Vibrant Postmortem Report

1.0 Project Overview

1.1 Game Details

- **Genre:** 2D Platformer
- **Platform:** PC
- **Project Duration:** 2.5 Months
- **Game Engine:** GameMaker Studio 2
- **Art Software:** Paint.NET

Vibrant is a skill-based 2D platformer designed around minimal controls, abstract visuals, and high mechanical clarity. The project was developed as a self-contained, releasable game within a strict time frame, prioritising scope control and polish over feature breadth.

1.2 Controls

The game is intentionally designed to be playable using only **three inputs**:

- **Horizontal Movement:** Left / Right
- **Jump:** Single action with access to two jumps

Players can use:

- **Keyboard:** WASD or Arrow Keys for movement, Spacebar to jump
- **Controller:** Supported for movement and jumping

This simplified control scheme was chosen to ensure accessibility for non-gamers while still enabling depth for experienced players.

1.3 Project Goals

The primary goals for the project were:

1. To complete and ship a **fully playable, releasable game**
2. To actively control my tendency toward **over-scoping**
3. To design a **skill-based platformer** focused on player mastery
4. To create a game that **non-gamers could understand and enjoy**, without sacrificing challenge

These goals guided all major design, technical, and artistic decisions throughout the development process.

2.0 Art Style

2.1 Level Elements

The visual language of *Vibrant* relies on **basic geometric shapes** and **solid, high-contrast colours**. A deliberately abstract aesthetic was chosen to:

- Reduce visual noise
- Improve gameplay readability
- Emphasise mechanical difficulty over visual realism

A colourful theme was adopted to communicate increasing difficulty across levels while maintaining a consistent and energetic visual identity.

2.2 Menu Backgrounds

Menu screens feature **abstract art compositions** that align with the game's overall colour-forward aesthetic. These backgrounds were designed to complement the gameplay visuals rather than distract from them, reinforcing the game's tone before and between play sessions.

3.0 Gameplay

3.1 Player Movement

The player's movement set is intentionally limited to ensure clarity and mastery:

- **Sideways Movement:** Horizontal movement in either direction
- **Jumping:** The player always has access to two jumps
- **Double Jumping:** The second jump can be performed mid-air, even while falling

Jumps reset whenever the player lands on a platform. This system allows for precise air control, mid-air corrections, and intentional risk-taking during traversal.

3.2 Gameplay Ingredients

Across the game's **25 levels**, new mechanics (referred to as *ingredients*) are introduced gradually. These ingredients appear in the following order:

- **Lava:** Instantly kills the player on contact
- **Sticky Walls:** Prevent falling while in contact, but do not reset jumps, forcing strategic planning
- **Blobs:** Hazardous entities that grow and shrink over time
- **Refills:** Restore the player's jumps on contact; respawn after 1 second
- **Elevators:** Apply an upward force to the player
- **Gravity Pits:** Pull the player downward

- **Portals:** Teleport the player from an entry point to an exit point

Each ingredient was designed to interact meaningfully with the player's limited movement set, increasing complexity without adding new controls.

3.3 Particles and Sound Design

Given the high frequency of player death—often multiple times per level—death feedback was treated as a critical design element.

- **Death particles and sounds** are short, exaggerated, and visually striking
- The intention was to reduce frustration by making failure feel immediate and visually satisfying

This approach helped maintain momentum and encouraged experimentation rather than punishing mistakes.

4.0 Level Design

4.1 Core Design Criteria

All levels were designed around the following principles:

Short and Fast Levels

As a skill-based platformer, *Vibrant* features increasingly difficult challenges. To prevent frustration from repeated failure, levels were kept deliberately short so players could retry quickly without losing significant progress.

Learning Through Failure

Levels were designed so that failed attempts clearly communicate *why* the player failed, encouraging experimentation and improvement rather than trial-and-error guessing.

Speedrun-Friendly Structure

All levels are deterministic, with no reliance on randomness. This ensures consistency across attempts and allows the entire game to be speedrun. Some levels include optional, more difficult routes that reward mastery with faster completion times.

4.2 Teaching the Player

The game uses a combination of **explicit and implicit tutorials**:

- **Written Tutorials:** On-screen text explaining mechanics directly
- **Hidden Tutorials:** Level layouts that naturally teach mechanics through player interaction

Example: Level 1

In the first level, the player is introduced to jumping mechanics. The initial obstacle can be cleared with a single jump, while the next requires a double jump. This teaches the concept without explicit instruction.

However, an alternative solution—running from the first platform onto the second—was intentionally left possible. This decision accommodates players with little platforming experience and gently introduces basic movement principles. A later level explicitly focuses on teaching the double jump mechanic.

4.3 Pacing

To support fast iteration and learning, a strict pacing rule was applied:

- **No level exceeds 20 seconds** in a flawless run along the intended route

This constraint allowed for high-difficulty jumps without overburdening the player and ensured frequent moments of success even during challenging sections.

4.4 Speedrun Routes

Several levels include **optional alternate routes** designed for advanced players and speedrunners.

Example: Level 15 – *"My Escape"*

In this level, the most obvious solution is to exit the enclosure from the top and move horizontally. However, a hidden opening at the bottom allows skilled players to complete the level in under 5 seconds.

This route is accessible to all players but requires significantly higher precision, reinforcing the game's philosophy of optional mastery rather than forced difficulty.

5.0 Conclusion

Vibrant successfully met its original goals by delivering a complete, polished, and mechanically focused platformer within a constrained scope. The project reinforced the importance of clear mechanics, strong pacing, and player-friendly failure in skill-based game design.

The postmortem highlights how deliberate limitations—in controls, art style, and scope—can lead to a more cohesive and refined gameplay experience.