CLI Configuration

Configuration files and options

Overview

Origami Engine uses three levels of configuration:

1. **Global Config** (`~/.origami/config.json`) - User-wide settings
2. **Engine Config** (`.origami/config.json`) - Engine settings
3. **Game Config** (`game.json`) - Project-specific settings

Global Configuration

Location

Path: `~/.origami/config.json`

Created: During `npm run begin` installation

Purpose: Stores engine installation path and global preferences

Format

```
{
  "enginePath": "D:/Projects/TypeScript/Origami-Engine",
  "version": "0.1.0",
  "installedAt": "2026-01-29T12:00:00.000Z",
  "preferences": {
    "autoUpdate": false,
    "telemetry": false,
    "editor": "code"
  }
}
```

Fields

`enginePath` (string, required)

Absolute path to engine installation.

Example:

```
"enginePath": "/home/user/origami-engine"
```

Usage: CLI uses this to locate engine files

`version` (string, required)

Current installed engine version.

Example:

```
"version": "0.1.0"
```

Usage: Update commands check this for compatibility

`installedAt` (string, required)

ISO 8601 timestamp of installation.

Example:

```
"installedAt": "2026-01-29T12:00:00.000Z"
```

Usage: Tracking and diagnostics

`preferences` (object, optional)

User preferences for CLI behavior.

Fields:

`autoUpdate` **(boolean)**:

```
"autoUpdate": false
```

- `true` - Check for updates on every command
- `false` - Only check when explicitly requested (default)

`telemetry` **(boolean)**:

```
"telemetry": false
```

- `true` - Send anonymous usage data
- `false` - Don't send data (default)

`editor` **(string)**:

```
"editor": "code"
```

- Editor command for `ori open` (e.g., "code", "vim", "subl")

Manual Editing

Edit the file directly:

```
# On Unix/Mac
nano ~/.origami/config.json

# On Windows
notepad %USERPROFILE%\.origami\config.json
```

Or use the CLI:

```
ori config --global --edit
```

Engine Configuration

Location

Path: `.origami/config.json` (engine root)

Created: During `npm run begin` installation

Purpose: Engine behavior, versioning, and templates

Format

```
{
  "version": "0.1.0",
  "lockVersion": false,
  "templateBranches": {
    "fresh": "template/fresh",
    "platformer": "template/platformer"
  },
  "migrations": {
    "0.2.0": [
      "rename-api-functions",
      "add-lifecycle-methods",
      "update-configs"
    ]
  },
  "debug": {
    "enableF3": true,
    "showFPS": true
  }
}
```

Fields

`version` (string, required)

Current engine version (semver).

Example:

```
"version": "0.1.0"
```

Usage: CLI uses this for update checks

`lockVersion` (boolean, required)

Prevents automatic updates.

Example:

```
"lockVersion": false
```

Values:

- `false` - Updates allowed (default)
- `true` - Updates blocked (stable production version)

Usage:

```
ori update
# ❌ Version is locked. Updates are disabled.
```

`templateBranches` (object, required)

Maps template names to git branches.

Example:

```
"templateBranches": {
  "fresh": "template/fresh",
  "platformer": "template/platformer",
  "topdown": "template/topdown"
}
```

Usage: `ori create` fetches these branches

`migrations` (object, required)

Maps versions to migration scripts.

Example:

```
"migrations": {
  "0.2.0": ["rename-api-functions", "add-lifecycle-methods"],
  "0.3.0": ["update-sprite-format"]
}
```

Usage: `ori update full` runs these for version jumps

`debug` (object, optional)

Debug mode settings.

Fields:

`enableF3` **(boolean)**:

```
"enableF3": true
```

- Enable F3 debug overlay

`showFPS` **(boolean)**:

```
"showFPS": true
```

- Show FPS counter in debug mode

`showBBoxes` **(boolean)**:

```
"showBBoxes": true
```

- Show collision boxes in debug mode

Manual Editing

From Engine Directory:

```
cd origami-engine
nano .origami/config.json
```

From Anywhere:

```
ori config --engine --edit
```

Game Configuration

Location

Path: `game.json` (game project root)

Created: During `ori create`

Purpose: Game metadata and settings

Format

```
{
  "name": "My Platformer",
  "version": "1.0.0",
  "author": "John Doe",
  "description": "A fun platformer game",
  "entryRoom": "rm_level1",
  "engineVersion": "0.1.0",
  "resolution": {
    "width": 640,
    "height": 360,
    "scale": 2
  },
  "fps": 60,
  "backgroundColor": "#2d2d2d"
}
```

Fields

`name` (string, required)

Game title.

Example:

```
"name": "Super Platformer"
```

`version` (string, required)

Game version (semver).

Example:

```
"version": "1.0.0"
```

`author` (string, optional)

Game creator.

Example:

```
"author": "John Doe"
```

`description` (string, optional)

Game description.

Example:

```
"description": "A fast-paced platformer with pixel art graphics"
```

`entryRoom` (string, required)

Initial room to load.

Example:

```
"entryRoom": "rm_menu"
```

Note: Must match a room name in `rooms/` folder

`engineVersion` (string, required)

Engine version used by this game.

Example:

```
"engineVersion": "0.1.0"
```

Usage: CLI checks compatibility before updates

`resolution` (object, optional)

Game canvas size.

Example:

```
"resolution": {
  "width": 640,
  "height": 360,
  "scale": 2
}
```

Fields:

- `width` (number) - Canvas width in pixels
- `height` (number) - Canvas height in pixels
- `scale` (number) - Display scale multiplier

`fps` (number, optional)

Target frame rate (default: 60).

Example:

```
"fps": 60
```

`backgroundColor` (string, optional)

Canvas background color (default: #000000).

Example:

```
"backgroundColor": "#2d2d2d"
```

Manual Editing

From Game Directory:

```
nano game.json
```

Using CLI:

```
ori config --game --edit
```

TypeScript Configuration

Location

Path: `tsconfig.json` (game project root)

Created: During `ori create`

Purpose: TypeScript compiler settings

Default Format

```
{
  "compilerOptions": {
    "target": "ES2022",
    "module": "ES2022",
    "moduleResolution": "node",
    "lib": ["ES2022", "DOM"],
    "strict": true,
    "esModuleInterop": true,
    "skipLibCheck": true,
    "resolveJsonModule": true,
    "outDir": "./dist",
    "rootDir": "./src"
  },
  "include": ["src/**/*", "objects/**/*"],
  "exclude": ["node_modules", "dist"]
}
```

Common Customizations

Strict Mode

Disable strict mode (not recommended):

```
{
  "compilerOptions": {
    "strict": false
  }
}
```

Target Version

Use older JS version:

```
{
  "compilerOptions": {
    "target": "ES2020"
  }
}
```

Paths Mapping

Add path aliases:

```
{
  "compilerOptions": {
    "baseUrl": ".",
    "paths": {
      "@objects/*": ["objects/*"],
      "@sprites/*": ["sprites/*"],
      "@rooms/*": ["rooms/*"]
    }
  }
}
```

Usage:

```
import { obj_player } from '@objects/obj_player';
```

Package Configuration

Location

Path: `package.json` (game project root)

Created: During `ori create`

Purpose: Node.js dependencies and scripts

Default Format

```
{
  "name": "my-platformer",
  "version": "1.0.0",
  "type": "module",
  "scripts": {
    "dev": "ori dev",
    "build": "ori build"
  },
  "dependencies": {
    "origami-runtime": "^0.1.0"
  },
  "devDependencies": {
    "typescript": "^5.3.3"
  }
}
```

Adding Custom Scripts

```
{
  "scripts": {
    "dev": "ori dev",
    "build": "ori build",
    "deploy": "ori build && gh-pages -d dist",
    "test": "vitest"
  }
}
```

Configuration Validation

Check All Configs

```
ori config --validate
```

Output:

```
✅ Global config valid
✅ Engine config valid
✅ Game config valid
✅ TypeScript config valid
```

Fix Invalid Configs

```
ori config --fix
```

Output:

```
🔄 Fixing configs...
✅ Added missing fields to game.json
✅ Updated engineVersion in game.json
✅ All configs valid
```

Configuration Presets

Create Custom Preset

File: `.origami/presets/my-preset.json`

```
{
  "name": "My Custom Preset",
  "game": {
    "resolution": {
      "width": 800,
      "height": 600,
      "scale": 1
    },
    "fps": 60
  },
  "typescript": {
    "compilerOptions": {
      "strict": true
    }
  }
}
```

Apply Preset

```
ori config --preset my-preset
```

Output:

```
✅ Applied preset: My Custom Preset
✅ Updated game.json
✅ Updated tsconfig.json
```

Environment-Specific Configs

Development

File: `.origami/config.dev.json`

```
{
  "debug": {
    "enableF3": true,
    "showFPS": true,
    "logLevel": "debug"
  }
}
```

Production

File: `.origami/config.prod.json`

```
{
  "debug": {
    "enableF3": false,
    "showFPS": false,
    "logLevel": "error"
  }
}
```

Load Specific Config

```
ori dev --config .origami/config.dev.json
ori build --config .origami/config.prod.json
```

Best Practices

Version Locking

Lock version in production:

```
{
  "lockVersion": true
}
```

Keep Configs in Git

Add to `.gitignore`:

```
# Don't commit global config
~/.origami/config.json

# Commit game configs
!game.json
!tsconfig.json
!.origami/config.json
```

Backup Before Editing

```
cp game.json game.json.bak
nano game.json
```

Use Version Control

```
git add game.json
git commit -m "Update game resolution"
```

Troubleshooting

Config Not Found

Error:

```
❌ Config file not found: ~/.origami/config.json
```

Fix:

```
# Reinstall engine
cd origami-engine
npm run begin
```

Invalid JSON

Error:

```
❌ Invalid JSON in game.json
   Unexpected token } at line 15
```

Fix: Validate JSON syntax

```
# Use JSON linter
cat game.json | jq .

# Or fix manually
nano game.json
```

Version Mismatch

Error:

```
❌ Engine version mismatch
   Game requires: v0.2.0
   Engine version: v0.1.0
```

Fix:

```
ori update v0.2.0
```

Next Steps

- **30-cli-commands.md** - CLI commands
- **32-cli-templates.md** - Template system
- **11-cli-reference.md** - Quick reference

← Back to Index