

Deployment

Publishing your game to the web

Overview

Origami Engine games are static HTML/JavaScript applications that can be hosted anywhere. No server-side code needed!

GitHub Pages

Setup

1. Initialize Git (if not already):

```
cd your-game
git init
git add .
git commit -m "Initial commit"
```

2. Create GitHub Repository:

- Go to github.com
- Click "New repository"
- Name it (e.g., "my-platformer")
- Don't initialize with README

3. Push to GitHub:

```
git branch -M main
git remote add origin https://github.com/username/my-platformer.git
git push -u origin main
```

4. Enable GitHub Pages:

- Go to repository Settings → Pages
- Source: Deploy from branch
- Branch: main, folder: / (root)
- Click Save

5. Access Your Game:

- Wait ~1 minute for deployment
- Visit: `https://username.github.io/my-platformer/`

Custom Domain (Optional)

1. Add `CNAME` file with your domain:

```
game.yourdomain.com
```

2. Configure DNS with your domain provider:

- Type: CNAME
- Name: game
- Value: username.github.io

3. In GitHub Pages settings, enter custom domain

itch.io

Preparation

1. **Zip your game folder:**

- Include all files: index.html, js/, lib/, sprites/, etc.
- Don't include: node_modules/, src/, .git/

Windows: Right-click folder → Send to → Compressed folder **macOS:** Right-click folder → Compress **Linux:** `zip -r my-game.zip my-game/`

Upload

1. Go to itch.io and create account
2. Click "Upload new project"
3. Fill in game details:
 - Title
 - Project URL (unique slug)
 - Description
 - Screenshots
4. Upload your ZIP file
5. **Important settings:**
 - Kind of project: **HTML**
 - Check: **"This file will be played in the browser"**
 - Viewport dimensions: Match your game.json (e.g., 640 x 480)
 - Check: **"Mobile friendly"** (if applicable)
 - Embed options: **"Click to launch in fullscreen"**
6. Save & view page

Advantages

- Built-in community
 - Analytics dashboard
 - Easy updates (just upload new ZIP)
 - Payment system (for paid games)
 - Download support (besides browser play)
-

Netlify

Method 1: Drag and Drop

1. Go to netlify.com and create account
2. Click "Add new site" → "Deploy manually"
3. Drag your game folder into the drop zone

4. Wait for deployment
5. Get your URL: `https://random-name.netlify.app`

Method 2: GitHub Integration

1. Push code to GitHub (see GitHub Pages section)
2. In Netlify, click "Add new site" → "Import from Git"
3. Connect GitHub account
4. Select your repository
5. Build settings:
 - Build command: (leave empty)
 - Publish directory: `/`
6. Click "Deploy"

Custom Domain

1. In Netlify: Site settings → Domain management
2. Click "Add custom domain"
3. Follow DNS configuration instructions

Advantages

- Instant deploys
- Automatic HTTPS
- Preview deployments for branches
- Serverless functions (advanced)

Vercel

Setup

1. Install Vercel CLI:

```
npm install -g vercel
```

2. Deploy:

```
cd your-game  
vercel
```

3. Follow prompts:

- Set up and deploy: Yes
- Project name: (auto-detected)
- Deploy: Yes

4. Get deployment URL

GitHub Integration

1. Go to vercel.com and create account

2. Import repository from GitHub

3. Build settings:

- Framework Preset: Other
- Build Command: (leave empty)
- Output Directory: `./`

4. Deploy

Advantages

- Very fast global CDN
- Automatic HTTPS
- Great GitHub integration
- Preview deployments

Firebase Hosting

Setup

1. Install Firebase CLI:

```
npm install -g firebase-tools
```

2. Login:

```
firebase login
```

3. Initialize:

```
cd your-game  
firebase init hosting
```

- Select: Create new project or use existing
- Public directory: `.` (current directory)
- Configure as single-page app: No
- Overwrite index.html: No

4. Deploy:

```
firebase deploy
```

5. Get URL: `https://your-project.web.app`

Advantages

- Google infrastructure
- Excellent reliability
- Custom domains included
- Good analytics

Cloudflare Pages

Setup

1. Go to pages.cloudflare.com
2. Create account
3. Connect GitHub repository
4. Build settings:
 - Build command: (leave empty)
 - Build output directory: `/`

5. Deploy

Advantages

- Global CDN (very fast)
 - Unlimited bandwidth
 - Free SSL
 - DDoS protection
-

Self-Hosting

Any web server can host your game!

Requirements

- Web server (Apache, Nginx, IIS, etc.)
- Support for static files
- HTTPS recommended

Upload via FTP/SFTP

1. Connect to your web server
2. Upload all game files to public directory
3. Ensure index.html is accessible
4. Test in browser

Common Hosting Providers

- **Shared Hosting:** Bluehost, HostGator, SiteGround
 - **VPS:** DigitalOcean, Linode, Vultr
 - **Cloud:** AWS S3 + CloudFront, Google Cloud Storage
-

Pre-Deployment Checklist

Testing

- ☒ Test in multiple browsers (Chrome, Firefox, Safari, Edge)
- ☒ Test on mobile devices
- ☒ Check all levels/rooms load correctly
- ☒ Verify all sprites display
- ☒ Test game completion (start to finish)
- ☒ Check for console errors (F12)

Optimization

- ☒ Remove `show_debug_message()` calls
- ☒ Disable debug mode by default
- ☒ Compress PNG sprites (use TinyPNG, ImageOptim)
- ☒ Minify JavaScript (if desired)
- ☒ Test loading time

Files to Include

- ☒ index.html
- ☒ game.json
- ☒ js/ folder (built game code)
- ☒ lib/ folder (engine runtime)
- ☒ sprites/ folder
- ☒ rooms/ folder

Files to Exclude

- ☒ node_modules/
 - ☒ src/ (TypeScript source)
 - ☒ objects/ (TypeScript source)
 - ☒ .git/
 - ☒ tsconfig.json
 - ☒ package.json (optional, not needed for playing)
-

Updating Your Game

GitHub Pages

```
# Make changes
git add .
git commit -m "Update game"
git push
# Wait ~1 minute for deployment
```

itch.io

1. Create new ZIP with updated files
2. Go to your game's edit page
3. Upload new version
4. Players see update immediately

Netlify/Vercel/Others

- Automatic deployment on git push (if connected to GitHub)
- Or re-upload manually

Analytics

Google Analytics

Add to `index.html` before `</head>` :

```
<!-- Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/js?id=G-XXXXXXXXXX"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());
  gtag('config', 'G-XXXXXXXXXX');
</script>
```

Tracks:

- Page views
- Session duration
- User locations
- Device types

itch.io Analytics

Built-in dashboard shows:

- Views
- Downloads
- Plays (browser)
- Ratings

Monetization

itch.io

- Set price or "Pay what you want"
- Donations button
- itch.io handles payments (takes small cut)

Ads

Simple ad integration:

1. Sign up for ad network (e.g., Google AdSense)

2. Add ad code to index.html
3. Place ads outside game canvas

Note: Consider user experience. Too many ads = bad reviews.

Premium Features

- Use localStorage to track purchases
 - Implement unlock codes
 - External payment → provide unlock code
-

Common Issues

Game Not Loading

Check:

- All files uploaded?
- Correct directory structure?
- Check browser console for errors
- Verify file paths are relative (not absolute)

Sprites Not Showing

Solution:

- Ensure sprites/ folder uploaded
- Check capitalization (case-sensitive on Linux servers)
- Verify metadata.json files present

Slow Loading

Solutions:

- Compress PNG sprites
 - Use CDN (many platforms provide this automatically)
 - Reduce number of frames in animations
 - Consider lazy loading for large assets
-

Security & Privacy

Don't Include

- API keys (if any)
- Personal information
- Database credentials
- Source code (unless open source)

Content Security

If implementing saves:

- Validate localStorage data
- Don't trust client-side validation
- Encrypt sensitive data

Marketing Your Game

1. Create a Press Kit:

- Screenshots
- GIFs of gameplay
- Description
- Logo

2. Share on Social Media:

- Twitter with #gamedev, #indiegame
- Reddit: r/WebGames, r/IndieGaming
- Discord communities

3. Create a Trailer:

- 30-60 seconds
- Show best gameplay moments
- Upload to YouTube

4. Get Feedback:

- Post to game dev forums
 - Request reviews
 - Iterate based on feedback
-

Next Steps

- [10-debugging.md](#) - Testing before deployment
 - [40-common-patterns.md](#) - Polish your game
 - [42-performance.md](#) - Optimization techniques
-

[← Back to Index](#)