evocam a sensação de um pesadelo.
 - **Comic Art:** Linhas de contorno fortes para personagens e objetos importantes, texturas com um toque de pincelada ou hachura, e um certo nível de abstração que permite a otimização.
- **Implementação Técnica do Estilo:**
 - **Shaders Personalizados:** Desenvolvimento de shaders no URP para:
 - **Cel-Shading / Toon Shading:** Para dar um aspecto de desenho animado ou quadrinhos, com transições de luz e sombra mais abruptas.

- **Outlines (Contornos):** Para enfatizar personagens e objetos importantes, dando-lhes uma presença visual forte.
- **Efeitos de Pós-Processamento Estilizados:** Vinhetas, aberração cromática sutil, grão de filme, e talvez um filtro de cor que reforce a paleta gótica.
- **Iluminação:** Uso de iluminação volumétrica e luzes pontuais para criar focos de luz e sombras dramáticas, realçando a atmosfera.
- **Modelagem:** Modelos 3D com contagem de polígonos otimizada, mas com detalhes expressivos na silhueta e na forma.
- **Texturização:** Texturas com um toque artístico, que complementam o cel-shading e os contornos, evitando o fotorrealismo.

8.3. Ferramentas de Desenvolvimento e Workflow:

- **Controle de Versão:** Git (com LFS para assets grandes) será o sistema de controle de versão principal. Repositórios serão hospedados em plataformas como GitHub, GitLab ou Bitbucket.
- **Gerenciamento de Projetos:** ClickUp para rastreamento de tarefas, bugs e progresso do projeto.
- **Comunicação Interna:** Slack para comunicação diária da equipe.
- **Pipeline de Assets:** Definição de um pipeline claro para importação e otimização de assets da Unity, garantindo consistência e performance.

Conclusão do GDD:

Com a seção de Considerações Técnicas Adicionais, o **Game Design Document (GDD) - Gala Of The Hidden (GotH)** está completo em sua versão inicial. Este documento serve como a base para o desenvolvimento do MVP e como um guia para a expansão futura do projeto.