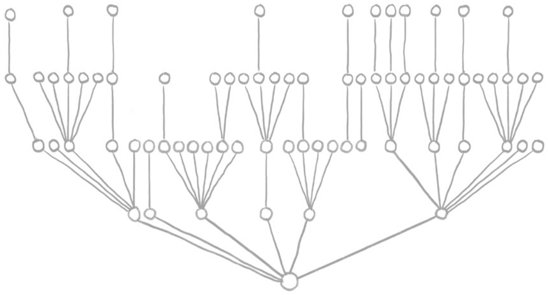
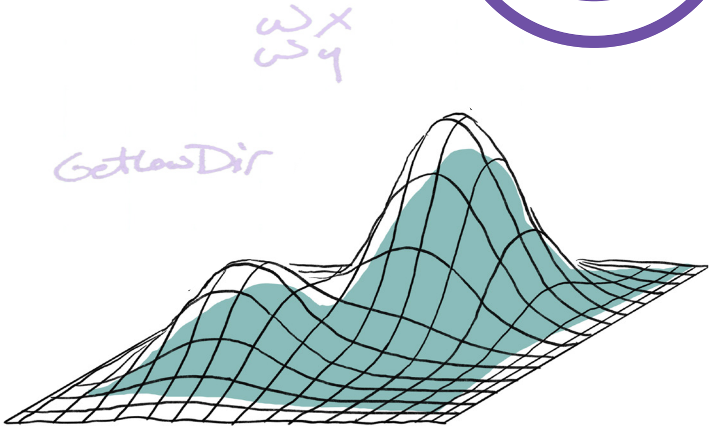




WILL WRIGHT | CHAPTER 11

Iteration and Scoping

TERMS

scripting language (n.) A type of programming language in which you can run your code to test it without compiling it first.

local maximum (n.) The point at which prototyping or testing no longer improves your design.

engineer (n.) The person(s) on a game design team responsible for writing the underlying code. Also referred to as a developer, programmer, or coder.

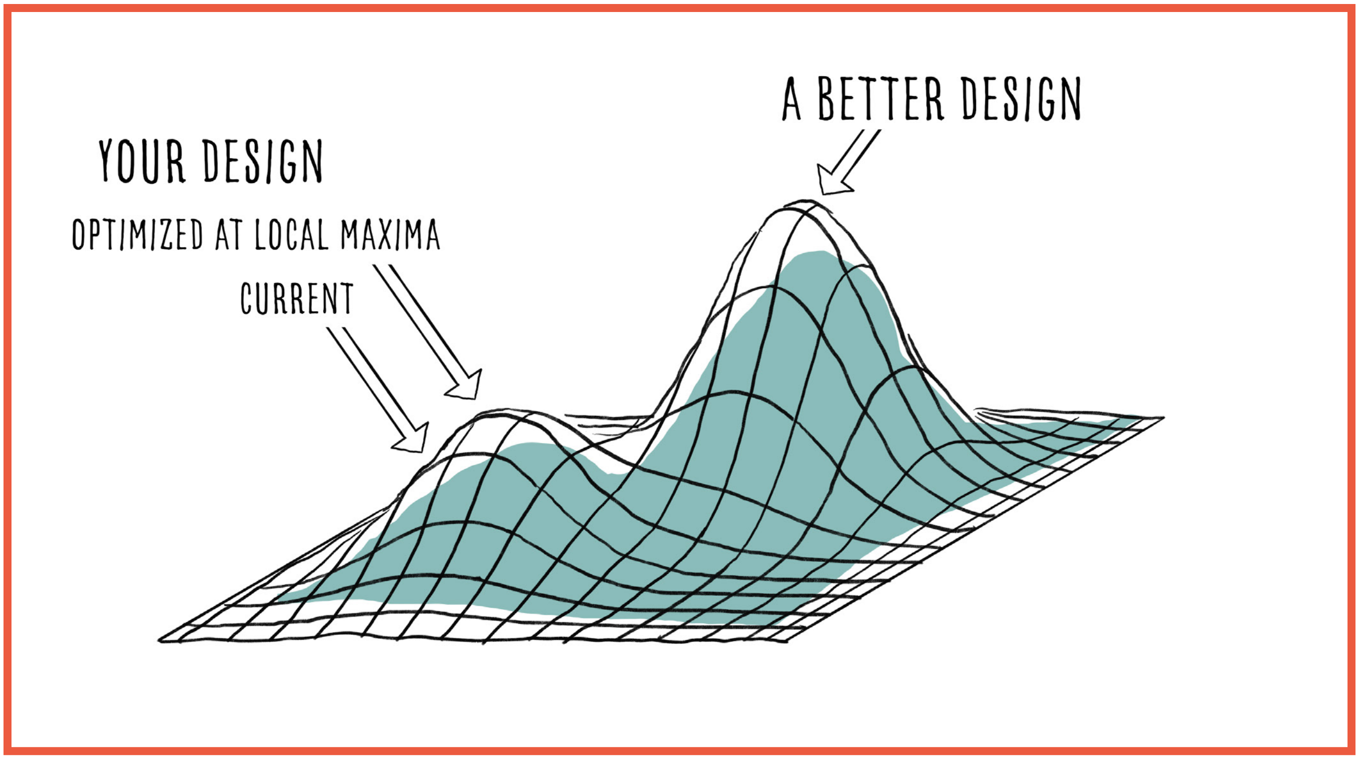
designer (n.) The person or persons on a game design team who determine the player experience and develop game features to produce that experience.

scope (n.) The total number of features a design team can realistically implement, given the time and resources at their disposal.

open-world (adj.) A game environment with multiple active objectives and a large map through which the player travels with full freedom.

In the late stages, always prototype the riskiest features of your design. Successful prototypes that don't fit in your current game can always show up in future designs, just as the city building tool from Will's first game ended up becoming his foundation for *SimCity*. In the late stages of prototyping, there are a few things you should keep in mind.

1. **Recognize when you've reached the local maxima of your design**. To overcome the local maxima, you have to make some dramatic change in direction that takes you onto a new design hill. Often this means ripping out core features of your design, or rendering certain prototypes unplayable.



The best designs have to overcome local maxima two or even three times before they become as good as they can possibly be.

2. **Use "feature triage" to find the scope.** In feature triage, designers assign priority values to the features they'd like to see in the game. Then, engineers assign effort costs indicating the amount of time it will take to execute those features. This produces a prioritized list of features for the team to pursue in order to produce the game without wasting time or resources.

From the perspective of your players, the scope presents itself as the "size" of your game, or the totality of possible interactions available in the experience. Scoping your design is always a balancing act between what's cool and what's possible. Be disciplined and ruthless while scoping, but never lose sight of your original vision. Be confident in your instincts about what is fun.

> "You need to… learn when to fail, and recognize that failure, and sometimes celebrate it."

LEARN MORE

- Play *Downwell* by Ojiro "Moppin" Fumoto or *FTL*: *Faster Than Light* by Subset Games. Consider the kind of experience that can be produced by a game with a relatively small scope. Focus on the small design decisions that produce a lot of fun, and take note of them in your Concept Book.
- Play one of the open-world Rockstar Games, like *Grand Theft Auto V* or *Red Dead Redemption.* Consider the experience of a large-scope game, focusing in particular on the kind of fun that is only possible in an expansive game world. Take note of these in your Concept Book.

ASSIGNMENT

Use the template on the following page to practice Will's discipline of feature triage. As you do, consider how balancing features and tasks helps you set a realistic scope for your game.

FEATURE TRIAGE TEMPLATE

Below is a blank template to use as you apply "feature triage" to your own games during production. List the features, then have the engineers assign an effort cost on a 1-10 scale. Next, have your design team decide how valuable a given feature is in executing the core experience of your gameplay, and assign a value on an A-F or 1-10 scale. Finally, analyze the two values together to prioritize features as either Critical (C), Non-Critical (NC), or Out of Scope (OOS).

FEATURE DESCRIPTION	ENGINEERING COST	DESIGN VALUE	PRIORITY