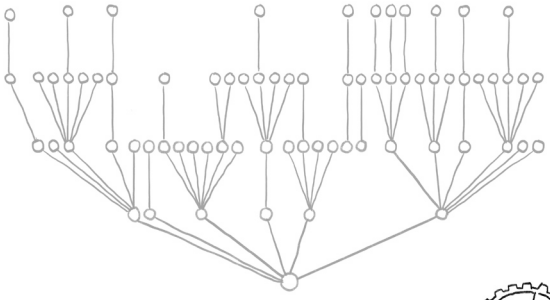
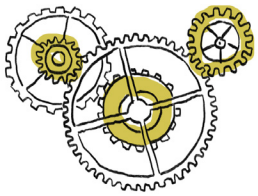





WILL WRIGHT | CHAPTER 06

Design Player-Centered Experiences

TERMS

flow state (n.) A state of complete absorption in a task, characterized by a loss of time.

Dynamic Difficulty Adjustment (DDA) (n.) A game feature in which the system automatically raises or lowers the difficulty of the game to match the player's skill level moment by moment.

nested game loops (n.) A structure of game loops in which smaller challenges represent constituent parts of larger, more complex challenges.

failure state (n.) The condition in which a player doesn't accomplish a stated challenge or objective. Usually found at the end of a game loop and accompanied by negative feedback.

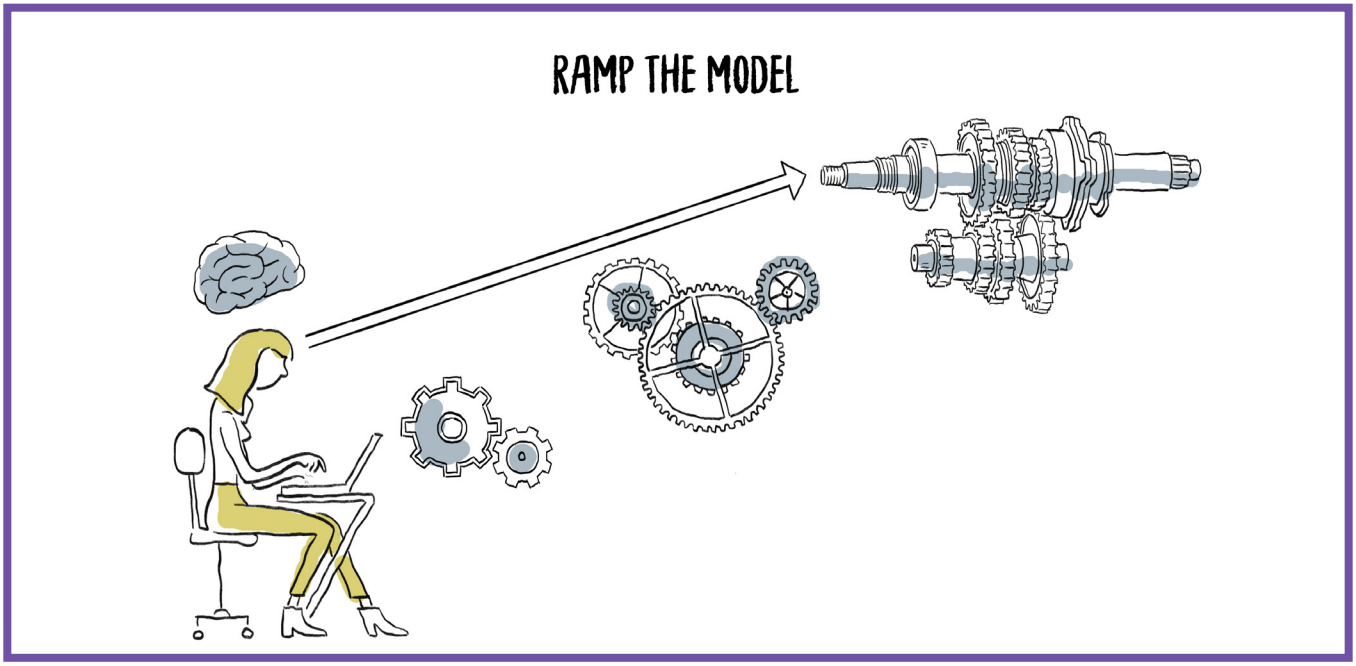
incentives (n.) Elements of your game system which motivate player behavior and foster engagement in your game world.

core game loop (n.) The repeating challenge or interaction progressing the player through objectives to eventually achieve the win condition.

orthogonal game loop (n.) A repeating challenge or interaction that does not advance the player toward the win condition.

As a designer, your primary focus should be the player's experience. Your game rules and system only exist to enable that experience. Will uses this set of strategies to increase the probability that players remain focused on his games throughout their experience.

1. **Put your players in a flow state.** To create it, balance the difficulty of your game against the maximum ability of your player. If your game has fail states, they have to be quick, understandable, and immediately factored into the next game loop. Build small, repeatable game loops within larger, complicated loops. For instance, offer resources to your player after they succeed or fail at a small task. This creates a nested structure of challenge, so the player can always engage with a new task.



2. **Gradually ramp up difficulty and complexity.** Give your player the agency to decide which task to face next, and slowly ramp up the difficulty of that task over time. Build a similar ramp in the complexity of your game. Start with a few options, then start adding more. Don't just drop players into a complete game world.

3. **Design fun, interesting, and humorous fail states.** In the real world, people try to avoid failure at all costs. But in a video game, players are always falling down chasms, driving into walls, and accidently blowing up planets. Strive to make your fail states as enjoyable as possible. Use fast or humorous animations, and don't punish the player too harshly if they don't succeed. If you do this, players will begin to embrace failure as a way to learn about the rules and regulations of the game world.

4. **Use psychological incentives to motivate your player,** rather than relying on material rewards like coins or points. Discovery is a powerful incentive: Offer your player a glimpse of a larger world they can access later in the game. Self-expression is an equally underutilized reward. Let your players pick their own teams and decide their own avenues of attack. Designing a strategy and watching it succeed or fail can also be a strong incentive.

"The really interesting player rewards are the things that they came up with themselves."

LEARN MORE

- Learn more about flow by thinking about when you encounter it in your own life. Take out your Concept Book and write about a recent time you were in a flow state. Think about times you've been so absorbed in a challenging task that time seemed to disappear. How was that flow state achieved? How did it feel? How long did it last? Was it a sustainable state? What could be adjusted to make the experience more sustainable?

- Play a complex strategy game, such as Sid Meier's *Civilization* or *World of Tanks* from Wargaming. Keep an eye out for when you achieve the flow state, and try to discover what factors of the game experience brought you there.

- Read Rami Ismail's essay "Game a Week: Getting Experienced at Failure" on *Gamasutra*. Consider his advice on the value of rapid prototyping, and how it intersects with Will's thinking on the value of failure for both players and designers.

- Play *Hotline Miami* from Dennaton Games. Observe how short the game loops are. Consider how often you are encountering failure on a mission, then factoring the lessons learned in failure into your next attempt at that same mission.

- Play *Celeste* from Matt Makes Games. Consider how the "death" animation, or the fail state on level screens, differs from more punishing games.

ASSIGNMENTS

- Isolate the primary fail state in one of the prototypes from your library. Then, make the game more fun by adding rewarding feedback to that state.
  - If you have a video game of your own in development, test out new, more rewarding feedback for fail states. Add humor to the animations for when your player character takes damage or dies. Make respawn animations immediate, and make the setbacks for player failure as minor as possible.

- Create a simple Dynamic Difficulty Adjustment (DDA) system. Take your favorite prototype from your library and introduce a way to scale the difficulty of a task to match the player's skill. Build in a simple system that evaluates the player's success and reduces the challenges they face when they are behind. They should also face greater difficulty when they are ahead. Add the new prototype to your library.