Interactive Portfolio Project Notes

Concept

- Pixel art portfolio website for CS degree
- Landing page = interactive room/computer game
- Navbar with GitHub/projects links
- Responsive: desktop=desk view, mobile=zooms to computer screen

Tech Stack

- Godot (HTML5 export)
- GitHub API integration for live contributions graph
- Pixel art style throughout

Godot Project Settings

Display:

- Size: 1920x1080
- Resizable: ON

Stretch:

- Mode: canvas_items
- Aspect: keep
- Scale: 1.0
- Scale Mode: integer (for crisp pixels)

Textures:

- Default Texture Filter: Nearest (pixel perfect)
- Import sprites: Filter OFF, Mipmaps OFF

Scene Structure (Complete)

```
Main (Node2D)
├── Camera2D
├── DeskView (Node2D)
│   ├── Background (Sprite2D)
│   ├── Desk (Sprite2D)
│   ├── Computer (Sprite2D)
│   │   ├── ComputerClickArea (Area2D)
│   │   │   └── ComputerCollision (CollisionShape2D)
│   └── Posters (Node2D)
│       ├── GitHubPoster (Sprite2D)
│       │   ├── GitHubClickArea (Area2D)
│       │   │   └── GitHubCollision (CollisionShape2D)
│       └── SkillsPoster (Sprite2D)
└── ComputerView (Node2D)
    ├── Screen (Sprite2D)
    ├── Desktop (Node2D)
    │   ├── Taskbar (Sprite2D)
    │   ├── DesktopIcons (Node2D)
    │   │   ├── ProjectsFolder (Sprite2D + Area2D + Collision)
    │   │   ├── TerminalIcon (Sprite2D + Area2D + Collision)
    │   │   └── ResumeFile (Sprite2D + Area2D + Collision)
    │   └── Windows (Node2D)
    │       ├── TerminalWindow (Control)
    │       └── FileExplorer (Control)
    └── UIElements (Node2D)
        ├── BackButton (Button)
        └── FullscreenButton (Button)
```

GitHub API Integration

- Use GitHub GraphQL API for contributions data
- Convert to pixel art texture (52x7 grid)
- Each day = 1 pixel, color by contribution count
- Display as poster in desk view, wallpaper in computer view

Responsive Camera Code Template

gdscript

```
# Main.gd
extends Node2D

@onready var camera = $Camera2D
@onready var desk_view = $DeskView
@onready var computer_view = $ComputerView

func _ready():

get_viewport().size_changed.connect(_on_viewport_size_changed)
    _setup_initial_view()

func _setup_initial_view():
    var screen_size = get_viewport().get_visible_rect().size
    if screen_size.x < 768:  # Mobile
        switch_to_computer_view()
    else:
        switch_to_desk_view()
```

Interactive Features Ideas

Computer Interface:

- Functional terminal with commands (ls, cd, cat resume.txt)
- Code editor showing real projects
- File explorer with portfolio sections
- Mini-games (Pong, Snake, sorting visualizer)
- Live GitHub data integration
- Screensaver with algorithms/fractals

Positioning:

- DeskView: (0, 0)
- ComputerView: (2000, 0) - off-screen

Current Status

- Project structure created ✅
- Need to fix CollisionShape2D warnings (assign RectangleShape2D)
- Next: Add placeholder visuals, implement camera switching

Key Technical Notes

- Use integer scaling for crisp pixels
- Area2D + CollisionShape2D for all clickable elements
- Export to HTML5 with custom shell for navbar
- Cache GitHub API data to avoid rate limits