# #023 Level Design - Process - Pre-production - Scope

> General size and complexity of a project; validated through prototyping
> From *The Level Design Book*.

**Scope** is the general size and complexity of a project.

In pre-production, you set the initial scope and expectations.

The conventional wisdom in game development is to assume every project begins massively overscoped. Even small reasonable projects hide secret problems. Level design is no different; even a small level may require many iterations and playtests to prove out, if ever. When in doubt, scope down and cut unnecessary non-pillar game aspects.

With more work experience (or in a group, more trust) you eventually become slightly better at estimating project scope. If you're new to game dev, we suggest a 50% rule: imagine the smallest possible project you can... and then mercilessly cut -50% of the scope, for a +50% chance of actually finishing it.

## Feasibility

When working on a large or commercial project, pre-production is a **feasibility** study: can this project be completed with the proposed time and resources? Is this a reasonable idea or a fanciful dream? How doable is this?

The choice of game engine and tools greatly affects which mechanics and systems will feel feasible. For example, an open world driving game in the Doom 3 engine would be very difficult to make, since that engine has no built-in support for large landscapes nor vehicles.

### Engine and tools

Feasibility is more complicated than just using your favorite game engine. Every project has different needs and goals.

- Ownership: is it wise to depend on another company? Many large studios maintain their own engines rather than depending on Unity or Unreal, even though making your own game engine is obviously a lot more work.
- Labor: can you recruit and retain workers? For example, a small studio might avoid Unreal if they can't resist bigger studios poaching their staff with much higher wages.
- Tech-first: can you change the design to suit the tech? Imagine a job that mandated use of the Doom engine; making a shooter would be more feasible than a platformer (Doom has no jump button).

## Pipeline

In game development, a **pipeline** is a sequence of steps to transform a file into a more usable format. Pre-production is when you figure out most of your pipelines.

A build pipeline makes a playable executable to distribute to players, while an art pipeline might focus on converting a 2D image or 3D model into a usable format for the game engine. Basically if you ever see these words -- convert, import, export, compile, bake -- then it is part of a pipeline.

Level design pipelines can vary a lot. Sometimes you may need separate tools or plugins to generate landscapes or construct buildings. Before 2010, it was very common for game engines to require a "map compile" optimization / error-checking step before loading the level file in-game.

## Prototyping

Perhaps the best way to assess feasibility is to make a **prototype,** a small focused project that answers a question.

In 1997, Apple researchers Stephanie Houde and Charles Hill published a famous paper called "What Do Prototypes Prototype?". Houde and Hill define three general categories of prototypes:

- **Role:** how does it meet user needs, what role does it play in their life?
- **Implementation:** is it technically possible, how will we build it?
- **Look and feel:** what will it look like, how does it feel?

When prototyping any of these aspects, you're make what's called a **proof of concept.** It might be a paper prototype, an engine tech demo, or a test level with a player controller.

Layouts and blockouts are also prototypes that seek to answer design questions.

## Vertical slice

When you attempt to combine multiple aspects together into a bigger integrated prototype, you're making a **vertical slice** -- an example "slice" of a finished game with experience design (role), some audio / visual polish (look and feel), and functional technical foundation (implementation).

Making a vertical slice forces a team to prove their pipelines. If they were able to make a good looking slice with their current tools and workflows, then maybe the tools are good enough. Or alternatively, if making the vertical slice felt impossible, then there'd need to be big changes in how the team works.

The vertical slice also helps us gauge project scope. If it took 5 months to make 10% of the project, then it could potentially take 45 months to make the other 90% of the game. (Well, obviously it's more complicated than that, but you get the idea.)

> ***write by Hellinus - 2023.9.19***