# 2D LEVEL DESIGN

## Overview

Title: Crumbling Skyways

Engine: Godot

Time to design and develop: 2 weeks.

Genre: 2D Puzzle Platformer

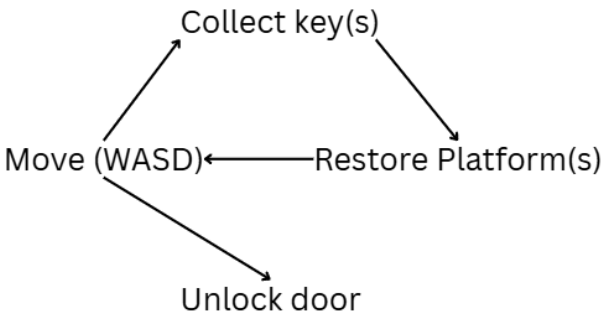
Controls: A/D -> Movement, W -> Jump

One line description: A minimalist puzzle-platformer where every movement is irreversible - unless you earn the power to undo it.

Core idea: Players must strategically collect and spend keys to progress: keys are needed both to unlock the exit and to restore the collapsing platforms they walk on, planning each step carefully.

Description:  Crumbling Skyways is a 2D puzzle-platformer set in a crumbling castle. You play as a knight ascending floating platforms toward the castle doors, where each platform can collapse after jumping The core challenge lies in collecting keys and using them wisely to restore lost paths. The levels become progressively complex, pushing the player to plan movement, manage risks, and decipher platform patterns in both horizontal and vertical space.

## Core loop




**Simple goal, deep strategy**: Just get the key and reach the door—but the way is not so simple.



**Restore mechanic**: Breaking platforms forces players to make conscious, permanent choices.



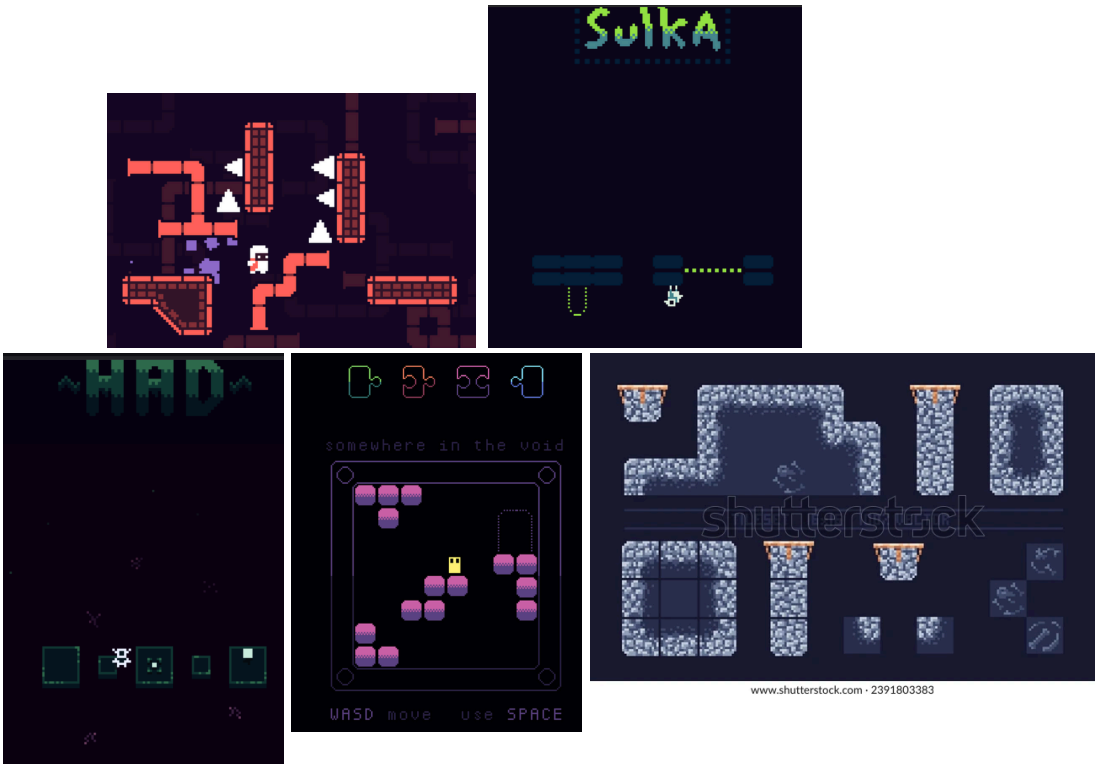



**Puzzle-driven progression**: Levels teach movement and planning through clever placement and misdirection.

# The Process

Took inspiration from various game design ideas to learn about level design in a base level.
**Kultisti, an itch game developer with 49 published games** was the primary inspiration.
🌐 Kultisti
Studied the following games:
Sulka and Had: 🌐 Sulka 🌐 Had

Inferences:
The games had simple mechanic, cleverly thought out level design and simple art style.
My goal was to recreate that.
References:




Idea formation:

**Design goal:**

- **To give the player feeling of “Missed by an inch”.**
- **To make each move count and worthy.**

Having a knack for action and puzzle decided to come up with a game which tests both: **Motor skills and Logical thinking.**

So based on this I had an idea where the classic unlock the door mechanic with a key. So the challenge was where to add the puzzle.

**Actual jump distance VS Perceived jump distance.**
So to give the feeling of missed by an inch the player had to fall from platforms and feel like it was a mistake, this can be done by placing the platforms just little away from player's jump distance.

So from here the idea of **platforms as game resource** came into being.

Then the game had to have a platform where it could force a player to think before walking on it. I knew about the platforms which were temporary.

**Three kinds of platforms**:

1) Weight sensitive Platforms: Platforms that stand only for limited time duration, and you have to cross within that time frame else you'll fall along with it.

2) One time use Platforms: Platforms which fall after you walked over them once.

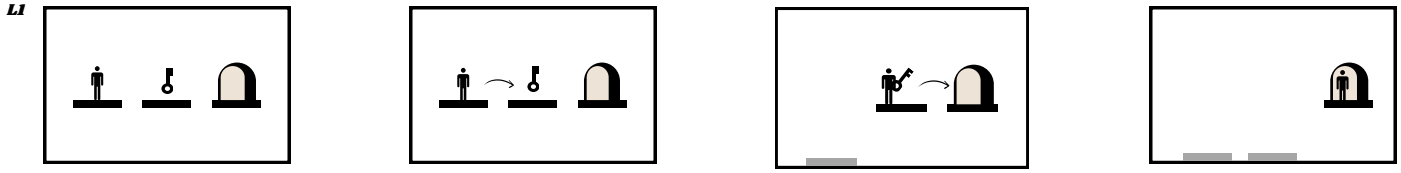
3) Timed restore: Platforms which restore themselves after a while of falling.

I decided to use a combo of 2) and 3). Where I can give the choice of restoring the platform to the player.

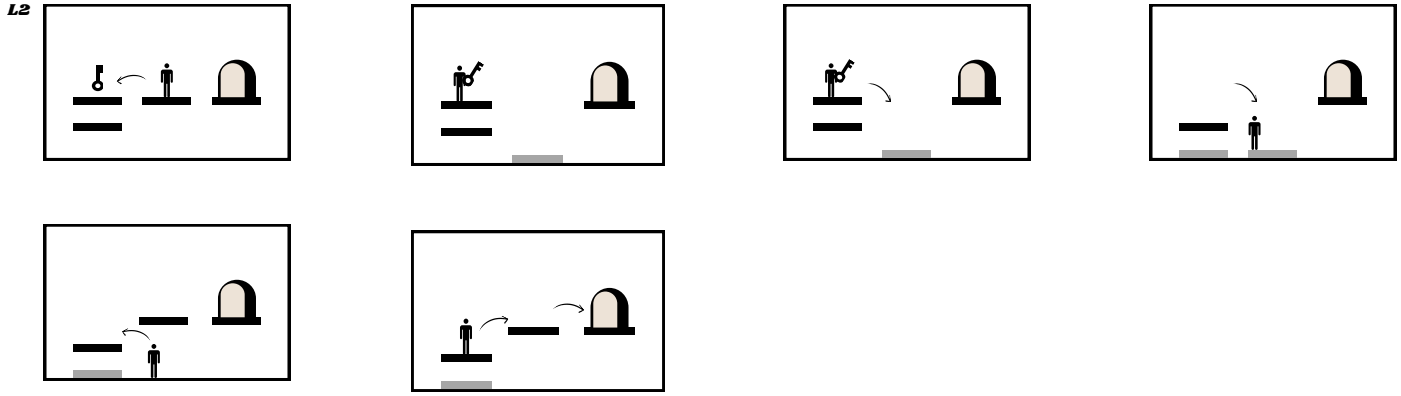
This made up the puzzle part: **Deciding between which platform to restore and when to restore it.**

The initial paper prototype:

**Level 1:**




**Level 2:**




Decided to dig deeper for more level design ideas:

**Mechanic table:**

Read as: From left to right with respect to time.

| Mechanics | Jump | Key | Movement | Platform | Enemy |
|---|---|---|---|---|---|
| Jump | Double jump? | | | Breaks if health is 1. | Timed to attack |
| Key | Breaks? | Can be in pieces | Can move if platform breaks | | |
| Movement | Diagonal Jump | | Timing | Rotate + Glide | Timed to attack |
| Platform | Breaks! | | | Can be stacked | |
| Enemy | Kill enemy | Cant kill without the key? | | Can be crushed under platform | Special enemies give +1 platform |
| | | | | | |

The above table allows me to explore different possible use cases of a same game element, ponder its interaction with other game elements and figure out what combination of interaction should be allowed and not allowed.

# Design strategies

Level progression table:

🌐 Level Design 2d progression chart

Table1

| # | Level number | Type of level | Mechanic introduced | Red herring | # | Avg time taken to finish in secs | Avg number of retries |
|---|---|---|---|---|---|---|---|
| | 1 | Introductory | Teaching movement Left to right | | | 8 | 0 |
| | 2 | Introductory | Teaching Jump | | | 6 | 0 |
| | 3 | Introductory | Falling platforms | | | 6 | 0 |
| | 4 | Testing | collect the required number of keys by risking life | | | 7 | 0 |
| | 5 | Normal | Falling platforms + jump | | | 5 | 0 |
| | 6 | Introductory | Platform Restore mechanic | | | 7 | 1 |
| | 7 | Testing | | Extra platform | | 7 | 0 |
| | 8 | Introductory | Jumping down strategy | | | 8 | 1 |
| | 9 | Normal | Bonus Level | | | 10 | 2 |
| | 10 | Normal | Teaching actual Jump distance | | | 9 | 0 |
| | 11 | Testing | | Extra platform, extra key | | 10 | 2 |
| | 12 | Introductory | Green platforms for high jump | Extra platform | | 7 | 0 |
| | 13 | Testing | | Positioning of platforms | | 8 | 2 |
| | 14 | Normal | | Positioning of platforms | | 11 | 1 |
| | 15 | Normal | | Percieved jump distance | | 10 | 1 |
| | 16 | Normal | | Positioning of platforms, percieved jump distance, extra platform | | 17 | 4 |
| | 17 | Normal | | Positioning of platforms, extra Platforms | | 13 | 3 |

## Level design:

**1) Level Division**: The levels were divided into series of sets of 3-4 levels which taught one particular gameplay mechanic to the player. It was divided as Introduction to mechanic, then testing if the player has learnt the mechanic by presenting it in a form of a puzzle. This ensured the teaching to learning curve was gradual and smooth. Also this negated the need for tutorial or help level to teach the player. And after every set there was a bonus level to take off the pressure; ensuring players were not overwhelmed and stayed in flow state.

Levels 1-5: Teach movement, jump, and introduces falling platforms.

Levels 6-8: Introduce the restore platform mechanic and tests it.

Level 9: Fun level.

Level 10-17: Combination of different puzzles and testing the previously taught mechanics.

**2) Red herrings**: Game resources like key and platforms are used as red herrings to distract the player from reaching the door in first try.

- Platforms: Platforms are against a blank background and hence are lit pathways to the door. According to player psychology they are resource tweenies showing the path. Hence giving a hint to the player how to reach the door. Clever placement of platforms was used as red herring confuse the player.

- It was precisely placed after playtesting. Based on perceived jump height and actual jump height, placement of platforms made it a good mechanic to lead the player towards or away from desired solution.
- Placement of red and green platforms indicated possibility of high and low jump, subtly indicating/ hinting at what path to take.

- Key: A resource that has two uses: Unlocking the door and restoring the platform. This creates a conflict between the players to use them wisely. Player psychology will tend to amass as many keys as possible per level to feel safe and free. So clever placement of keys was used are red herring to trap the player in compromised position

Final 17 levels:

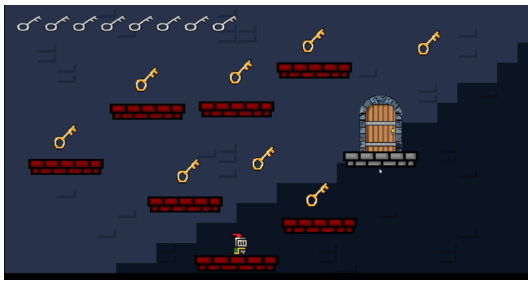

Collect the key and unlock the door!

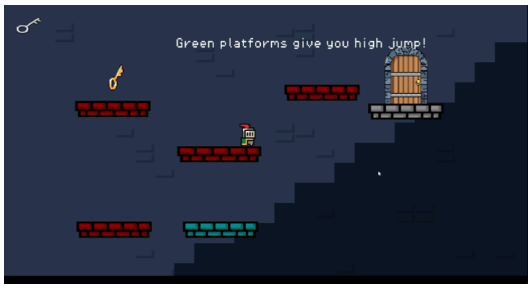

Don't Fall Down!


Careful!

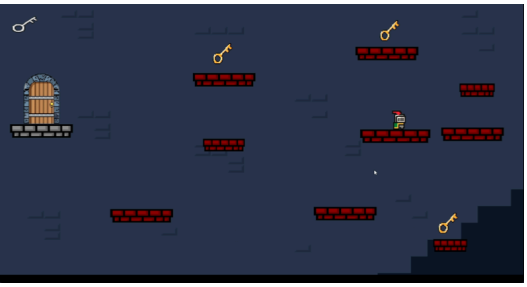

Collect the Required number of keys!

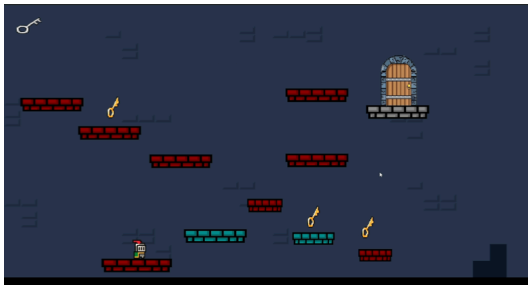

Press R to restore a fallen platform.
Restoring a platform requires 1 key.




Green platforms give you high jump!


Restore a red platform to make it green!





**Figuring out movement**:

- Horizontal: Initial levels I promoted left to right moment just as classic platformer. Then I show them right to left. Then I introduce the puzzle. The problem That once you go right you can't go left without using key. Hence the restore mechanic is introduced. Hence the left to right carefully planned movement is taught. Here jump is also taught.
- Vertical movement: Players can go down from anywhere but can't jump as high. So that was also an opportunity which I exploited to make people think to achieve their goals. People learnt to think in downwards direction how they can move and use the height/elevation to their advantage. Then vertical high jump is introduced to make them think in upper direction as well. This made them think how they can reach in upper direction and planned accordingly.

**Actual jump height and perceived jump height:**

- Perceived jump height: Player visually thinking it is possible to make this jump, but realizing he cant after few tries. Hence switches to a better strategy rather than brute forcing his way out of levels. -
- Actual jump height: Players precisely making a good jump to get to places. Its a reward in itself. Using the perceived jump as red herring I could trick the player to take the wrong path. To take the correct path is through trail and error. This is one of the ways I used to design the levels.

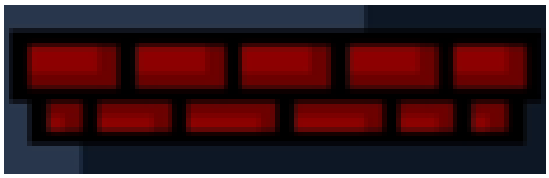
**High jump:**

- An important mechanic that made its way later is the high jump. I added it to platforms as health to give high jump if its green. Orchestrating where player can use the high jump platform and where he can plan the jump down from there after.

Color coded platforms:

Grey platform: All reliable and classic platform (Can be used many times)

Red platform: Its gonna break after you walk off it. (Health = 1, Can be used only once)



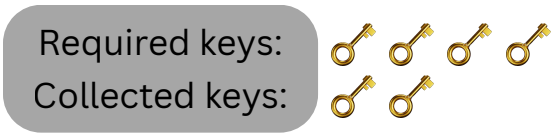
Teal platform: Your jump height will be double after jumping off this once. (Health = 2, can be used twice.)

Once you jump off the teal platform, it turns red, and on second usage, it collapses.

**HUD**:

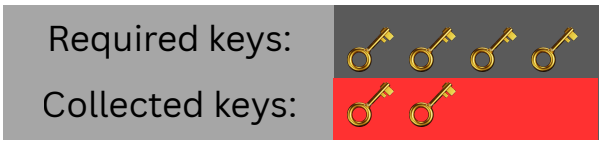
1) Low-fi Prototype Keys will be depicted as numbers


2) 1st Iteration Keys will be symbols which will appear on screen as we collect more keys

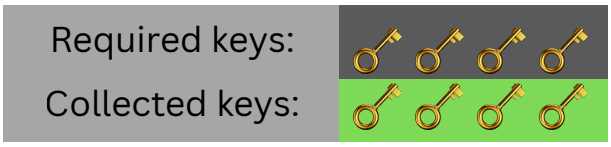



3) 2nd Iteration

Before collecting




After collecting




4) 3rd Iteration: Finally decided to depict required keys in grey and collected keys in gold. This removed the requirement of textual labels.

*A visual cue instead of words*

**Technical improvements:**

**Movement Speed** adjustment to allow smooth experience.

**Coyote Time**: Allows a short delay to jump after leaving platform.

**Jump Buffering**: Allows queued jump before landing.

# Graphs:

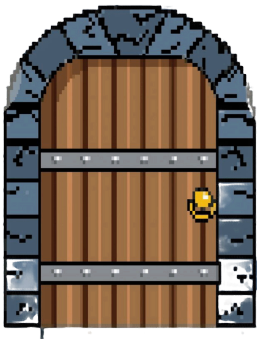
## Complexity graph: (Used actual playtested values)

Ensured a gradual complexity curve over the levels.

**Art style:**

Tried to keep the art style simple and intuitive. No extra elements. All done on Procreate.

Inspired by the free knight character asset, decided to give a Stone castle like vibe to the overall game.

Golden key (collected key)

Grey Key (Required key)

Double Bricks

Single bricks

Animated key spritesheet

Animated platform spritesheet

**THANK YOU**