

GDD PROJETO GAME POBREZA CRUZEIRO DO SUL:

parte com Gemini:

parte 1:

É uma excelente ideia! Fazer o jogo em formato de **texto com escolhas binárias (SIM/NÃO)** simplifica muito o desenvolvimento na Unity (focando quase totalmente em UI e lógica de script) e, ao mesmo tempo, aumenta o foco na **narrativa e no peso emocional** das decisões.

Vamos adaptar o Game Design Document (GDD) para esse formato, simplificando as mecânicas para a estrutura SIM/NÃO.

Game Design Document: [Título Provisório do Jogo] (Formato Texto/SIM ou NÃO)

1. Visão Geral e Conceito

Seção	Descrição
Título Provisório	Ex: O Voto Diário, Simplesmente Sobreviver, Escolhas Sem Saída
Plataforma	PC/Web (Unity Build)
Gênero	Novela Visual/Narrativa Interativa (Foco em Decisões Morais)
Mecânica Principal	Leitura de um cenário/problema e escolha de uma entre duas opções limitadas (SIM ou NÃO), que têm impacto direto nos Status do jogador.
Objetivo do Jogo	Usar a simplicidade do SIM/NÃO para dramatizar a falta de boas opções. O jogador deve sobreviver ao maior número de Ciclos (dias/turnos) possível.

2. Gameplay Core: O Ciclo de Decisão

O jogo se desenrola em uma série de ciclos (turnos), onde o jogador enfrenta um problema e deve resolvê-lo rapidamente.

Ação por Ciclo

1. **Apresentação do Status:** O jogador vê seus medidores de Status (Dinheiro, Saúde, Moral).
2. **Apresentação do Dilema:** Uma caixa de texto exhibe uma situação ou pergunta (ex: "Seu filho(a) precisa de um remédio caro. Você usa o dinheiro do aluguel?").
3. **Escolha:** O jogador clica em **SIM** ou **NÃO**.
 - **SIM** geralmente é a solução de **curto prazo** (alivia a dor imediata, mas gera problema futuro).
 - **NÃO** geralmente é a solução de **longo prazo/sacrifício** (mantém a estabilidade, mas causa sofrimento imediato).
4. **Consequência:** Uma mensagem curta aparece, detalhando como a escolha afetou os Status. (Ex: *Saúde +20, Dívida +R\$300*).
5. **Próximo Ciclo:** O jogo avança para a próxima situação.

Mecânica de Status Adaptada

Com o foco em SIM/NÃO, a lógica de Status precisa ser muito clara:

Status	Função	Impacto SIM/NÃO
Dinheiro (R\$)	Recurso primário. Se esgotado, aumenta Dívida ou causa penalidade em Moral.	SIM/NÃO define se o dinheiro é gasto ou poupado.
Saúde/Bem-Estar	Representa a condição física. Se baixo, o jogador não pode trabalhar/estudar. (Se zerar = Game Over) .	Geralmente aumenta com SIM (gastar para cuidar) e diminui com NÃO (sacrificar o cuidado).
Moral/Esperança	Representa a força mental. Se esgotada, o jogador desiste. (Se zerar = Game Over) .	Pode diminuir com escolhas que geram culpa ou sacrifício (NÃO) ou aumentar com um alívio temporário (SIM).

Dívida	Penalidade de longo prazo. Aumenta o custo de vida e pode levar ao Game Over .	Aumenta com SIM (empréstimos, gastos impulsivos) e pode diminuir com NÃO (pagamento, sacrifício de Status).
---------------	---	---

3. Estrutura da Cena e Requisitos Técnicos

Cena Única na Unity

A cena deve conter apenas os seguintes elementos UI, dispostos de forma clara:

1. **Painel de Status (Topo/Fixo):** 4 caixas/barras mostrando o valor atualizado de **Dinheiro, Saúde, Moral e Dívida**.
2. **Painel de Narrativa (Centro):** Onde o texto do Dilema aparece. É o foco da tela.
3. **Botões de Ação (Base):** Os dois botões grandes, **SIM** e **NÃO**, que acionam a lógica do script.

Lógica de Script (C#)

O script principal deve gerenciar:

1. **Lista de Eventos:** Um `ScriptableObject` (ou uma lista simples) contendo todos os dilemas. Cada dilema deve ter:
 - o `string` Pergunta;
 - o `struct` ConsequênciaSIM (ex: `+50 Dinheiro, -10 Moral`);
 - o `struct` ConsequênciaNAO (ex: `-50 Dinheiro, +15 Saúde`);
 - o `int` Ciclo/Dia de aparição.
2. **Transição de Ciclo:** A cada clique (SIM ou NÃO), o script aplica as mudanças nos Status e carrega o próximo dilema na tela.

Dicas Visuais (Para Manter a Simplicidade)

- **Fonte:** Use uma fonte clara e legível.
- **Cor dos Botões:** Mantenha os botões SIM e NÃO neutros (não verde/vermelho), para que o jogador não associe "verde = bom" e "vermelho = ruim", reforçando que **não há escolha fácil**.

Este formato de texto e SIM/NÃO é ideal para focar na mensagem do seu trabalho. Concentre seus esforços em escrever **dilemas impactantes** que explorem as trade-offs reais da pobreza.

parte 2:

É fundamental definir claramente as condições de vitória e derrota, especialmente em um jogo focado em narrativa e no peso das decisões. Em um jogo sobre pobreza, a "vitória" não deve ser um final feliz hollywoodiano, mas sim um **alívio temporário ou a quebra de um ciclo**, o que é mais realista e impactante.

Aqui está um detalhamento de como o jogador pode **ganhar** (atingir um final positivo/neutro) ou **perder** (terminar o jogo) no seu formato SIM/NÃO:

Condições de Fim de Jogo (Vitória e Derrota)

1. Condições de Derrota (Game Over)

A derrota deve ser a consequência direta do esgotamento dos recursos ou da esperança, reforçando a ideia de que a pobreza é implacável. O jogo deve terminar **imediatamente** quando um destes Status críticos chegar a zero.

Status Zero	Resultado de Derrota	Mensagem para o Jogador
Saúde/Bem-Estar →0	Falência Física. O personagem fica gravemente doente, incapaz de trabalhar ou buscar ajuda, levando ao colapso.	"O corpo não aguentou. Sem tratamento ou descanso, você não pode continuar. Fim de Jogo. "
Moral/Esperança →0	Falência Mental. O peso das escolhas e sacrifícios se torna insuportável, e o personagem desiste de lutar.	"A esperança se esvaiu. O custo emocional foi alto demais para suportar. Fim de Jogo. "
Dívida → Máximo	Execução/Ação Legal. A dívida se torna tão grande que leva à perda de moradia ou de um bem vital.	"Você perdeu o essencial. A dívida acumulada não deu mais margem para manobras. Fim de Jogo. "

Falha em Decisão Crítica	Derrota Imediata. Ocorre em eventos raros e críticos, onde a escolha <i>errada</i> tem uma consequência imediata e fatal (ex: recusar ajuda crucial em um momento de pico de crise).	"Sua escolha levou a uma consequência imediata e irreversível. Fim de Jogo. "
---------------------------------	---	---

2. Condições de Vitória (Finais Positivos ou Neutros)

Em vez de uma única "Vitória", é mais apropriado ter **Múltiplos Finais Neutros ou Positivos (Relativos)**, que demonstrem a superação de uma fase difícil. O jogador "vence" ao atingir um objetivo de longo prazo e encerrar o ciclo de forma estável.

Finais Baseados em Ciclo/Duração (O Mais Simples)

A maneira mais fácil de implementar um final em um projeto pequeno é baseada no tempo de jogo.

1. **Sobrevivência de Ciclo:** Se o jogador conseguir completar um número predefinido de ciclos (ex: **30 Dias/Turnos**) sem que nenhum Status crítico chegue a zero.
 - **Mensagem:** "Você sobreviveu ao mês. As dificuldades continuam, mas você garantiu a próxima etapa. **Você quebrou este ciclo.**"
 - **Variação:** A mensagem final pode mudar de acordo com o Status restante (Ex: Se termina com Dívida alta, a mensagem é mais sombria; se termina com Moral alta, é mais esperançosa).

Finais Baseados em Metas (Mais Complexo, mas Mais Impactante)

Requer que o jogador gerencie ativamente um Status de "Progresso".

1. **Final da Estabilidade Financeira:** O jogador consegue manter o Status **Dinheiro** acima de um limite alto (Ex: R\$5.000) por 5 ciclos consecutivos E a **Dívida** em zero.
 - **Mensagem:** "Com muito sacrifício, você atingiu a estabilidade e conseguiu guardar uma reserva. Pela primeira vez, você tem uma pequena margem de segurança."
2. **Final da Educação/Qualificação:** O jogador prioriza consistentemente a escolha de "Conhecimento/Habilidade" (que você pode adicionar como um 5º Status oculto) e atinge o nível máximo (Ex: Nível 10).
 - **Mensagem:** "O esforço valeu a pena. Você conseguiu a qualificação para um emprego melhor e saiu da linha de subsistência. Um novo futuro se abre."

O ideal, para um projeto simples, é focar no **Sobrevivência de Ciclo**. É direto, fácil de programar e cumpre o objetivo de fazer o jogador sentir a dificuldade de simplesmente **manter a cabeça acima da água**.

parte 3:

Ótimo! A força do seu jogo estará na **qualidade e no peso emocional** desses textos. Para maximizar o impacto, cada dilema deve ter uma escolha **SIM** e **NÃO** que sejam igualmente difíceis, sem uma opção claramente "boa".

Aqui estão 10 exemplos de dilemas para as Cartas de Evento, formatados para o seu jogo, incluindo as consequências sugeridas nos seus Status (Dinheiro, Saúde/Bem-Estar, Moral/Esperança e Dívida).

Exemplos de Cartas de Evento (Dilemas)

1. Crise de Saúde Imediata

Cenário (Dilema)	Status Alvo: Saúde/Bem-Estar
Sua filha acorda febril. Ela precisa urgentemente de um medicamento que custa R\$80, o que consumirá quase todo seu dinheiro restante para o transporte da semana.	
Opção 1: SIM (Comprar o remédio)	SIM: Saúde/Bem-Estar ↑ Forte; Dinheiro ↓ Forte.
Opção 2: NÃO (Esperar e tentar remédios caseiros)	NÃO: Saúde/Bem-Estar ↓ Moderado (Risco de piora); Moral ↓ Leve; Dinheiro ↑ Leve.

2. Oportunidade de Renda vs. Dívida

Cenário (Dilema)	Status Alvo: Dívida
Um agiota da vizinhança oferece um pequeno empréstimo com juros altíssimos para cobrir a conta de luz atrasada e evitar o corte. Você aceita?	

Opção 1: SIM (Aceitar o empréstimo)	SIM: Dinheiro ↑ Forte; Dívida ↑ Forte; Moral ↓ Leve.
Opção 2: NÃO (Deixar a luz ser cortada por enquanto)	NÃO: Dinheiro → Neutro; Dívida → Neutro; Moral ↑ Leve (Orgulho); (Pode gerar penalidade futura, ex: -10 Saúde por falta de refrigeração).

3. Sacrifício Pessoal

Cenário (Dilema)	Status Alvo: Moral/Esperança
Você encontrou um trabalho extra, mas ele exige 4 horas de sono a menos por noite durante uma semana, esgotando suas reservas físicas. Você aceita a carga extra?	
Opção 1: SIM (Trabalhar a carga extra)	SIM: Dinheiro ↑ Forte; Saúde/Bem-Estar ↓ Moderado.
Opção 2: NÃO (Recusar o trabalho extra)	NÃO: Dinheiro → Neutro; Saúde/Bem-Estar ↑ Leve; Moral ↓ Moderado (Sentimento de culpa por perder oportunidade).

4. Necessidade de Reparo

Cenário (Dilema)	Status Alvo: Dinheiro e Saúde
O único par de sapatos que você usa para trabalhar está rasgando e causando bolhas nos pés, mas você precisa economizar para a comida de amanhã. Você gasta R\$50 em um par usado?	

Opção 1: SIM (Comprar os sapatos)	SIM: Saúde/Bem-Estar ↑ Leve; Dinheiro ↓ Forte.
Opção 2: NÃO (Continuar usando o par rasgado)	NÃO: Saúde/Bem-Estar ↓ Moderado (Risco de infecção); Dinheiro ↑ Leve (Economia).

5. Dilema Moral

Cenário (Dilema)	Status Alvo: Moral/Esperança
Você encontrou uma carteira grossa no chão. Ninguém está olhando. O dinheiro resolveria seus problemas por semanas, mas há documentos que a identificam. Você a devolve?	
Opção 1: SIM (Ficar com o dinheiro)	SIM: Dinheiro ↑ Fortíssimo; Moral ↓ Forte (Culpa); Dívida → Neutro.
Opção 2: NÃO (Devolver a carteira)	NÃO: Dinheiro → Neutro; Moral ↑ Fortíssimo (Integridade); (Pode gerar evento positivo futuro: o dono te dá uma pequena recompensa).

6. Educação vs. Renda Imediata

Cenário (Dilema)	Status Alvo: Dinheiro e Moral
Um amigo te convida para um curso técnico gratuito que pode garantir um emprego melhor no futuro, mas o curso é no mesmo horário de um bico diário crucial. Você vai ao curso?	

Opção 1: SIM (Priorizar o curso)	SIM: Dinheiro ↓ Moderado (Perde o bico); Moral ↑ Forte (Esperança de futuro).
Opção 2: NÃO (Priorizar o bico)	NÃO: Dinheiro ↑ Moderado; Moral ↓ Moderado (Perde a chance de avanço); Saúde/Bem-Estar ↓ Leve (Cansaço).

7. Sobrevivência vs. Relacionamento

Cenário (Dilema)	Status Alvo: Moral/Esperança
Sua mãe pede dinheiro emprestado para o próprio aluguel. Você sabe que isso pode zerar suas economias e é provável que ela não consiga pagar de volta. Você empresta?	
Opção 1: SIM (Emprestar o dinheiro)	SIM: Dinheiro ↓ Fortíssimo; Moral ↑ Forte (Ajuda à família).
Opção 2: NÃO (Dizer que não pode ajudar)	NÃO: Dinheiro → Neutro; Moral ↓ Forte (Sentimento de abandono); (Pode gerar evento negativo futuro: -10 Moral devido a brigas familiares).

8. Pequena Tentação

Cenário (Dilema)	Status Alvo: Saúde/Bem-Estar
Você está faminto e passou por uma padaria que está jogando fora sobras de pão fresco. O risco de contaminação é pequeno, mas existe. Você pega para comer?	

Opção 1: SIM (Pegar e comer o pão)	SIM: Saúde/Bem-Estar ↑ Leve (Alivia a fome); (Pode gerar evento negativo futuro: -30 Saúde por contaminação).
Opção 2: NÃO (Resistir à tentação)	NÃO: Saúde/Bem-Estar ↓ Leve (Continua a fome); Moral ↑ Leve.

9. Pressão da Dívida (Juros)

Cenário (Dilema)	Status Alvo: Dinheiro e Dívida
O credor da Dívida oferece reduzir drasticamente os juros se você conseguir pagar 30% do valor total hoje. Isso exigiria vender um item de valor sentimental. Você vende o item?	
Opção 1: SIM (Vender o item)	SIM: Dinheiro ↑ Forte (Venda); Dívida ↓ Moderado; Moral ↓ Moderado (Perda sentimental).
Opção 2: NÃO (Manter o item e pagar os juros)	NÃO: Dinheiro ↓ Leve (Pagamento de juros); Dívida → Neutro (Não diminui); Moral ↑ Leve.

10. Riscos no Trabalho

Cenário (Dilema)	Status Alvo: Saúde/Bem-Estar
Seu chefe pede que você realize uma tarefa perigosa sem o equipamento de segurança adequado, mas oferece um pequeno bônus em dinheiro para isso. Você aceita o risco?	

Opção 1: SIM (Realizar a tarefa)	SIM: Dinheiro ↑ Moderado (Bônus); Saúde/Bem-Estar ↓ Forte (Alto risco de acidente).
Opção 2: NÃO (Recusar a tarefa)	NÃO: Dinheiro ↓ Leve (Pode ser descontado ou perder o emprego); Saúde/Bem-Estar ↑ Leve (Segurança).

Lembre-se de que a **consequência negativa** de cada escolha é o que torna o dilema tão forte. A cada clique, o jogador deve se perguntar: "Qual problema eu consigo suportar por mais tempo?"

parte 4 - estratégia da codificação do game:

Excelente decisão! Começar com esses 10 dilemas e a estrutura SIM/NÃO é o caminho mais **inteligente e eficiente** para garantir que você entregue um projeto funcional e impactante a tempo. É o coração do seu **Produto Mínimo Viável (MVP)**.

Próximos Passos Essenciais no Desenvolvimento (Unity)

Com os 10 textos definidos, seu foco deve ser em estruturar a cena e a lógica para que eles funcionem perfeitamente.

1. Estrutura de Dados (C#)

Crie uma forma simples de armazenar esses 10 dilemas. O jeito mais limpo é usar uma **struct ou class em C#** (ou até um **ScriptableObject** se você quiser ser mais avançado) para cada evento:

C#

None

```
[System.Serializable]
public class Dilema
{
    public string Pergunta;

    // Consequências para a escolha SIM
    public int DinheiroMudancaSIM;
    public int SaudeMudancaSIM;
    public int MoralMudancaSIM;
    public int DividaMudancaSIM;

    // Consequências para a escolha NÃO
    public int DinheiroMudancaNAO;
    public int SaudeMudancaNAO;
    public int MoralMudancaNAO;
    public int DividaMudancaNAO;
}
```

Depois, crie uma lista (**List<Dilema>**) no seu script principal para carregar todos os seus 10 eventos.

2. Fluxo do Jogo (GameManager)

Um script chamado, por exemplo, `GameManager` deve controlar:

1. **Inicialização:** Definir os valores iniciais de todos os Status (ex: Dinheiro = R\$300, Saúde = 80, Moral = 70, Dívida = 0).
2. **Ciclo/Turno:** A cada clique (SIM/NÃO), o script avança para o próximo dilema na lista.
3. **Atualização da UI:** Após o clique, o script calcula os novos valores e atualiza os textos/barras na tela.
4. **Verificação de Fim de Jogo:** Após cada atualização de Status, o script verifica se **Saúde**, **Moral** ou **Dinheiro/Dívida** atingiram os limites de **Game Over**.

3. Implementação dos Botões

Conecte os botões **SIM** e **NÃO** do Canvas UI ao seu script `GameManager`:

- O botão **SIM** chama um método (ex: `FazerEscolha(true)`), que aplica as consequências SIM.
- O botão **NÃO** chama um método (ex: `FazerEscolha(false)`), que aplica as consequências NÃO.

Ao se concentrar em fazer esses 10 eventos rodarem de forma fluida, você terá o núcleo do seu jogo pronto. A partir daí, adicionar mais eventos no futuro será apenas uma questão de preencher mais dados na sua lista de dilemas!

Boa sorte com a implementação na Unity! Mantenha a simplicidade e a clareza, e o jogo passará a mensagem que você deseja.