Game Concept: "Last Light"

2D atmospheric puzzle-platformer.
You play as a small character carrying a fading lantern. You must reach safe zones before the light runs out.

Game Production Pipeline

Visual Pipeline Diagram

IDEA

↓

PRE-PRODUCTION

 - Vision

 - Mini GDD

 - Core Prototype

↓

PRODUCTION

 - Gameplay Systems

 - Art & Audio Assets

 - Level Design

↓

POST-PRODUCTION

 - Polish

 - Testing

 - Release Build

Pre-Production (Planning Phase)

In this phase, I define the vision: a short emotional 2D game focused on tension and light mechanics. I create a mini GDD describing the core loop and mechanics. Then I build a simple prototype of movement + lantern timer to test if the main idea is fun before making full content.

Production (Building Phase)

Here I create all major systems: player movement, lantern timer, shadow enemies, checkpoints, and UI. I design levels around light management. I produce art assets (character, backgrounds, UI) and basic sound effects/music. Everything becomes playable from start to finish.

Post-Production (Polish & Release)

I improve visuals, adjust difficulty, fix bugs, and test with players. I balance lantern duration and enemy speed. Then I prepare the final build, create a simple trailer, and publish it online.

Team Roles Mapping

Roles & Responsibilities

Game Designer
Defines the vision, mechanics, level layout, and player experience.

Programmer
Implements movement, lantern system, enemy AI, UI, and game logic.

Artist
Creates character sprites, environment art, UI visuals, and animations.

Sound Designer
Creates ambient sound, footsteps, lantern sound, and background music.

QA Tester
Tests for bugs, balance issues, and clarity of mechanics.

Producer
Manages timeline, task distribution, and communication.

Team Interaction Map

Designer → Programmer → QA

   ↓   ↓

 Artist   Sound

   ↘ ↙

   Producer

Designer works closely with Programmer and Artist. QA gives feedback to Programmer. Producer oversees all.

Rapid Prototyping

Chosen Mechanic: Lantern light slowly decreases over time.

I created a simple Unity prototype:

- Character can move and jump

- A timer decreases constantly

- Player loses if light reaches zero

Feedback from peer:
"The tension feels good, but maybe the lantern should recharge at checkpoints."

Reflection (4 sentences):
The mechanic creates natural tension and urgency. However, without recharge points it feels too punishing. Adding safe zones improves pacing. Next iteration should test different timer speeds.

Mini GDD

Title: Last Light
Genre: 2D Puzzle Platformer

Core Gameplay Loop:
Explore → Avoid shadows → Manage lantern → Reach checkpoint → Progress

Target Audience:
Players who enjoy atmospheric indie games (teens & young adults).

Key Mechanics:

- Light as survival resource

- Platforming

- Simple enemy avoidance

- Checkpoint recharge system

Art Style:
Dark minimalistic world with strong contrast lighting (inspired by Limbo-like silhouette style).

Post-Mortem Reflection (Imagined)

What went well:
The core mechanic was clear and easy to prototype. Small scope kept development manageable.

Challenges:
Balancing difficulty and keeping levels interesting without adding too many mechanics.

Pipeline improvement:
Earlier playtesting would improve pacing and reduce redesign later.

Team communication impact:
Clear communication between designer and programmer prevents mechanic confusion and rework.