

A blurred, dark blue background image of a video game controller, likely a PlayStation DualShock 4, oriented diagonally. The controller's buttons and touchpad are faintly visible through the blur.

Unity Crash Course

1

Outline

Game engines

Criando um projeto

Básicos da Unity

Criando códigos dentro da Unity

Organização de um projeto

Versionamento

Objetivo

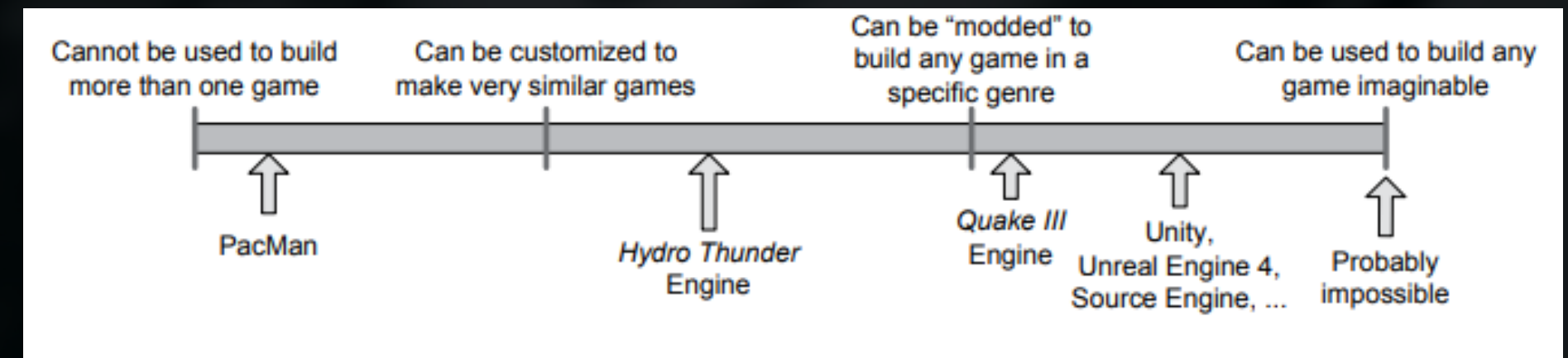
Começar a desenvolver o projeto e mecânicas básicas



Game Engines

O que é uma game engine?

Separação entre sistemas principais e específicos do jogo



Game Engine Architecture (Third Edition), Jason Gregory

Arquitetura *data-driven*: extensível e reutilizável alterando *apenas* dados, não código

Por que uma game engine?

Anos 90: Doom e licenciamento da engine

Reutilizar código (evitar “reinventar a roda”)

Renderização
2D/3D

Gerenciamento
de Assets

IA

Física

Scripting

Controles
(HID)

Animação

Streaming

Áudio

Networking

Gerenciamento
de memória

Profiling
Debugging

Generalização x Otimização

Engines costumam ser criadas com gêneros específicos em mente

FPS **ex. Half-Life**

Renderização eficiente
Controles responsivos
AI para NPCs
Animação fidedigna

Estratégia **ex. Civilization**

Otimização para
renderização distante e com
centenas de elementos

Customização do cenário
pelo jogador

Hardware moderno (+rápido) → Criar engines mais generalistas

Unity



Cross-plataform: desenvolver para diversas plataformas

- Desktop, Mobile, Consoles, VR, Web, TV box

Generalista: jogos 3D e 2D em diferentes gêneros

Real-time 3D para outras áreas

- Cinema, arquitetura, automotiva, governamental

Simplicidade de uso

Comunidade grande e diversos materiais

Versões gratuitas e pagas

- Versão “Pro” gratuita para estudantes



Criando um projeto

Versões do Editor

Tech stream

- Novos recursos mais rápido
- Mais instáveis
- 20xx.1, 20xx.2

LTS (Long Term Support)

- Focadas em estabilidade
- Atualizações para corrigir bugs
- 20xx.3

Importante

- Todos precisam utilizar a mesma versão para editar o projeto
- Não é possível voltar a versão de um projeto, somente atualizar (backup!)

Recomendação

2022.3.xxx
(LTS mais recente)

Pipelines de renderização

Built-in

Mais simples
Mantida para projetos
legados

Não usar

URP

Universal Render Pipeline

Focada em funcionar em
qualquer plataforma

Não possui todos os
recursos que a HDRP

HDRP

High Definition Render Pipeline

Recursos mais complexos
de renderização

Menos plataformas
suportadas (não suporta
mobile e web)

Utilizada aqui



Versionamento

O que é versionamento?

Como permitir que várias pessoas editem o mesmo projeto ao mesmo tempo?

- Como evitar conflitos entre alterações?
- Como evitar editar versões diferentes?



- Sistema de controle de versão distribuído
- Todos possuem uma local versão do projeto
- Hospedagem remota: GitHub ou GitLab
-

Orientações

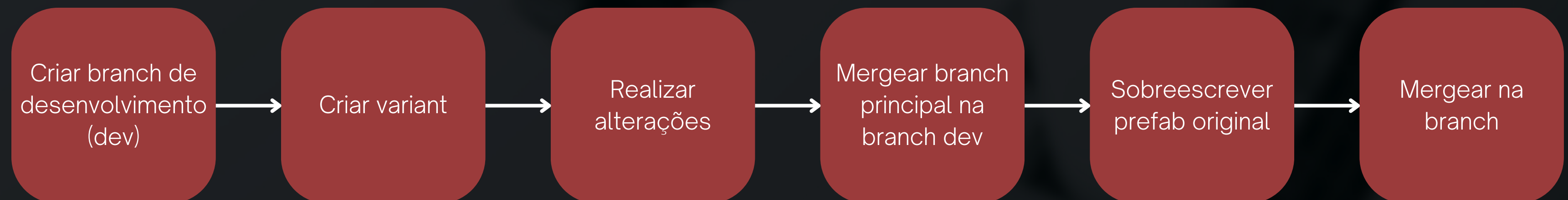
Utilizem um .gitignore

- Alguns arquivos da Unity não precisam e não devem ser colocados no repositório
- Existem templates online

Evitar conflitos - Duas pessoas não podem:

- Editar a mesma cena ao mesmo tempo
- Editar o mesmo prefab ao mesmo tempo

Use variants!





Primeiros passos

O que fazer agora?

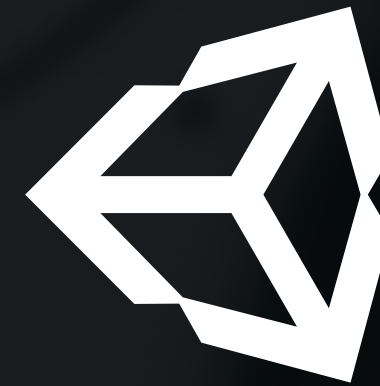
Criar repositório no GitHub

- Adicionar membros do projeto



Criar projeto da Unity

- Decidir versão e pipeline



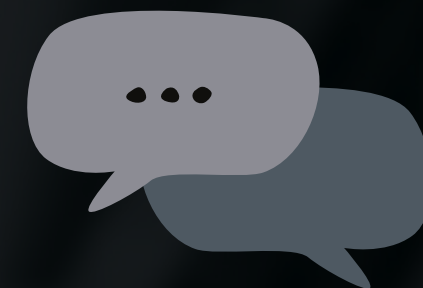
Discutir primeiras mecânicas

- Como implementá-las



Discutir como irão organizar o desenvolvimento

- Evitar conflitos



O que vocês planejam?

Quais temas são importantes para as próximas semanas?



Next: Level Design