Portfolio for MSc in GAMES, Design Track

5 descriptions of game development work done alongside studies, by Oskar Ingmar Larsen

Contents

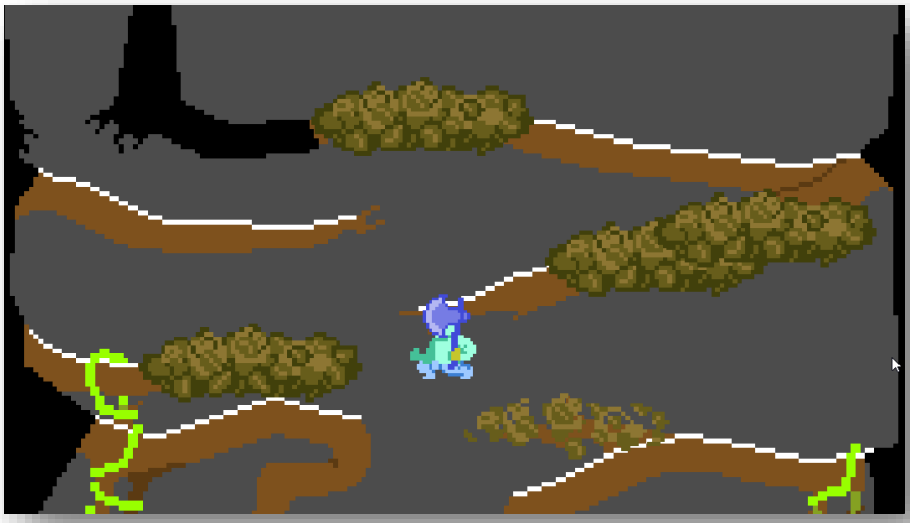
Godot project development “Vania”

This section offers an in-depth look at the progress of my latest project, currently titled "Vania" as a working title. My development setup includes a design document in Google Docs, a wall-mounted whiteboard for sketching, Godot as the game engine, and GitHub for version control and backup.

Description #1: Concept development and design

This project is primarily inspired by Konami’s “Castlevania” (1986)[1], Nexiles’ “Jump King”[2], and Nintendo R&D4’s “Zelda II: The Adventure of Link”. These inspirations have informed the design of a 2D action platformer with precise platforming, combat, and “Foddian” punishment as derived from my Jump King inspiration.

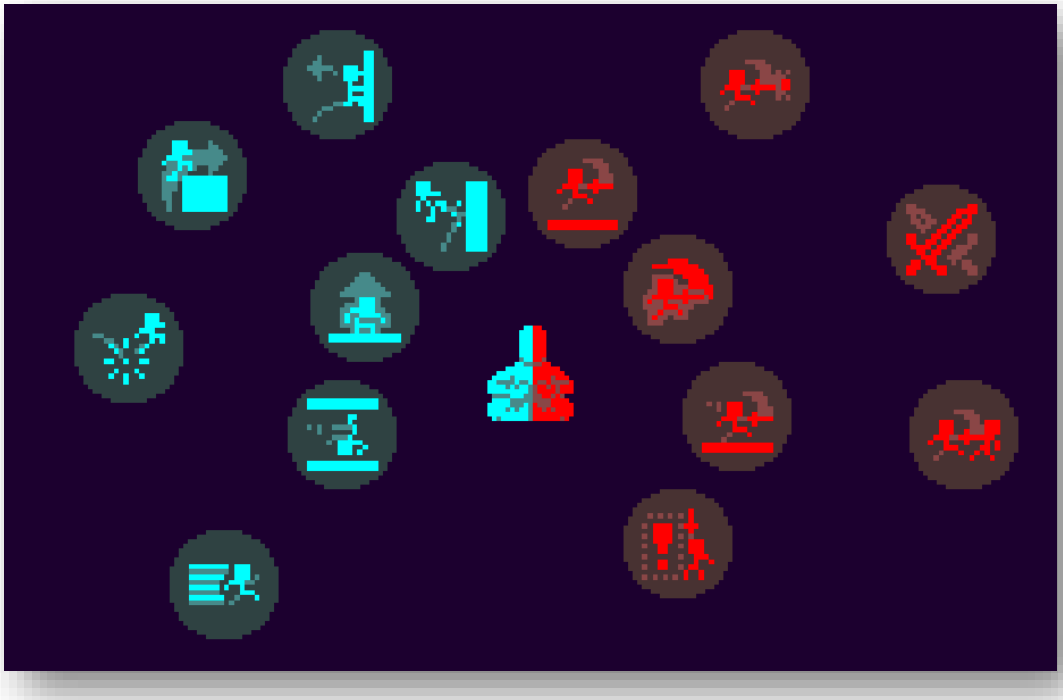
The definition of “Foddian” is arguable, so I’ll clarify it to be as a way to describe how player punishment in either combat or platforming gameplay sets the player back in their vertical level progression. I’ve implemented a knockback feature that potentially knocks players down through “semisolid”[3] platforms of the game’s environment to achieve this kind of player punishment.



Screenshot of my “Vania” Godot project: the player character jumps from a dissolving leaf platform to another platform, as signified by it’s white outline. The pixel art seen in these screenshots are NOT final.

[1] Wikipedia page: https://en.wikipedia.org/wiki/Castlevania_(1986_video_game)
[2] Steam page: https://store.steampowered.com/app/1061090/Jump_King/
[3] A term borrowed from Super Mario Maker: https://www.mariowiki.com/Semisolid_Platform

The player takes on the role of a character destined to ascend a vast vertical level, progressing from the bottom to the top. As they climb and encounter new platforming and combat challenges introduced through level design, they may occasionally be knocked back down. However, by interacting with designated areas to plant a seed in the soil, they create an opportunity for growth—if they fall far enough below the planted seed, a fruit will bloom. Interacting with this fruit grants an experience point, which can be invested into either the combat skill tree (represented by red icons on the right) or the platforming skill tree (represented by blue icons on the left):



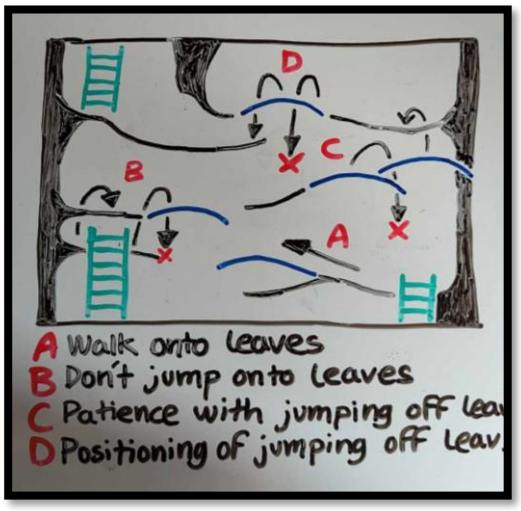
Screenshot of the skill tree from my "Vania" Godot project

If this concept is executed upon as intended, it should encourage players to overcome the combat and platforming challenges, encouraged by the "horizontal progression"[4] afforded to the player via the skill tree.

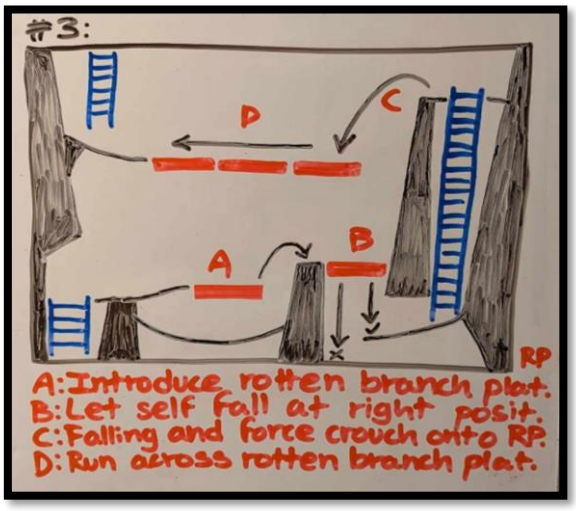
[4] Mmorpg forums discussion: https://forums.mmorpg.com/discussion/502333/vertical-vs-horizontal-progression-how-i-think-you-can-have-both
r/truegaming thread:
https://www.reddit.com/r/truegaming/comments/27hsry/whats_the_difference_between_vertical_and/

To maintain a structured production process, I detailed this concept extensively in a game design document[5]. This document encompasses various pre-production notes, including narrative contextualization within world building, level design progression, and the gradual escalation of challenges as new gameplay mechanics are introduced. The approach to introducing gameplay concepts is heavily inspired by Koichi Hayashida's design philosophy, following the four-step *Kishōtenketsu* structure[6].

This level design approach is first explored through whiteboard sketches, mapping out how different sections of the game's vertical level should be structured to challenge the player. In these sketches, black lines represent semisolid platforms, while solid black areas indicate impassable terrain.



Whiteboard Sketch #1: Beginning with a climb up the ladder in the bottom-right corner, this level section introduces and challenges the player with fragile leaf platforms (depicted as blue arches), which dissolve upon jumping onto or off them.

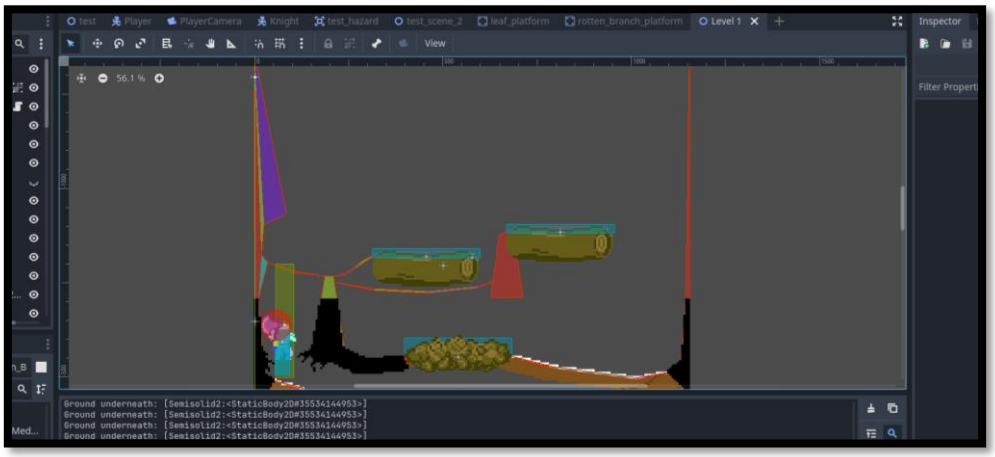


Whiteboard Sketch #2: Starting from the ladder in the bottom-left corner, this section presents the challenge of traversing rotting branches (represented as red horizontal rectangles), which gradually disintegrate while the player stands on them.

After creating an initial level design sketch, I move on to prototyping the level in Godot, constructing semisolid and solid environments using 2D collision polygons. This approach

[5] Dropbox link til midlertidigt design dokument:
https://www.dropbox.com/scl/fi/eabnjmt7vwed5lxq1g9y8/GODOT-VANIA.pdf?rlkey=aymeygneit86z4110bnbkg8di&st=qlqxmx15&dl=0
[6] Game Makers Toolkit:
https://www.youtube.com/watch?v=dBmIkEvEBtA&ab_channel=GameMaker%27sToolkit

offers flexibility, allowing the level to adapt seamlessly to pixel art without being constrained by tilesets:



Screenshot of my "Vania" Godot project, 2D collision polygon shapes visible.

Once the initial semisolid and solid environments are in place, I begin programming the gameplay mechanic introduced in the level. This process involves iterative adjustments to the existing environment, refining player navigation paths through testing and fine-tuning.

Description #2: Feature implementation, testing and art

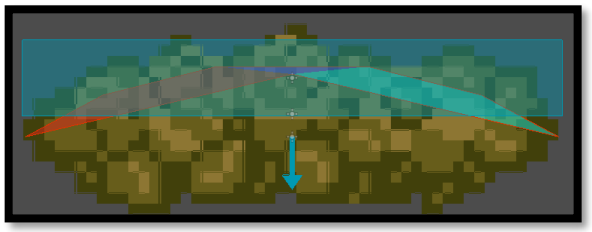
With the core level structure in place, I proceeded to refine interactive elements, focusing on the leaf platforms. This section details their design, testing, and iterative improvements.

Zoom-in of screenshot of my "Vania" Godot project: Stable leaf platform.

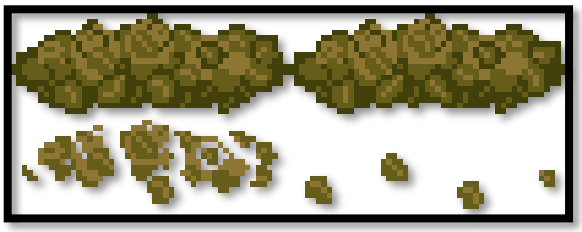
Zoom-in of screenshot of my "Vania" Godot project: Dissolving leaf platform.

Leaf platforms are intended to gradually challenge players' decision-making on when and where to jump or land. This encourages careful navigation while discouraging reckless jumping. To ensure this mechanic functions as intended, I've programmed the player character to emit signals upon jumping and landing, which can be detected by any environmental entities subscribed to that signal. In addition to programming the leaf

platform's behavior, its entity is composed of key components such as a collision shape, sprite graphic, and trigger zone.

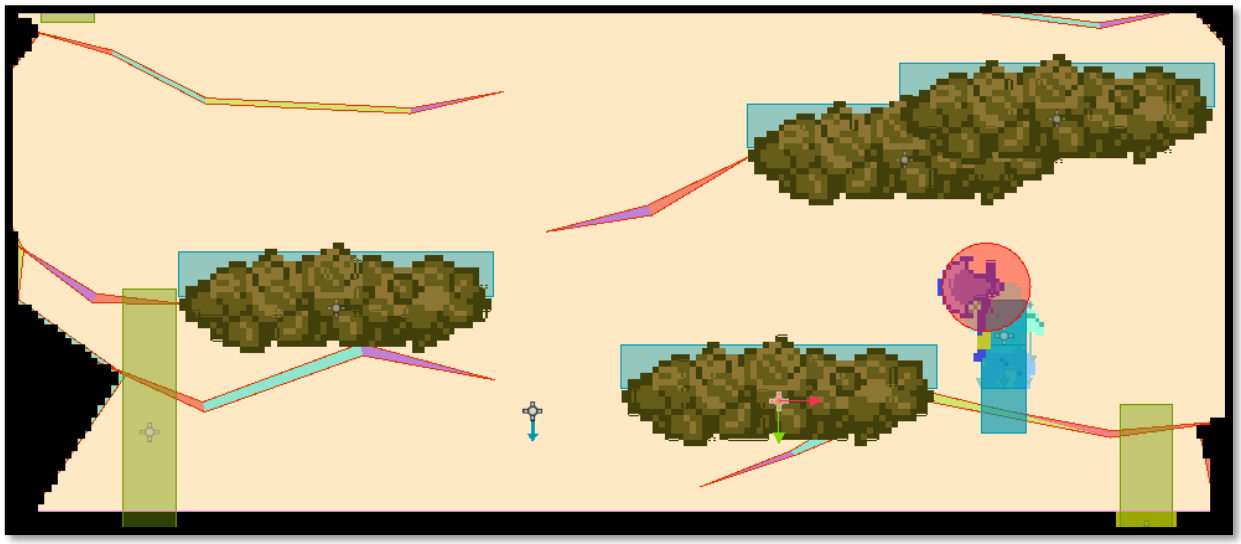


Screenshot from my Vania Godot project: A leaf platform with visible 2D semisolid collision and trigger-zone shapes. The blue arrow indicates the direction of one-way collisions, meaning the platform only collides with and reacts to the player when they are moving downward.



Leaf platform sprites (51x18 px): The top two sprites alternate to create a "wiggly" animation, making the platform visually distinct from the environment. The bottom two sprites play sequentially from left to right when the platform dissolves.

Testing new features, such as leaf platforms, requires thorough playtesting to ensure their stability and proper functionality. Since I am currently working on this project alone, all testing is conducted solely by me at this stage.



Screenshot of my "Vania" Godot project: Leaf platform introduction section with 2D collision polygon shapes and trigger zones visible.

Once a feature has been thoroughly tested, iterating through multiple prototypes until no unintended behaviors or bugs remain, I move on to the next level or feature for development. This involves designing, sketching, implementing, testing, and refining.

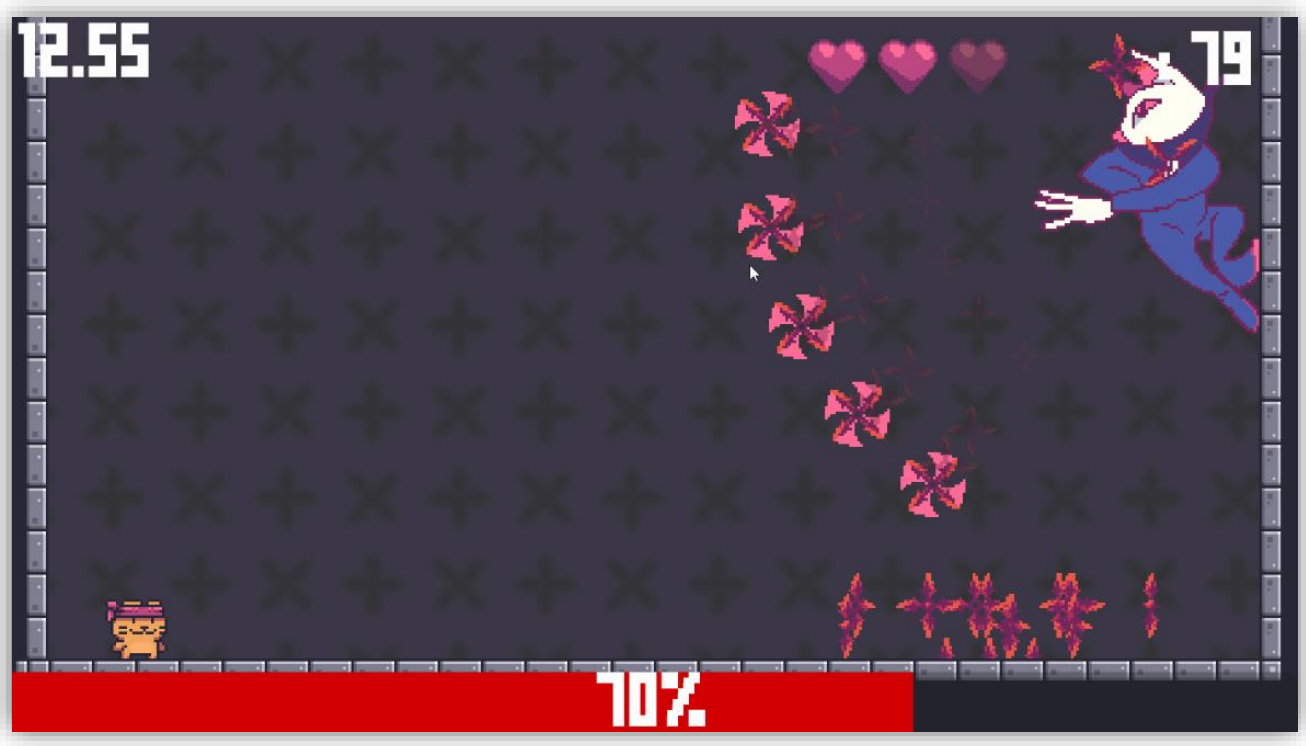
Several key aspects of the core gameplay loop, such as sowing fruit seeds, their blooming into fruits, and the ability to consume them for experience points, have yet to be implemented. According to my development plan, this feature also needs to be in place before the functional implementation of skill tree investment. The delay in implementing these mechanics is due to my priority of developing a presentable level, both to test new features and to gather feedback from play testers selected through convenience sampling.

Description #3: gameDevCorner community project pixel art

“gameDevCorner” (GDC) is a student organization dedicated to game development, fostering meetups, discussions, and collaboration among independent developers. Open to all skill levels, it serves as both a learning space for newcomers and a creative outlet for experienced developers.

During a GDC-organized game jam at ITU, I collaborated with fellow members to design a level for our community game, developed in Godot. We chose to create a boss level, for which I contributed custom graphics for the boss enemy; a white ninja cat wielding ninja stars and performing various ninja moves.

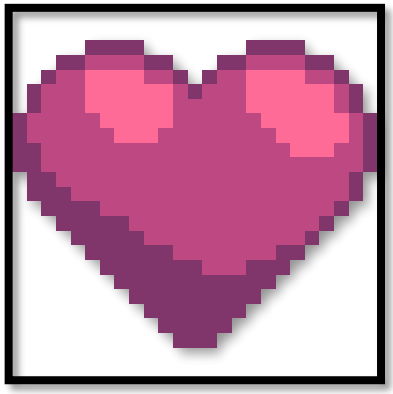
A snippet of gameplay can be seen in the following GIF:
https://media2.giphy.com/media/v1.Y2lkPTc5MGI3NjExa3c0dXhrdTA3ZXVwZmN3aGVtdDVrdjFhY21meWUzOG41MnA0eW5zayZlcD12MV9pbnRlcm5hbF9naWZfYnlfaWQmY3Q9Zw/gTXDJ5ykcwAd85yYhu/giphy.gif



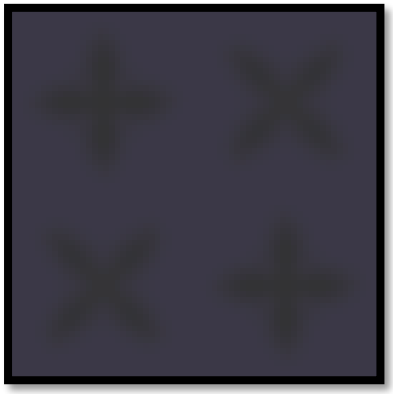
Screenshot of level ”Ninja time” in gameDevCorners’ community project

The artwork was created in GIMP, a free image editor, where I designed pixel art sprites, background patterns, and animation spritesheets for use in the Godot game engine, all saved as .png images:

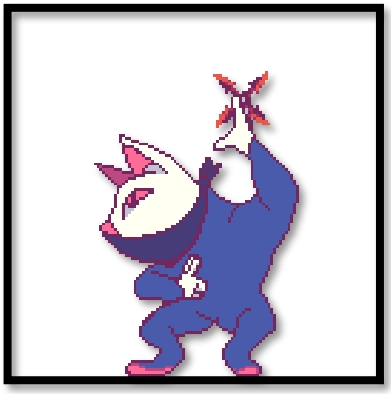
Sprites



25x25 px Player Heart Sprite: The smallest sprite among my contributions to this project, though resized in Godot for better visual clarity. Since these sprites don't appear in the foreground, the issue of mixed pixel resolutions (mixels) is less disruptive.



64x64 px Background Tile Sprite: Intentionally blurred to avoid drawing attention away from foreground elements. Designed to seamlessly tile when repeated in the background.

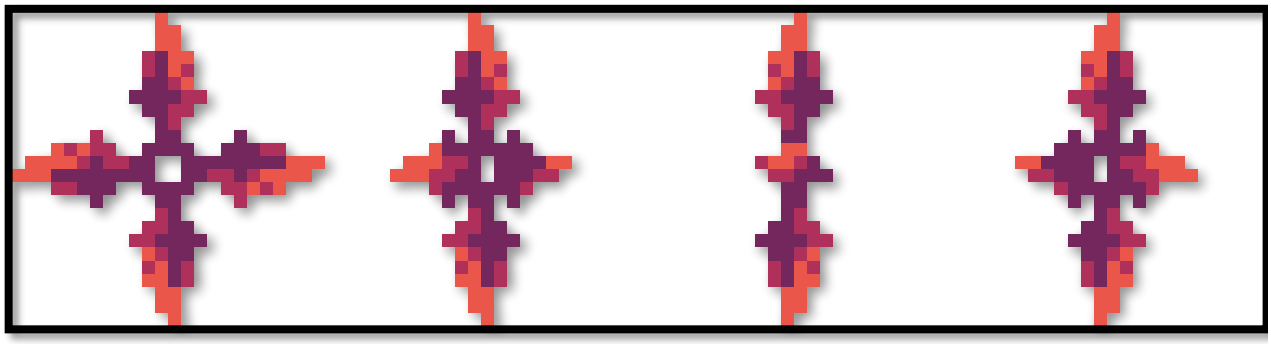


128x128 px Ninja Cat Idle Sprite: A large sprite, as the boss is meant to be bigger than the player. Resizing it in Godot instead would cause mixels, a mix of pixel resolutions, which would become visually distracting as opposed to the player heart sprites.

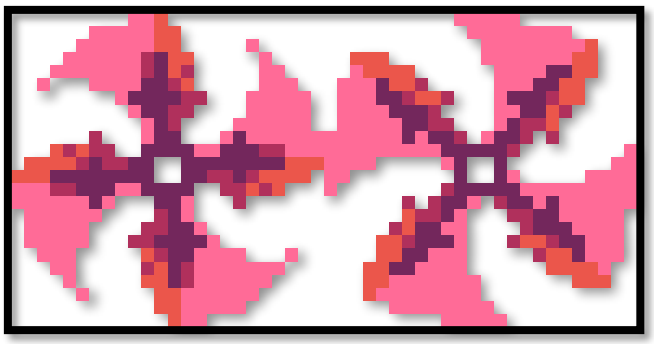
Both character design and animations for the Ninja Cat was drawn by me in GIMP, implemented effectively into the game by collaborating with my fellow GDC members to ensure seamless integration into Godot and gameplay.

Spritesheets

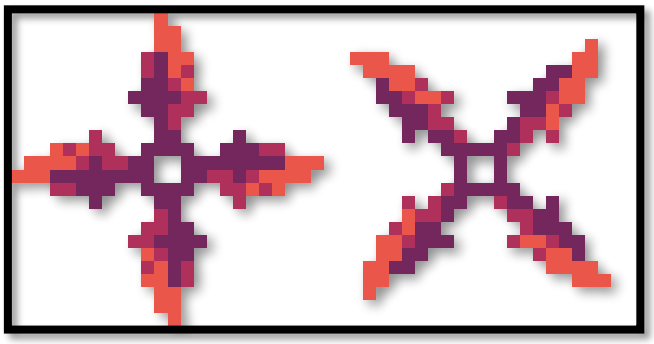
Throwing star projectiles



Idle Ninja Star Sprites (4 x 24x24 px): Animated to appear as if they are spinning horizontally on their vertical blades, signaling that they are non-damaging to the player unless thrown and may be picked up like coins, borrowing from their animated affordance conventions.



Boss-Thrown Ninja Star Sprites (2 x 24x24 px): Feature pink "smear" trailing effects to indicate motion and danger. These stars will damage the player when thrown but become idle upon hitting the environment.

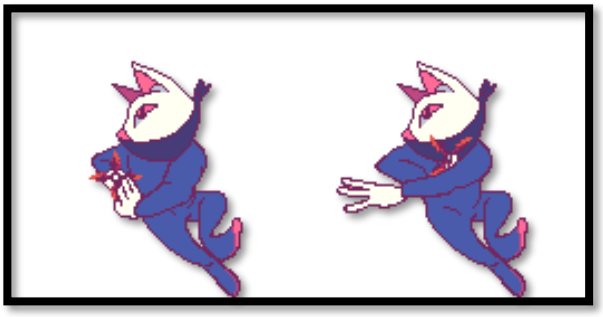


Player-Thrown Ninja Star Sprites (2 x 24x24 px): A variant of the thrown ninja stars, lacking the pink "smear" trails to distinguish them as non-harmful to the player, like their idle state.

Ninja cat attacks

Grounded Claw Charge Sprites (2 x 128x128 px): Used for animating the ninja cat boss's charge attack while on the ground. Includes a "wind-up" sprite to telegraph the move before the "release" sprite executes the attack.

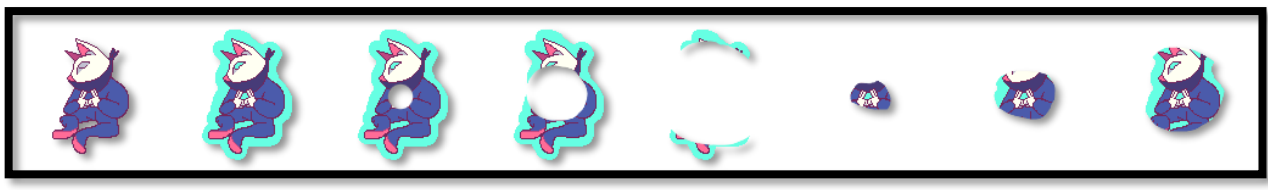
Mid-Air Claw Charge Sprites (2 x 128x128 px): Similar to the grounded variant but executed when the ninja cat boss charges downward from the air. Both versions use a smear effect to visually signal danger.



Grounded Ninja Star Throw Sprites (2 x 128x128 px): Animate the ninja cat boss's throwing attack while on the ground, with a "wind-up" sprite indicating the attack before the "release" sprite.

Wall-Mounted Ninja Star Throw Sprites (2 x 128x128 px): A variation of the ninja star throw attack, performed when the ninja cat boss is mounted on a wall and throwing downward.

Unlike the claw charge, ninja star throw animations lacks a smear effect, as the threat comes from the thrown stars, not the boss itself.



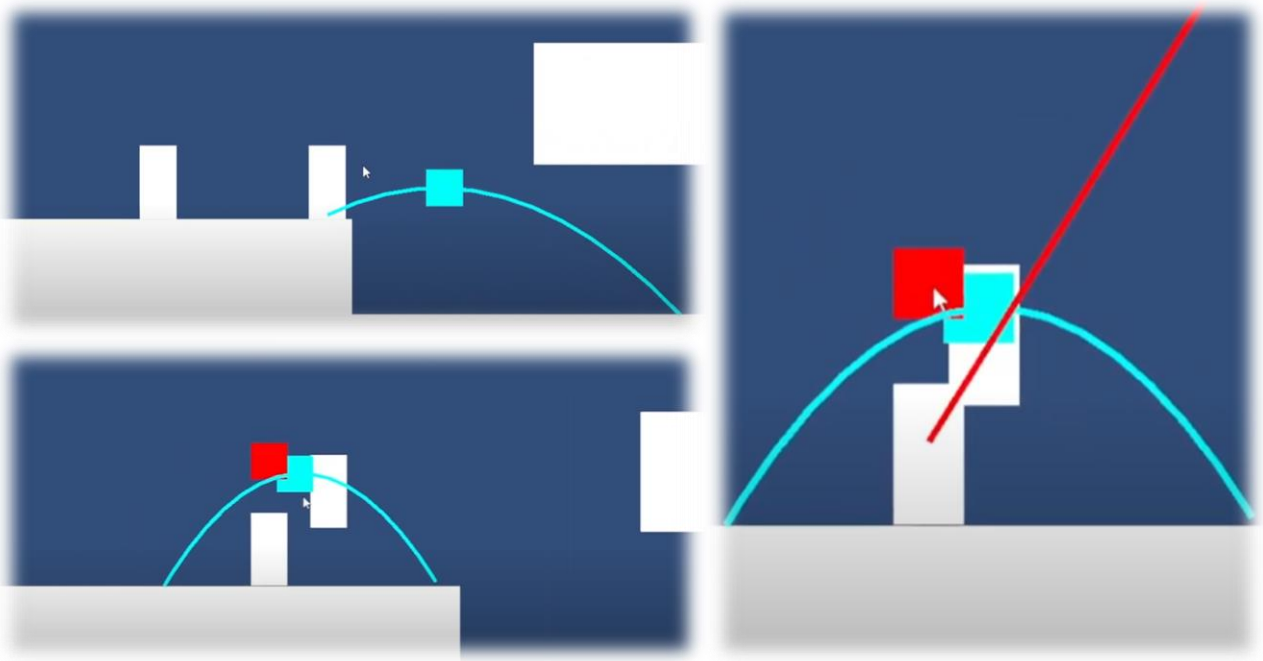
Teleportation Sprites (8 x 128x128 px): Used to animate the ninja cat boss's movement within the boss arena, replacing running or jumping animations due to time constraints. A blue outline highlights the boss's vulnerability during this phase, signaling to the player that it's a key opportunity to attack.

Unity projects

A pair of older projects from my time working in Unity are also relevant in demonstrating the range of gameplay concepts I've explored, though they are currently on hold. A video showcasing these projects can be viewed here, with project chapters in description: https://www.youtube.com/watch?v=nLFElRrilLc&ab_channel=Albidalbi

Focusing on the first two projects featured in the video, I'll refer to them respectively as "Point-and-Click Platformer" and "Torch FPS". The visuals in these projects are quite basic, but the provided screenshots will be accompanied by explanations for clarity:

Description #4: Point-and-click platformer



Collage #1: Screenshots of Point-and-click platformer. The player (represented as a white vertical rectangle at the center of the screen) jumps along a blue jump arc, positioning themselves close enough to an enemy to trigger a hovering kill-button. A red line extending from the enemy's center through the player indicates the enemy's line of attack.

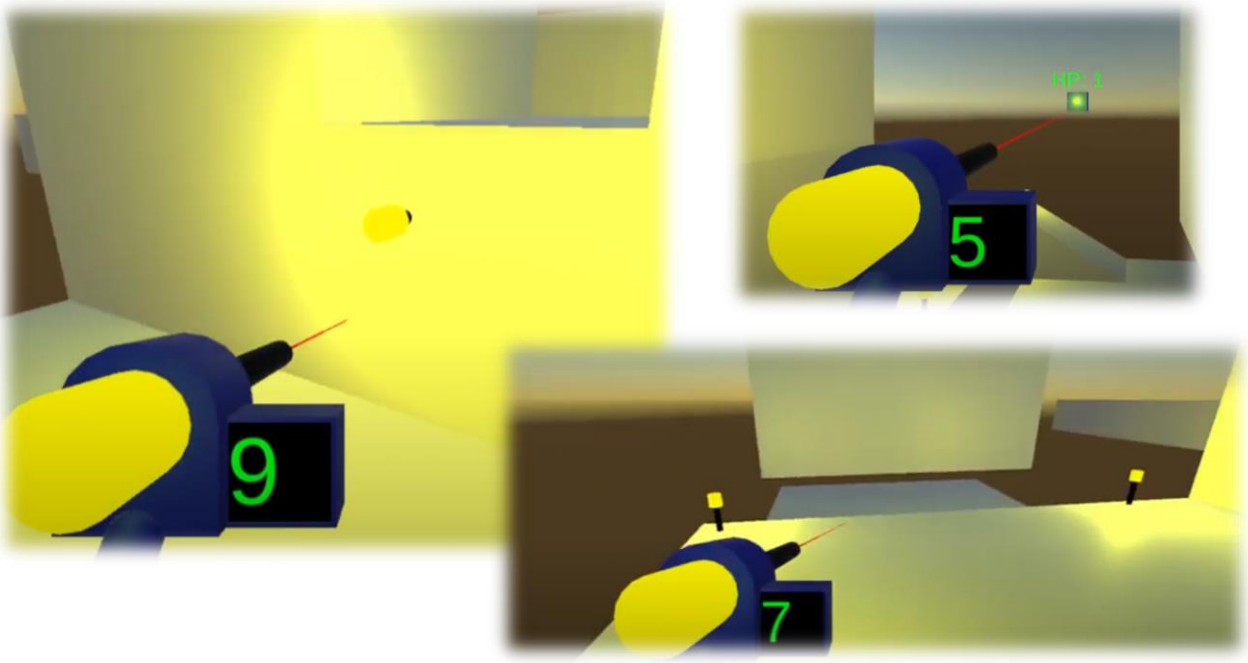
Heavily inspired by Tomasz Wacławek's Ronin, this project was designed as a turn-based 2D action puzzle platformer. During the player's turn, they controlled a character who moved according to a jump arc that followed the mouse. Pressing the left mouse button would make the character jump along this arc. At the peak of the jump, the game would pause (indicated by the blue squares at the arc's peak in Collage #1), allowing the player to make strategic

decisions such as attacking, allowing the character to fall along the arc, or performing other actions that could be implemented in future iterations.

When gameplay resumed, enemies would execute their actions, which were telegraphed during the player's turn through attack lines. This system encouraged strategic planning, as players needed to avoid ending their turn in an enemy's line of attack, reinforcing the puzzle-like aspect of the action platformer.

Further development would have focused on creating a vertical slice with the game's minimum viable features, followed by playtesting to refine gameplay through iterative design.

Description #5: Torch FPS



Collage #2: The player wields a torch-shooting gun equipped with a laser pointer, indicating where the torch will land and stick.

Inspired by the Immersive Sim genre, particularly "Cruelty Squad" by Consumer Softproducts, and games with diegetic UI, such as "Dead Space 2" by Visceral Games, this project was designed as a first-person immersive sim. The focus was on creating versatile tools that enabled both FPS-style combat and puzzle-platforming mechanics.

The torch gun, as shown in Collage #2, served multiple purposes: it could be used as a weapon to damage enemies, as a light source to illuminate dark areas, and as a platforming tool—allowing the player to create surfaces for traversal, such as navigating through tight spaces.

Further development would have expanded the game's toolset beyond the torch gun (e.g., a wire gun for connecting electrical circuits and rope-swinging) while also introducing primitive enemies to encourage creative use of these tools in both combat and environmental interactions.