

Game Design Document

Phase Shifting Platformer

Developer: Kendrick Krause

Engine: Unity (C#)

Platform: PC

Genre: 2D Puzzle-Platformer

1.0 Core Concept

1.1 Elevator Pitch:

A minimalist 2D platformer where the player alternates between two parallel realities to overcome obstacles. Platforms, hazards, and paths exist only in one “phase” at a time, requiring timing, awareness, and rhythm to progress.

1.2 Design Goal:

Create a short, focused prototype that teaches players the phase-shift mechanic through level design — no text, no tutorials — just layered challenges that build mastery through play.

2.0 Design Pillars

Pillar	Description
<u>Simplicity</u>	Minimal mechanics, clear visuals, intuitive input.
<u>Progressive Learning</u>	Each level introduces or layers one new challenge.
<u>Player Feedback</u>	Smooth transitions, distinct audio/visual cues for shifting.
<u>Purposeful Challenge</u>	Obstacles arranged to test timing and awareness, not reflex spam.

3. Core Mechanics

Mechanic	Description	Teaching Moment
<u>Move / Jump</u>	Standard 2D platformer controls	Introduced immediately — basic navigation.
<u>Phase Shift</u>	Toggle between two different ‘worlds’ Each has different platforms & hazards.	The first safe area requires shifting to reach a visible path.

<u>Momentum Retention</u>	Movement continues seamlessly when shifting.	Used in advanced jumps — teaches players to think across worlds.
----------------------------------	--	--

4.0 Level Progression (3-Level Structure)

4.1 Level 1 – Discovery

Simple platforms and one gap that requires a shift mid-air. Introduces visual distinction between the two worlds (e.g., color palette swap).

4.2 Level 2 – Combination

Platforms exist in opposite phases; players must plan switches mid-jump. Adds timing elements (e.g., moving hazards only visible in one world).

4.3 Level 3 – Mastery

Tight timing challenge combining movement, phase shifting, and momentum. “Aha” moment when the player realizes they can chain shifts to bypass an intended obstacle.

5.0 Player Feedback & Feel

Element	Implementation
<u>Visuals</u>	Screen tint or vignette color changes between worlds.
<u>Audio</u>	Subtle tone shift or sound cue when phasing.
<u>Camera</u>	Smooth follow using Cinemachine; small screen shake for impact actions.
<u>Particles / FX</u>	Dust or shimmer effect when shifting.
<u>UI Feedback</u>	Small icon or overlay showing current world phase.

6.0 Technical Notes

Built with: Unity 2023.3.17f1, C#

Core Systems:

PlayerController (movement + jump)

PhaseManager (handles environment toggling + event broadcast)

LevelManager (handles respawn / transitions)

Optional: ObjectPooling for hazards

Design Focus: Clean, modular code architecture with clear separation between gameplay systems.

Stretch Goals: Moving platforms, timed switches, collectibles that exist in only one phase.

7.0 Reflection / Intent (Halfbrick Studios application)

After feedback from my previous application, I wanted to focus on a single, strong mechanic taught entirely through play.

This project emphasizes:

- Iterative learning through design,
- Polished feedback and clarity
- Code readability and modularity for quick feature extension.