

MATERIAL DE APOIO

Semana do Desenvolvimento de Jogos

PLANO DE DESENVOLVIMENTO DE JOGOS



DO ZERO
AO JOGO

Introdução

Olá, pessoal, aqui é o Guilherme, desenvolvedor de Jogos da Uniday Studio e muito obrigado por estar aqui! Espero profundamente te ajudar a alavancar seus com esse material, tirar as ideias incríveis que você tem em mente e executa-las no formato de Jogos Digitais!

Nesse livro eu vou te explicar o passo a passo não só para você planejar, mas também para ter um PLANO de como desenvolver os seus próprios JOGOS de uma forma assertiva, livre de surpresas e procrastinação no meio do caminho. Use esse livro como seu guia, imprima-o se achar necessário.

É o passo a passo que eu uso pessoalmente em meus jogos de mais sucesso e dezenas de alunos tem replicado com êxito. Mal vejo a hora de ver o que você fará a partir desse conteúdo!

Lembre-se:

O mais importante é nunca parar de Desenvolver Jogos!

Então vamos começar!

Como Planejar seu Jogo?

A verdade é que MUITAS pessoas tem problemas com planejamento, foco, procrastinação na hora de criar um Jogo. Não se sinta mal se você é uma delas, é um problema mais comum do que imagina.

Nesse e-book você terá um guia, não só para planejar seu jogo, mas também para pensar e **elaborar a IDEIA** e também para ter clareza quanto ao **passo a passo** do projeto. Em outras palavras, você pode (e deve) seguir esse guia mesmo se nem sequer tiver ideia do que vai criar.

Mas antes...

É MUITO importante você saber de uma coisa antes de começar. Algo que te contaram errado a vida toda. **Existe um “conto de fadas”** no desenvolvimento de jogos que prega que você vai pensar na ideia, montar um GDD (significa **Game Design Document**, é um “roteiro” com tudo que terá no jogo para você desenvolver) e só depois que ele estiver pronto, trancá-lo as sete chaves e começar a criar o game.

Isso não é verdade!

Esse “planejamento” de jogos virou algo místico e utópico, que era para ser algo que deveria te ajudar, mas acaba te atrapalhando. Jogos são dinâmicos. Jogadores mudam a todo o momento, o que é popular e legal hoje pode não ser amanhã. Ideias podem (e vão) dar errado. Se seu “planejamento” não é flexível e adaptável a essas mudanças, você já começou no lugar errado.

Os grandes jogos e grandes estúdios de desenvolvimento hoje possuem um planejamento extremamente flexível, exatamente para estarem preparados as mudanças necessárias no percurso.

Então vamos definir (ou redefinir) algumas coisas aqui antes de partir para o passo a passo, combinado?

PRIMEIRO: O planejamento serve, antes de qualquer coisa, **para você não perder seu tempo fazendo algo que não vai dar certo!**

Eu já disse e vou repetir: **ideias podem (e vão) dar errado**. Quanto mais resistente você for a essa afirmação, mais frustrado ficará no final da sua jornada. Entretanto, quanto antes você aprender a testar sua ideia o mais rápido possível e validar se ela realmente vai funcionar, melhor!

As pessoas costumam me dizer que minhas ideias e jogos são ótimos, muito divertidos. Mas eu não sou um guru das ideias. Na verdade, a maioria das ideias que tenho não são boas. O que aprendi a fazer é filtra-las em questão de horas ou minutos e não gastar meu precioso tempo criando outro jogo ruim.

Não é fácil...

Eu costumo dizer que sua ideia de jogo é como um filho para você. E sabe o que todo o pai ou mãe não admite dizer? Exatamente, dizer que seu filho é feio!

Mas sua ideia não é seu filho! É só uma ideia!

Pode ser que ela seja incrível, imbatível, perfeita e impecável na sua cabeça. É como se fosse impossível que ela falhe algum dia, você já pensou em tudo, não é? Bom, tenho que te dizer que eu já vi essa história antes... No mínimo mil vezes.

Mas nenhuma ideia, por si só, pode superar o campo e batalha. A única forma de saber se um barco que você construiu vai afundar quando você coloca-lo no mar é colocando ele no mar! Você pode passar ANOS calculando todos os detalhes, mas se na hora da ação tiver um furinho no casco, milimétrico que você nunca percebeu, seu barco vai afundar!

Saiba testa-la e descarta-la sempre que necessário, sem ressentimentos. Tenho certeza que seu jogo no topo do **Monte Everest** vai te agradecer por isso!

Então a primeira coisa que vamos aprender aqui é a testar e validar ideias, o mais rápido possível!

SEGUNDO: O planejamento **só acaba quando você termina o Jogo.**

Essa aqui é ótima, pois é um erro muito comum de quem vai “planejar um jogo”. Eu acabei de te dizer que você precisa aprender a testar e validar (ou descartar) suas ideias o mais rápido possível. Também te disse que existe uma grande chance da sua ideia não ser tão boa quanto você pensa e você ter que descartá-la.

Então me diga: Qual o sentido de sentar e ficar duas semanas planejando um jogo que a ideia não é boa?

E lembre-se do que falamos na página anterior: sua ideia pode (E VAI) parecer perfeita na sua mente, mas nada supera o campo de batalha! Coloque seu barquinho no mar o quanto antes.

Nós vamos aprender a fazer isso aqui!

TERCEIRO: **O planejamento é o seu passo a passo**, seu guia. Não tente enganar seu próprio trajeto!

Essa é a última coisa que eu preciso te dizer antes de partirmos para o passo a passo. O planejamento não é só um documento que você vai escrever, dar um beijo e guardar em uma pasta. Para falar a verdade, ele nem precisa ser um documento!

Hoje, depois de tantos jogos concluídos, eu já memorizei o que irei te dizer aqui. Então em vários jogos que faço, eu mal escrevo um “planejamento”, simplesmente sigo esse guia.

Em outras palavras, ele não vai só dizer o que seu jogo é...
Ele vai te dar o passo a passo de como você deve desenvolvê-lo!

Guarde isso, é importante!

Então não tente pular etapas, nem fazer “B” antes de “A” só porque você acha que faz sentido. Siga o passo a passo! Se você não o seguir, estará enganando a você mesmo e a mais ninguém. E você não gosta de ser enganado, gosta?

Se você entender esses três tópicos, então pode partir para a próxima página para iniciar seu planejamento!

O Passo a Passo

Vamos para a ação! A partir daqui, você vai encontrar uma lista de etapas que você deve seguir para criar seu jogo. Não pule etapas, como já vimos anteriormente!

Cada parte possui uma **ação necessária** que você deve fazer. Preste bastante atenção em cada uma delas!

E outra coisa importante:

Esse guia foi feito para você seguir durante TODO o desenvolvimento do seu Jogo. Não espere chegar ao final dele sem ter um jogo pronto. Se você fizer isso, então está fazendo algo errado!

1 – Qual a Mecânica Principal?

Chamamos de “*core mechanic*” aquela ação/mecânica que o jogador faz repetidamente o jogo inteiro, do início ao fim, inclusive combinando com outras ações (veremos a seguir). No clássico Super Mario Bros, por exemplo, a *core mechanic* é “pular”. Mario é simplesmente um jogo de Pular.

Você pula o jogo inteiro, para fazer tudo: subir em plataformas, ganhar o jogo (pulando na bandeira ao final da fase), matar inimigos, coletar moedas, quebrar blocos, tudo! No **Do Zero ao Jogo** você encontrará uma explicação em vídeo detalhada sobre a mecânica principal. Pode te ajudar a passar por essa etapa.

Agora responda:

Qual a mecânica principal do seu Jogo? Ou ainda: Seu jogo será um jogo de quê?

Sua resposta:

Descreva em uma frase, em uma linha. Se sua descrição está grande, então está errada. Se tiver coisas demais, então não é a mecânica principal.

Aqui estão alguns exemplos de coisas que NÃO são respostas válidas para esse Jogo:

- Um RPG
- Um Jogo de Corrida de bicicletas
- Um jogo medieval Multiplayer

E aqui estão alguns exemplos de respostas válidas:

- Um jogo de Pular (*Super Mario Bros*)
- Um jogo de Roubar Carros (*GTA IV*)
- Um Jogo de Cordas! (*Caveman Stories*)

Você pode achar um pouco difícil no começo. **Mas essa é a parte mais importante do projeto.** E novamente, é para escrever uma linha, não faça um texto aqui!

Junto à mecânica principal, é preciso responder outra pergunta...

Qual o estilo e câmera do seu Jogo?

São quatro tipos principais de câmeras. Pense na que melhor se encaixa no seu jogo. Se nenhuma delas for a sua, escolha a mais próxima. Importante: você terá que confirmar essa decisão na próxima etapa, será um exercício interessante!

Fixa. (*Dificuldade: fácil*)

A câmera não se move, nem rotaciona, geralmente nem troca de câmera. Exemplo de jogos com a câmera fixa: LEGO Builder's Journey, Clash of Clans, Candy Crush.

Móvel Limitada/Top Down. (*Dificuldade: fácil/médio*)

A câmera se move, mas de forma limitada. Exemplo de jogos: Diablo, Torchlight, Path of Exile. Super Mário Bros, jogos side scroller no geral também se encaixam aqui.

Em Primeira Pessoa (*Dificuldade: médio*)

Câmera tenta simular “a visão da personagem”, posicionada próxima aos olhos, o jogador controla com o movimento do mouse. Exemplo: Battlefield, Call of Duty, Cyberpunk 2077.

() Em Terceira Pessoa (*Dificuldade: difícil*)

Câmera posicionada atrás da Personagem, jogador pode rotaciona-la com o movimento do mouse e forma que ela orbite em torno do protagonista. Exemplo: Uncharted, GTA IV, The Last of Us, God of War.

Escolheu?

Elas foram ordenadas na ordem da mais simples para a mais difícil. No Do Zero ao Jogo você terá acesso a um workshop de câmeras e irá entender o motivo de cada câmera ser fácil, médio ou difícil de fazer.

Mas... não é tão simples assim!

Posso afirmar baseado na minha experiência ensinando jogos que 99,9% as pessoas que estão começando a desenvolver jogos escolhem a câmera baseado em uma dessas duas coisas:

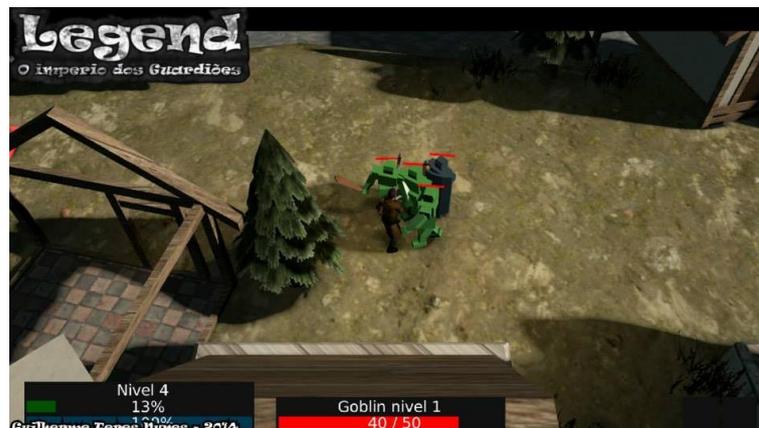
- É sua câmera preferida, os jogos que você mais gosta de jogar são assim e você acha um máximo!
- Você já viu um jogo parecido com o que quer desenvolver usando essa mesma câmera!

Isso é legal e muito romântico, mas não é para isso que estou escrevendo esse conteúdo! Eu quero te ajudar a desenvolver seus jogos DE VERDADE! Então você precisa entender o PORQUÊ de você ter escolhido tal câmera.

Acredite em mim, eu já vi jogos incríveis serem abandonados e jogados na lixeira simplesmente por terem escolhido a câmera errada. Eu absolutamente não quero que você abandone sua ideia só por conta disso e tenho certeza de que você também não quer.

Em 2014, muito antes de eu publicar meu primeiro jogo no steam e certamente antes de eu saber disso tudo que estou te passando HOJE, eu tinha um projeto que eu amava muito. Chamava-se **Legend – O império dos guardiões**.

Não irei entrar em detalhes sobre como era o game, mas eu dediquei meses da minha vida no projeto. Ele estava praticamente pronto para ir ao steam, faltava apenas trocar os personagens “temporários” por modelos de verdade:



Infelizmente, esse jogo incrível que eu coloquei tanta energia nunca viu a luz do dia. O motivo? Perceba que ele tinha uma câmera simples, top down. Mas bem no meio do desenvolvimento eu decidi, sem o menor motivo, trocar para uma câmera em terceira pessoa. Afinal, tinha feito um teste e tudo ficou mais bonito assim! Meus jogos preferidos são em terceira pessoa, então pareceu a decisão certa a ser tomada.

O que eu não sabia naquela época é que isso aumentaria (e aumentou) **exponencialmente** a complexidade do projeto. Depois da mudança, em poucas semanas, que eu provavelmente teria terminado o game se tivesse mantido a câmera top down, acabei desistindo. E o jogo não ficou nem um pouco melhor, de fato, com a câmera nova. Na verdade, o jogo nem existe mais por conta disso.

Então é importante você entender e responder o **porquê** você selecionou sua câmera. Acredite: como vemos em detalhes no Do Zero ao Jogo, uma câmera não é escolhida por acaso. Ela tem um motivo e impacta diretamente na complexidade e qualidade do jogo. A câmera errada pode ser responsável por acabar completamente com a qualidade e a expectativa de sucesso do seu jogo.

Saiba o motivo. Uma câmera em primeira pessoa, por exemplo, pode ser justificada caso sua mecânica principal requeira que o jogador tenha uma mira muito precisa ou observe coisas de muito perto, em um ambiente interno e pequeno. Já a terceira pessoa, pode ser justificada se a *core mechanic* envolver exploração ou a consciência do que está ao redor do jogador.

Ao responder a próxima pergunta, **não assumam nada** só porque você acha ou quer justificar sua escolha emocional de câmera. Tenha clareza e certeza de que você precisa dessa câmera. Se você encontrar dificuldades aqui, na hora de justificar uma câmera, então experimente selecionar uma mais simples, como a fixa ou móvel limitada. Encontrar dificuldades em responder a próxima pergunta é um indicador de que sua escolha de câmera não é necessária.

Por que você PRECISA dessa câmera?

Sua resposta:

E se você tiver escolhido uma câmera em primeira ou terceira pessoa, responda também a pergunta a seguir.

Por que uma câmera fixa ou móvel limitada/top down não vai funcionar?

Sua resposta:

Algumas respostas que **não são válidas** aqui:

- “Porque é um jogo de tiro” (*Alien Shooter é um jogo de tiro com câmera top down!*).
- “Pouca gente gosta de câmera fixa/móvel”. (*Mentira, pouca gente gosta de jogo ruim, mesmo com a câmera que for.*)

Com um pouco de prática, você vai conseguir fazer essa etapa quase que mentalmente. Mas se é sua primeira vez aqui, então eu recomendo que escreva todas as respostas em um caderno e pense sobre elas!

Ação necessária:

Agora que você tem a mecânica principal e a câmera definida, abra seu programa de desenvolvimento de jogos preferidos (como o Blender ou o UPBGE! 😊) e, **usando apenas cubos ou formas parecidas**, crie essa mecânica e câmera.

Não gaste muito tempo aqui, nem continue o planejamento pulando essa etapa. Repetindo:

NÃO PULE ESSA ETAPA!

Você vai sentir tentado a continuar sem essa etapa, ou a fazer coisas demais aqui.

Exemplo:

- **Modelar e/ou animar o personagem**
- **Montar um cenário de testes**
- **Escrever uma história**
- **Demorar mais que algumas horas** (*Se você está fazendo essa etapa há mais de cinco horas, mesmo se for sua primeira vez, está errado!*).
- **Fazer inimigos** (*em alguns casos, a core mechainc tem a ver com inimigos, nesse caso, tudo bem*).

Não faça isso. Não é a hora certa para fazer isso. Pode ser, também, que depois de criar a mecânica principal e a câmera usando cubos, não fique nem um pouco divertido.

Sua mente vai pensar, aqui, que colocando mais coisa, gráficos sons e história vai ficar mais divertido...

Mentira!

Quer dizer, é mentira em 99% dos casos! Mas você não vai querer arriscar esse 1%, né? Eu já criei cubinhos se movendo pela tela que me fizeram gastar **HORAS**

brincando com eles e vendo aquela mecânica funcionando. Se você tem um cubinho se movendo e isso é divertido, seu jogo está no caminho certo. Se aqui não é interessante, então parabéns, você acabou de descartar uma ideia. Volte ao começo e tente outra vez!

Não fique desanimado se acontecer de não ficar interessante, provavelmente você vai tentar umas quatro ou cinco vezes até achar algo divertido. Talvez até dez vezes! Mas o segredo aqui é que você só gastou alguns minutos ou horas em cada uma dessas vezes e com o tempo vai ficar ainda mais rápido nesse processo.

2 – Quais as Sub Mecânicas?

A primeira parte desse plano é a mais importante, por isso a mais longa. Uma vez que você tem a mecânica principal, tudo tende a fluir de forma mais suave.

Agora está na hora de pensar em sub mecânicas.

Basicamente, utilizando a mecânica principal, como você pode combina-la de formas diferentes, talvez inserindo lógica extra, para dar ainda mais desafios ao jogador?

Se você fez um bom trabalho na etapa anterior e ficou horas brincando com seu teste feito com cubinhos, provavelmente pensou em algumas sub mecânicas enquanto se divertia. Mesmo que você não estivesse consciente disso ainda!

Pense no Super Mario Bros, que eu disse que a mecânica principal era Pular.

Como o Mário combina (ou poderia combinar) o Pulo para trazer mais desafios ao Jogador?

- 1 - Pular em uma plataforma no ar.
- 2 - Pular em uma plataforma móvel.
- 3 - Coletar uma moeda no ar.
- 4 - Pular em cima do Goomba (inimigo) para mata-lo.

- 5 - Pular em baixo de um bloco para quebra-lo.
- 6 - Pular na bandeira final para obter mais pontos.

Percebe? Tudo isso que eu citei usa a mecânica principal e a estende para trazer mais desafios. Bom, isso são sub mecânicas em sua forma mais básica!

Agora é sua vez. Não tem mistério e quanto mais simples e ligado à mecânica principal for, melhor! Você irá utilizar essas sub mecânicas durante todo o desenvolvimento do Jogo, inclusive populando as fases do game, então é bom pensarmos em pelo menos cinco delas!

Aqui, a mesma regra da mecânica principal se aplica: Não escreva muito. Se possível, faça em menos de uma linha!

Quais são as Sub Mecânicas do seu Jogo?

<i>Sub mecânica 1 –</i>
<i>Sub mecânica 2 -</i>
<i>Sub mecânica 3 -</i>
<i>Sub mecânica 4 -</i>
<i>Sub mecânica 5 –</i>

Para cada uma delas, pense também em como você pode varia-la dentro do game.

Finalmente, mas não menos importante, pense em pelo menos três combinações diferentes de duas (ou mais) sub mecânicas para criar novos desafios. Exemplo: Volte uma página e veja as sub mecânicas que pensamos para o Mário. Agora vamos pensar em combinações para elas:

- Pular em um inimigo em cima de uma plataforma (1, 4)
- Quebrar um bloco que se transforma em uma moeda (5, 3)
- Quebrar um bloco que adiciona um inimigo (5, 4)

Agora é a sua vez!

Combine ao menos três sub mecânicas do seu Jogo:

Combinação 1 –

Combinação 2 -

Combinação 3 -

Novamente: perceba que deixei pouco espaço para você escrever. Se estiver complexo demais, então tem algo errado!

Com o tempo você terá cada vez mais facilidade de fazer isso!

Ação necessária

Agora é hora de testar e implementar suas sub mecânicas! As mesmas regras da ação anterior (da mecânica principal) se aplicam aqui. Então foque no básico!

Se você chegou a essa segunda etapa, as chances de você descartar a ideia já são menores, mas ainda existem. De qualquer forma, está tudo bem você querer modelar (ou baixar) uma primeira versão do personagem aqui. Só não se empolgue demais, nem gaste mais tempo do que o necessário!

Não tem segredo, crie suas sub mecânicas usando formas básicas, se alguma delas precisar de um inimigo, ou de um *asset* específico, não os crie ainda, faça um modelo temporário. Pode ser que você tenha que descartar ou mudar alguma sub mecânica no caminho, leve isso em consideração.

Lembre-se: precisa ser divertido/bom de jogar!

Se durante essa etapa seu “cubinho que se move” da anterior ficar chato, pode ser um indicador de duas coisas:

PRIMEIRO: Suas sub mecânicas não são divertidas ou não combinam com a mecânica principal!

Muito comum! Você se empolga demais e quer colocar várias coisas que acha legal, mas acaba perdendo o foco, saindo da mecânica principal no meio do caminho. A solução caso tenha caído aqui é repensar as sub mecânicas para aproxima-las da principal.

Ou...

SEGUNDO: Sua mecânica principal era na verdade... uma sub mecânica!

Esse aqui também é muito comum. Na verdade, é provável que isso aconteça com você nas primeiras vezes que seguir esse passo a passo. Até você internalizar os conceitos de core mechanic (reveja as aulas sobre isso no Do Zero ao Jogo, vão te ajudar!).

O que acontece aqui é o seguinte: Você escolheu e implementou uma mecânica na etapa anterior, mas aquilo na verdade era uma sub mecânica de outra mecânica, mas você ainda não conseguiu enxergar isso. Se esse for o caso, você vai ter dificuldades em pensar em mais sub mecânicas para ela e até mesmo para desenvolvê-la.

A solução aqui é voltar na primeira etapa, **simplificar e tentar quebrar sua mecânica principal em peças menores**, até você encontrar o ouro! Que é a verdadeira core mechanic.

E se ainda sim não estiver bom?

Destarte a ideia.

Novamente, sua mente vai achar que se você adicionar mais coisas e colocar um gráfico bonito, sons, história, vai melhorar.

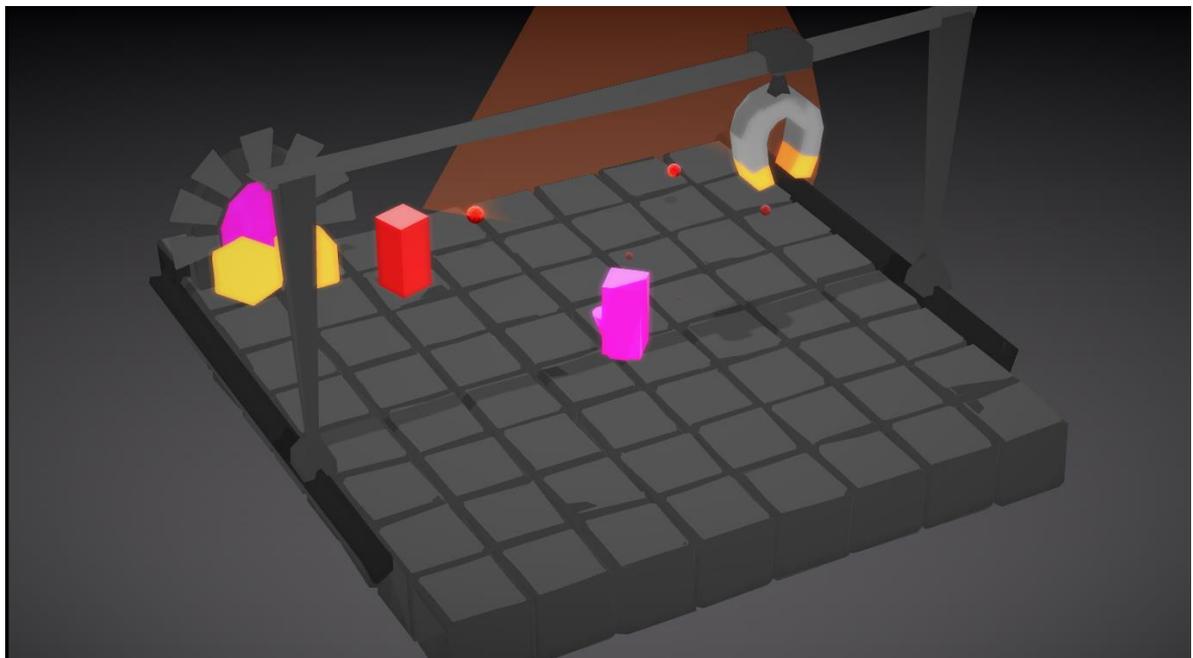
FUJA DISSO! ☺

É mentira. Não gaste muito tempo aqui, a não ser que dê certo!

Não fique triste se você tiver que descartar sua ideia aqui. Lembra do The Looper's Path, meu jogo que ficou em quinto lugar MUNDIAL na Ludum Dare (uma das maiores competições de desenvolvimento de Jogos do Mundo)?

Pois é...

Eu tive que falar duas vezes nessas duas primeiras etapas até conseguir acertar na mecânica. Dá uma olhada em como ele era antes, na primeira tentativa:



Eu abandonei a ideia exatamente aqui. Mas o que eu quero que você perceba é que ele era praticamente feito de cubos.

Na próxima etapa eu vou incluir outra captura de tela, do mesmo jogo, mas já com a mecânica definida e na hora do level design. Você verá, novamente, que ele era cheio de cubos! 😊

Importante:

A partir dessa etapa, a quantidade de programação necessária para prosseguir vai diminuir DRÁSTICAMENTE. A partir daqui, pode ser até mesmo que você nem toque mais na programação, inclusive (apesar de não ser uma regra).

Se isso não acontecer com você, é porque você deixou passar partes importantes do jogo na hora da core mechanic e sub mecânicas.

Um erro comum, por exemplo, é não pensar em Puzzles, armadilhas, interação com NPCs, etc, na hora da mecânica principal ou sub mecânicas. Isso tem a ver com Mecânica, não com o restante do Jogo.

Se não pensou nisso até agora, não tente adicionar de última hora. Vai mais te atrapalhar do que te ajudar. Se você executou corretamente as duas etapas anteriores e já tem um “mini jogo” (mesmo que na forma de alguns vários cubos se movendo) super divertido, mantenha-o até o fim e deixe o que esqueceu para o próximo game. Está tudo bem!

NOTA: Você vai se sentir MUITO tentado a adicionar mais coisas a partir daqui. Não faça isso, agora é hora de virar a página da mecânica, siga o passo a passo!

Ah, finalmente, mas não menos importante... A partir daqui, tudo será uma **ação necessária**. Outra coisa que pode ser interessante chegando aqui é você já fazer os menus do jogo! Quase todo mundo deixa isso pro final mas é algo muito importante em um game.

3 – Coloque seus Assets e Mecânicas em Grupos!

Estamos saindo da etapa de mecânica e indo para o Level Design. Para isso, precisamos preparar o terreno!

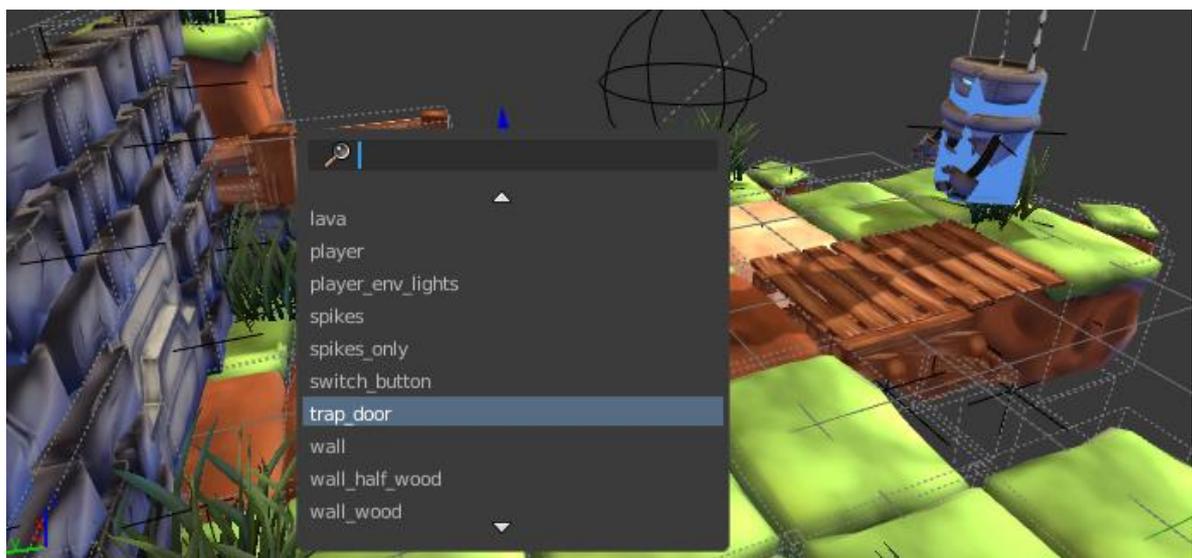
A etapa de criar as fases (a seguir) é muito prazerosa e você deve se sentir jogando the sims ou algum game com um editor de mundos! Com uma lista de tudo que pode adicionar no jogo para deixar sua mente e criatividade ser colocadas em ação!

Para isso, precisamos preparar o terreno.

São duas etapas importantes: Agrupar as Mecânicas e agrupar os Assets.

“Assets” é o nome técnico que damos para os objetos, os elementos 3D ou 2D do seu jogo.

Se estiver usando o Blender/UPBGE, crie uma cena só para isso. Ainda nesses programas, crie um grupo diferente para cada elemento de **mecânica** que tiver criado anteriormente. A ideia é que, na próxima etapa, você tenha tudo em mãos para adicionar em um clique. Como no meu Jogo, **The Looper’s Path**:



Eu posso selecionar uma porta, um alçapão, o player, inimigos, tudo que é mecânica do jogo em um clique, simplesmente adicionando através do menu de grupos do Blender.

Se encontrar alguma dificuldade em fazer isso, também pode ver os vídeos de como eu faço essa organização em grupos lá no Do Zero ao Jogo. Explico tudo em detalhes para você conseguir reproduzir da mesma forma.

Ah, e se você usa Unity ou Unreal, pode fazer algo semelhante utilizando os Prefabs ou Blueprints.

Agora, vamos montar os assets!

Primeiro, faça uma lista de lugares, ambientes que seu jogo vai se passar. Dessa lista, crie outra lista de coisas que existem naqueles lugares. Exemplo:

- Casa de madeira
- Moinho
- Ponte de Madeira
- Parede de Pedra
- Chão de terra

Cada item desta sub-lista é um asset que você deve criar e colocar em um grupo. Você também vai encontrar um tutorial de como fazer tudo isso no Do Zero ao Jogo, se encontrar dificuldades por aqui.

“Devo modelar os Assets agora?”

A resposta é que depende...

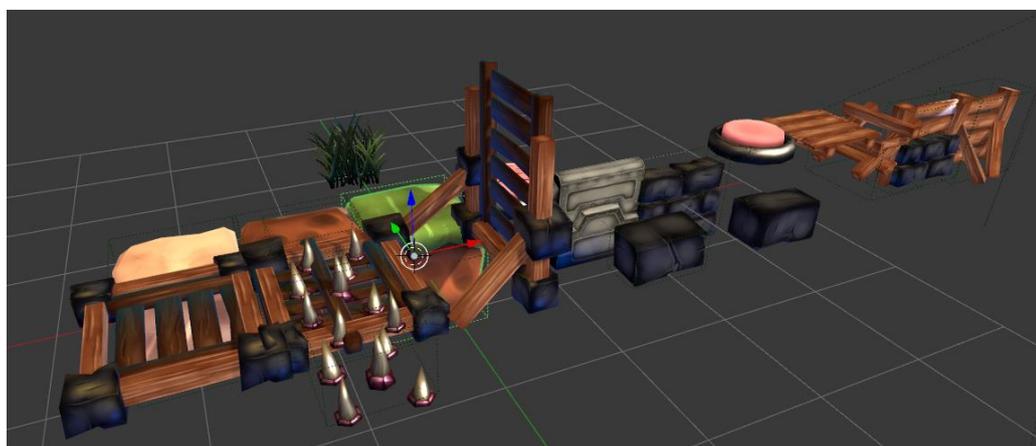
Eu recomendo que faça apenas as formas “básicas” desses assets agora, mesmo que sem texturas. Você vai ver na próxima etapa que isso se chama Block Mesh, ou Grey Box, ou White Box, na verdade cada empresa batiza isso com um nome diferente. Mas significa que os modelos não são os finais, são apenas para prototipar e testar a ideia.

Mas existem duas exceções: Se você for usar modelos prontos, seja feitos por você anteriormente ou baixados da internet (recomendo o itch.io, sketchfab e opengameart para isso), então pode coloca-los desde já. Pois o objetivo e não fazer a arte aqui é poupar o seu tempo e garantir que sua ideia seja realmente boa antes

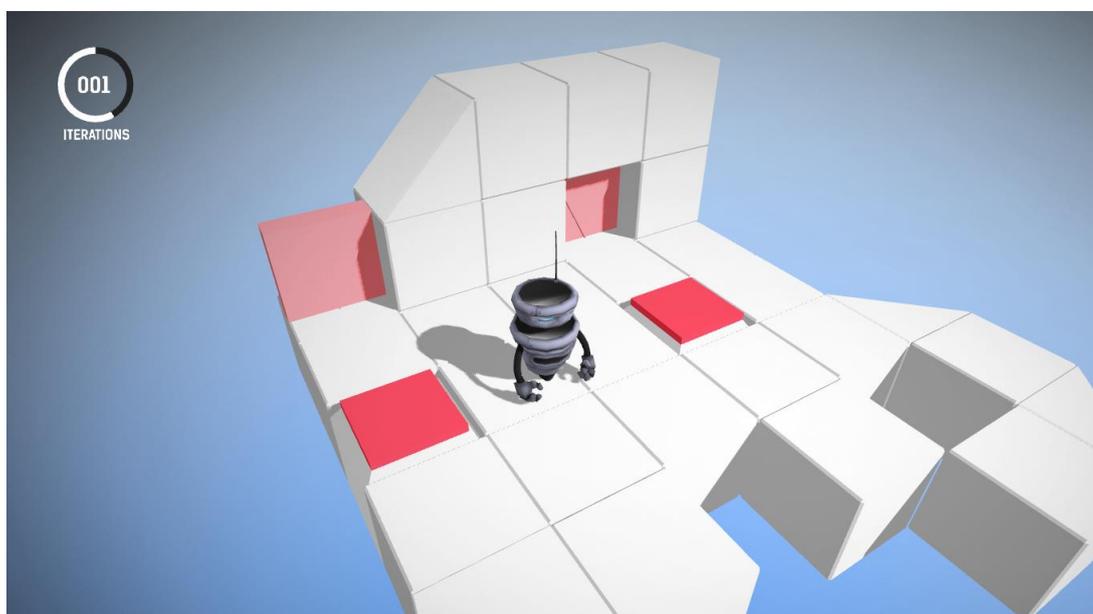
de sair desenvolvendo. Mas, ora, se você já tem os assets prontos, então não estará gastando seu tempo colocando eles, então está tudo bem!

A outra exceção é no caso de estar trabalhando em equipe. Aí você vai criar, de qualquer forma, as formas básicas e enquanto está trabalhando nos níveis (próxima etapa), a outra pessoa da sua equipe pode ir modelando os assets (ou vice-versa).

Uma vez que você cria tudo e organiza em grupos, pode chamar esses grupos de **KITS**. Aqui está uma foto dos KITS de cenário (e algumas armadilha e portas – mecânicas) do The Looper's Path:



Esse jogo foi feito em equipe. Eu modelei todos esses assets enquanto o Daniel, meu programador, fazia os níveis. Mas ele os fez como? Usando cubos! Olha só:



Depois, como se fosse mágica (mas não é), tudo ficou bonito! Tudo isso só foi possível devido a organização em grupos.

4 – Level Design

Já organizou tudo em grupos? Maravilha! Agora crie uma cena para cada level do seu jogo, adicione os grupos do personagem, para que todos esses levels sejam jogáveis e vamos colocar as mãos na massa!

Mas antes, é importante que você saiba de uma coisa aqui...

As chances são de que **te ensinaram a definição de “Level Design” de forma errada!**

Level Design não é construir um cenário bonito. Na verdade, level design não tem NADA a ver com arte. As pessoas confundem isso com o que chamamos de **Environment Art**. São duas coisas diferentes. Environment Art tem a ver com, como o próprio nome já diz, fazer a arte do ambiente, deixar tudo bonito. Mas o **Level Design** não!

Esse está ligado à experiência do jogador, ao caminho que ele irá percorrer pelo mapa, do início ao fim e quais os desafios que ele irá encontrar. Bem como as mecânicas e sub mecânicas que irá utilizar para resolvê-los.

Então preste bastante atenção:

Seu objetivo aqui NÃO é fazer arte!

Na verdade, seu jogo provavelmente vai continuar cheio de cubos quando você terminar essa etapa, mas vai ser completamente jogável.

O objetivo aqui é fazer seu jogo ser jogável do começo ao Fim.

Para cada level, você vai definir onde o jogador começa e onde ele termina e “construir” o caminho até lá. Adicionando os KITS que você criou na etapa anterior e colocando e combinando as sub mecânicas do Jogo!

O ideal é você não colocar todas as sub mecânicas em todos os levels:

Saiba distribuí-las no jogo, do começo ao fim, para que o jogador sempre encontre desafios novos ao jogar seu game. Então, para cada fase que você criou, responda as seguintes perguntas:

Quais sub mecânicas irei utilizar nessa fase?

Sua resposta:

Quais KITS irei utilizar para montar essa fase?

Sua resposta:

Novamente (lembrar disso nunca é demais), não é hora de fazer arte, deixe seu jogo jogável!

Se o projeto estiver grande demais, você vai saber disso bem cedo e poderá cortar alguma coisa sem muitos problemas graves. E não deixe de fazer isso se sentir que será necessário. Seu jogo não vai ser ruim se ele tiver só cinco fases... desde que sejam cinco fases bem feitas! Mas seu jogo com certeza será ruim se ele tiver mil fases mal feitas ou e você nunca termina-lo!

No final dessa etapa, seu jogo deverá estar pronto...

Pronto?

Sim, mas não no sentido da palavra. Ele deverá ser jogável do começo ao fim. Ser jogável do começo ao fim é diferente de ter arte, história ou sons. E é mais importante do que isso!

5 – História e Arte!

Agora que o jogo está “pronto”, vamos deixá-lo bonito e talvez com uma história interessante. Pode parecer loucura, mas chegando aqui, seu jogo já está “praticamente pronto”. Na verdade, o que acontece aqui é que poucas coisas podem “estragar” seu game, então é quase certo de que você irá terminá-lo.

Mas tome cuidado! Isso não significa que você vai quebrar as regras, mesmo e você sentir que pode fazer isso!

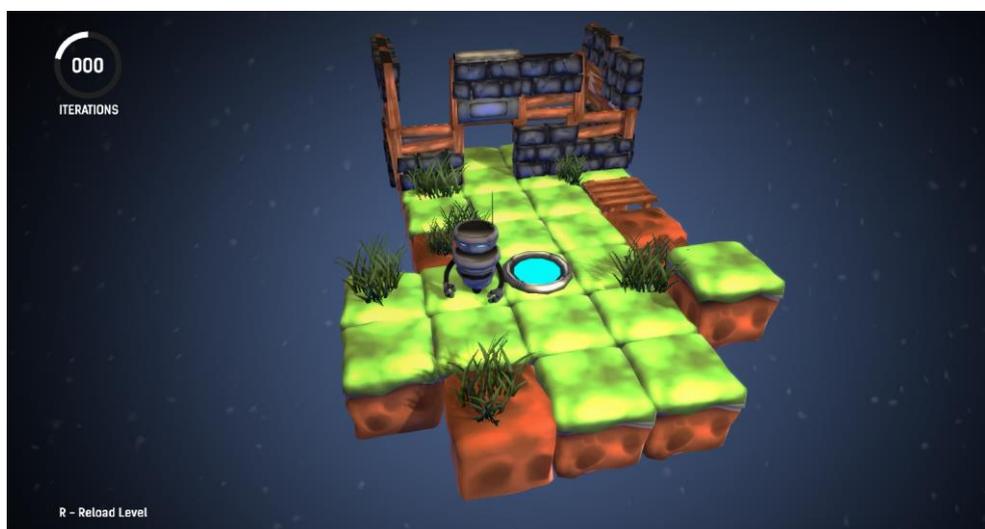
Lembra-se do **Legend – O império dos Guardiões**? Eu estraguei tudo logo no final dessa etapa aqui! Então se você não quiser cometer o mesmo erro, resista a tentação! Não mude as mecânicas nem o level aqui, mesmo se você achar que vai ficar melhor!

NOTA: Você vai achar que vai ser uma boa decisão mudar (ou adicionar) tudo agora, eu tenho certeza disso. Mas fuja, não é!

Agora seu jogo vai ficar “bonito” em questão de horas. Como?

Lembra que dividimos tudo em grupos no passo anterior? Pois é... Agora você só precisa ir à cena dos grupos, onde vai encontrar os “cubinho” e trocá-los por modelos de verdade!

Lembra da imagem do The Looper’s Path que eu postei algumas páginas atrás? Olha só como ela ficou depois disso:



Na verdade essa imagem aí também já está com a próxima etapa aplicada, não achei imagens de como estava antes, mas era bem próximo disso! Só não tinha a graminha, basicamente!

Aqui você também vai encaixar a história do Jogo (se houver).

Mas lembre-se: cutscenes, interações com NPCs, etc, tem que ser parte da mecânica do Jogo.

Se você chegou até aqui e não tem nada disso feito na mecânica, então é um indicador de que seu jogo não precisou desses elementos. E se lembra do que falamos algumas linhas atrás? Exatamente, não é hora de adicionar!

Use templates prontos de diálogos se achar necessário. Divirta-se!

Só tome cuidado para não estragar o trabalho que fez nas etapas anteriores.

6 – Polimento!

Pode parecer mágica, mas não é: seu jogo está praticamente pronto!

Aqui eu gosto de dizer que a regra de ouro é: Se alguém colocar uma arma em sua cabeça e te obrigar a publicar seu jogo AGORA, do jeito que ele está, se você tiver chegado aqui, poderá fazer isso sem medo e sem peso na consciência que ele vai estar bom o suficiente.

Em outras palavras, ele já é jogável do começo ao fim, não tem mais nenhum bloco de teste pelo mapa, nenhum diálogo faltando, não tem nada temporário.

O que falta agora é você passar por todas as fases e deixa-las bonitas! Coloque grama, tochas, elementos visuais que não servem para o gameplay, mas deixam tudo bonito, adicione sons, easter eggs, o que achar mais divertido. Deixe que seu jogo se torne uma obra de arte!

Não pule essa etapa, ela é simples mas prazerosa e eu te garanto que fará toda a diferença para você e seu jogo!

Publique seu Jogo! 😊

Chegou até aqui? Então meus parabéns!

Aperte o botão e lance seu Foguete rumo ao Ceu!

Sabe o que isso quer dizer?

Publique seu Jogo, ele está pronto!

Use o **Gamejolt** ou o **Itch.io** se estiver no começo e não quiser investir nada para colocar o game para download de graça ou pago. Se já tiver publicado alguns jogos, pague a taxa do **Steam** e coloque-o lá, **é onde ele merece estar!**

Não deixe a peteca cair agora!

Muito importante:

Tire um tempo final, agora que o jogo está pronto, para criar uma página bem caprichada na internet para que todos possam ver e baixar sua obra de arte!

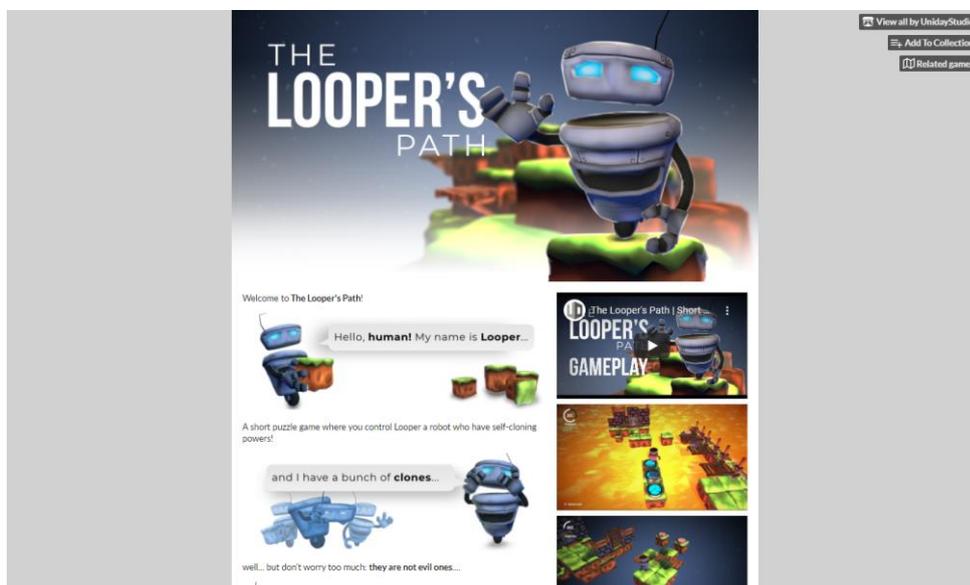
Dê uma olhada na página do **The Looper's Path**:



The Looper's Path

The Looper's Path is a short puzzle game where you control Looper, a robot who has self-cloning powers.

Puzzle



É uma página bem criativa e interessante, com uma historinha no meio, recomendo que você dê uma olhada pelo link, já que não dá para ler muito bem por aqui:

<https://unidaystudio.itch.io/the-loopers-path>

Então dedique um bom tempo para deixar sua página fantástica e tenha por certo que isso valerá a pena! Um dos grandes fatores que contribuiu para o sucesso do The Looper's Path for exatamente esse.

Quer mais?

Esse livro foi escrito com muuuito esforço e dedicação para compilar as melhores e mais importantes partes dessa jornada do desenvolvimento. Tenho certeza que se você aplicar esses ensinamentos, principalmente já que chegou até aqui, vai ter resultados fora da curva!

Inclusive, me mande um e-mail, guilherme@unidaystudio.com.br me contando o que você achou disso tudo e principalmente se conseguiu desenvolver algo! Eu quero ver com toda certeza do mundo!

Mas esse foi só um pedacinho de um conteúdo MUITO maior e completo que eu tenho para te ajudar. E estou falando do **Do Zero ao Jogo**, meu curso que te ajuda mesmo se você não souber NADA a chegar ao topo do seu monte Everest. Ou seja: a desenvolver os Jogos que você sempre quis!

Se quiser conhecer todos os detalhes, ver como o curso funciona, quais aulas estão disponíveis, garantia, etc, basta clicar no botão abaixo:

[Quero Conhecer o Do Zero ao Jogo!](#)

Se tiver qualquer dúvida sobre o Do Zero ao Jogo, sinta-se livre para me mandar no e-mail também! Enfim, boa jornada e eu vejo você por lá!