

# Game Jam 2026: Development Log

Developer Log - Day 1

January 30, 2026

## Overview

Following the announcement of the Game Jam theme, my initial objective was to build a functional foundation by faithfully recreating the mechanics of Namco's *Battle City*. This log documents the first 24 hours of development.

## 1. Time Management & Workflow

To maximize productivity and maintain focus, I implemented the **Pomodoro Technique**.

- **Total Development Time:** 6 hours of active coding and asset creation.
- **Engine:** Godot Engine 4.6.

## 2. Technical Implementation

### 2.1 Movement & Input Logic

For the player tank, I utilized `Input.get_vector()` to capture player intent. To honor the retro 4-directional constraint, I implemented the following logic:

Listing 1: 4-Directional Movement Constraint

```
var direction = Input.get_vector("A", "D", "W", "S")  
  
if direction.x != 0:  
    direction.y = 0 # Lock Y axis if X is active
```

While functional, this approach introduced a certain "stiffness" in handling. I have flagged this for a potential input-buffer refactor in Day 2.

### 2.2 Physics & Combat

- **Collision:** Used `CharacterBody2D` with `move_and_collide()` for high-precision projectile detection.
- **Memory Management:** Currently considering an **Object Pool** pattern for projectiles to ensure performance stability as entity counts scale.

### 2.3 Enemy AI & Pathfinding

Enemy navigation was built using Godot's **AStar** algorithm.

- **Obstacle Avoidance:** Integrated directly with `TileMapLayer` data.
- **Current Limitation:** Inter-enemy collision is not yet implemented. I plan to iterate on the avoidance system to allow tanks to recognize each other as dynamic obstacles.

### 3. Critical Reflection & Pivot

By the end of the first day, the prototype reached a "feature-complete" state regarding the original *Battle City* mechanics. However, I concluded that the current framework lacks a unique competitive edge and feels overly derivative.

**Decision for Day 2 (Jan 31):** I will retain the core backend (Pathfinding, Physics, and Input logic) but will initiate a complete **Visual Redesign**. New assets will be created to pivot away from the Namco aesthetic toward a more original concept.

### Next Steps

Redraw all sprite sheets and environment tiles.

Refine the movement "buffer" to reduce control stiffness.

Update the `AStar` logic for dynamic unit avoidance.