Sprites

Working with sprites and animation

Overview

Sprites are the visual assets for your game objects. Each sprite is stored in its own folder with PNG images and a metadata file.

Folder Structure

```
sprites/
├── spr_player/
│   ├── metadata.json
│   ├── frame_0.png
│   ├── frame_1.png  (optional - for animation)
│   └── frame_2.png  (optional)
├── spr_enemy/
│   ├── metadata.json
│   └── frame_0.png
└── spr_wall/
    ├── metadata.json
    └── frame_0.png
```

Requirements:

- Each sprite folder must contain `metadata.json`
- Frame files must be named `frame_0.png`, `frame_1.png`, etc.
- Start numbering at 0
- No gaps in numbering

metadata.json

Basic Example

```
{
  "origin": { "x": 16, "y": 16 },
  "fps": 0
}
```

With Animation

```
{
  "origin": { "x": 16, "y": 16 },
  "fps": 10
}
```

With Custom Collision Box

```
{
  "origin": { "x": 16, "y": 16 },
  "fps": 10,
  "bbox": {
    "left": 4,
    "top": 4,
    "right": 28,
    "bottom": 28
  }
}
```

Properties

`origin`

The **pivot point** of the sprite.

Common values:

- `{"x": 0, "y": 0}` - Top-left corner
- `{"x": 16, "y": 16}` - Center (for 32x32 sprite)
- `{"x": 16, "y": 32}` - Bottom-center (for 32x32 sprite)

Usage:

- Determines rotation point
- Determines sprite position relative to object's x/y
- Affects collision detection reference point

```
{
  "origin": { "x": 16, "y": 16 }
}
```

`fps`

Animation frames per second.

Values:

- `0` - No animation (static sprite)
- `1-60` - Animation speed
- `10` - Common for character animations
- `30` - Fast animation

```
{
  "origin": { "x": 16, "y": 16 },
  "fps": 10
}
```

`bbox` (Optional)

Custom collision bounding box.

If not specified: Entire sprite is used for collision (0, 0, width, height)

If specified: Define inset from edges

```
{
  "origin": { "x": 16, "y": 16 },
  "fps": 0,
  "bbox": {
    "left": 4,    // 4 pixels from left edge
    "top": 4,     // 4 pixels from top edge
    "right": 28,  // 4 pixels from right edge (32 - 4)
    "bottom": 28  // 4 pixels from bottom edge (32 - 4)
  }
}
```

Use case: Tighter collision for characters (ignore transparent edges)

Creating Sprites

Image Editors

Recommended tools:

- **Aseprite** - Pixel art editor with animation
- **GIMP** - Free, powerful image editor
- **Photoshop** - Professional tool
- **Piskel** - Free online pixel art tool
- **GraphicsGale** - Free pixel art animation

Image Requirements

Format: PNG with transparency

Size recommendations:

- `16x16` - Small sprites, UI elements
- `32x32` - Characters, items
- `64x64` - Large characters, bosses
- `128x128` - Very large objects

Tips:

- Use transparent backgrounds

- Keep power-of-2 sizes when possible (16, 32, 64, 128)
- Consistent size within sprite set
- Export with alpha channel

Frame Naming

Correct:

```
frame_0.png
frame_1.png
frame_2.png
```

Incorrect:

```
frame0.png      ❌ Missing underscore
frame_1.jpg     ❌ Wrong format
frame_01.png    ❌ Leading zero
```

Using Sprites in Code

Assigning Sprites

```
create(): void {
  this.sprite_index = 'spr_player';
  this.image_speed = 1.0; // Play at normal speed
}
```

Animation Control

```
// Play animation
this.image_speed = 1.0;

// Faster animation
this.image_speed = 2.0;

// Slower animation
this.image_speed = 0.5;

// Pause animation
this.image_speed = 0;

// Specific frame
this.image_index = 0; // First frame
this.image_index = 2; // Third frame
```

Switching Sprites

```
step(): void {
  if (keyboard_check(vk_d) || keyboard_check(vk_a)) {
    // Moving
    this.sprite_index = 'spr_player_run';
    this.image_speed = 1.0;
  } else {
    // Idle
    this.sprite_index = 'spr_player_idle';
    this.image_speed = 1.0;
  }
}
```

Flipping Sprites

```
step(): void {
  if (keyboard_check(vk_d)) {
    this.image_xscale = 1; // Face right
  }
  if (keyboard_check(vk_a)) {
    this.image_xscale = -1; // Face left (flip)
  }
}
```

Animation Patterns

Idle Animation

```
create(): void {
  this.sprite_index = 'spr_player_idle';
  this.image_speed = 0.5; // Slow idle animation
}
```

Sprite: 2-4 frames, subtle movement

Run Cycle

```
step(): void {
  if (this.hspeed !== 0) {
    this.sprite_index = 'spr_player_run';
    this.image_speed = 1.0;
  } else {
    this.sprite_index = 'spr_player_idle';
    this.image_speed = 0.5;
  }
}
```

Sprite: 6-8 frames, looping

Attack Animation

```
private attacking: boolean = false;
private attackTimer: number = 0;

step(): void {
  if (keyboard_check_pressed(vk_space) && !this.attacking) {
    this.attacking = true;
    this.sprite_index = 'spr_player_attack';
    this.image_index = 0; // Start from first frame
    this.image_speed = 1.5; // Fast attack
  }

  if (this.attacking) {
    this.attackTimer++;
    // Attack animation is 6 frames, 60 FPS
    // 6 frames / 1.5 speed = 4 frames duration
    if (this.attackTimer >= 4) {
      this.attacking = false;
      this.attackTimer = 0;
      this.sprite_index = 'spr_player_idle';
    }
  }
}
```

Sprite: 4-6 frames, non-looping

Advanced Sprite Manipulation

Rotation

```
step(): void {
  // Rotate sprite
  this.image_angle += 5; // Degrees per frame

  // Point at mouse
  this.image_angle = point_direction(this.x, this.y, mouse_x, mouse_y);
}
```

Scaling

```
create(): void {
  // Make sprite larger
  this.image_xscale = 2.0;
  this.image_yscale = 2.0;

  // Make sprite smaller
  this.image_xscale = 0.5;
  this.image_yscale = 0.5;
}
```

Transparency

```
draw(): void {
  // Fade out
  this.image_alpha = 0.5;

  // Flashing effect
  this.image_alpha = (Math.sin(get_timer() / 100000) + 1) / 2;
}
```

Common Issues

Sprite Not Appearing

Checklist:

- ✅ `metadata.json` exists in sprite folder
- ✅ PNG files named `frame_0.png`, `frame_1.png`, etc.
- ✅ No gaps in frame numbering
- ✅ Sprite folder name matches `sprite_index`
- ✅ Check browser console (F12) for errors

Animation Not Playing

Checklist:

- ✅ `fps` in metadata.json is greater than 0
- ✅ Multiple frame files exist (frame_0, frame_1, etc.)
- ✅ `image_speed` is not 0
- ✅ Sprite has more than one frame

Collision Box Wrong Size

Solution: Add custom `bbox` to metadata.json

```
{
  "origin": { "x": 16, "y": 16 },
  "fps": 10,
  "bbox": {
    "left": 8,
    "top": 8,
    "right": 24,
    "bottom": 24
  }
}
```

Enable debug mode (F3) to visualize collision boxes!

Best Practices

1. **Naming Convention**: Use descriptive names
   - `spr_player_idle`, `spr_player_run`, `spr_player_jump`
   - Not: `spr_player1`, `spr_player2`
2. **Organization**: Group related sprites
   - Keep player sprites together
   - Keep enemy sprites together
   - Keep UI sprites together
3. **Size Consistency**: Use same size for related sprites
   - All player animations should be same dimensions
   - Makes switching sprites seamless
4. **Origin Point**: Keep consistent within animation sets
   - Use same origin for idle, run, jump
   - Prevents "jumping" when switching sprites
5. **Frame Count**: Keep it reasonable
   - 2-4 frames for idle
   - 6-8 frames for run cycles
   - 4-6 frames for attacks
   - More frames = larger file size

Next Steps

- **04-gameobjects.md** - Using sprites in GameObjects
- **09-drawing.md** - Drawing sprites with functions
- **40-common-patterns.md** - Animation state machines

← Back to Index