

## Lesson 2 Quick Reference Guide

### Animated Sprites: Walk Cycle & Idle Poses

**Duration:** 40 minutes

**Software:** Adobe Photoshop CC 2024 + Godot 4.5

**Prerequisites:** Completed Lesson 1

---

#### What We're Building Today

##### Photoshop (20 min):

- Create 3-4 walk cycle frames (character walking animation)
- Create idle pose (character standing still)
- Organize frames in a sprite sheet
- Export animation-ready PNG

##### Godot (20 min):

- Replace Sprite2D with AnimatedSprite2D
- Set up "walk" and "idle" animations
- Code smooth animation transitions
- Make character animate while playing!

**Result:** Your character walks when moving and stands idle when stopped! 

---

#### Key Vocabulary

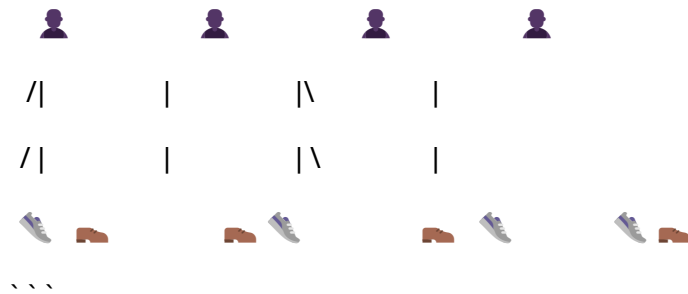
- **Walk Cycle:** Sequence of frames showing walking motion
- **Frame:** Single image in an animation sequence
- **Idle Animation:** Character standing still (can breathe, blink, sway)
- **FPS (Frames Per Second):** How fast animation plays (8 FPS = good for walk cycles)
- **Animation Loop:** Animation repeats continuously
- **Keyframe:** Important pose in animation (contact, passing, etc.)
- **AnimatedSprite2D:** Godot node that plays sprite animations

---

## 🎨 Understanding Walk Cycles

### The 4 Basic Positions:

Frame 1: Contact    Frame 2: Passing    Frame 3: Contact    Frame 4: Passing  
(foot down)    (legs cross)    (other foot)    (legs cross)



**\*\*Minimum walk cycle:\*\*** 2 frames (left foot forward, right foot forward)

**\*\*Better walk cycle:\*\*** 4 frames (includes passing positions)

**\*\*Professional walk cycle:\*\*** 8+ frames (super smooth!)

**\*\*Today we'll make 3-4 frames - perfect for learning!\*\***

---

### **\*\*Reference Examples:\*\***

Before starting, search for:

- "sprite walk cycle reference"
- "2D character walking animation"
- "pixel art walk cycle"

**\*\*Study professional games:\*\***

- Mario (classic smooth walk)
- Celeste (modern pixel walk)
- Hollow Knight (detailed walk cycle)

---

## 🎨 **\*\*Part 1: Photoshop - Create Walk Cycle (20 min)\*\***

### **\*\*Step 1: Open Your Previous Work (1 min)\*\***

**\*\*Open your Lesson 1 file:\*\***

\ \ \

Desktop/GameArt\_Year9/working\_files/player\_sprite.psd

\ \ \

**\*\*OR start fresh:\*\***

- **\*\*File → New\*\***

- Width: **\*\*256 pixels\*\*** (fits 4 frames of 64px)

- Height: **\*\*64 pixels\*\***

- Background: **\*\*Transparent\*\***

---

### **\*\*Step 2: Find Walk Cycle Reference Images (3 min)\*\***

**\*\*Option A: Download Existing Walk Cycle\*\***

Search for:

- "character walk cycle sprite sheet"
- "[your favorite game] walking sprites"
- "pixel art walk animation"

**\*\*Download 3-4 images showing different walking poses\*\***

**\*\*Option B: Use Your Lesson 1 Character\*\***

If you already have different poses, great! We'll modify them.

---

**### \*\*Step 3: Create Frame 1 - Contact Pose (4 min)\*\***

**\*\*This is the character with one foot fully on the ground\*\***

**#### \*\*Import or Draw Frame 1:\*\***

**\*\*If you have a reference:\*\***

1. **\*\*File → Place Embedded\*\*** your walk frame 1
2. Remove background (Magic Wand + Eraser - you learned this!)
3. Resize to 60 pixels tall ( ` Ctrl+T` , hold Shift)

**\*\*If drawing from scratch:\*\***

1. Create **\*\*New Layer\*\*** → Name: "Walk\_Frame1"
2. Use **\*\*Brush Tool (B)\*\*** to draw:
  - Body centered
  - Left leg forward (foot flat)

- Right leg back (foot raised)
- Arms opposite to legs (right arm forward)

**\*\*Position:\*\*** Place on LEFT side of canvas (x: 32)

**\*\*💡 Drawing Tips:\*\***

- Keep it simple - stick figures work!
- Legs in opposite positions to arms (natural walking)
- Body leans slightly forward

---

**### \*\*Step 4: Create Frame 2 - Passing Pose (4 min)\*\***

**\*\*Character with legs crossing (mid-stride)\*\***

**#### \*\*Method A: Duplicate & Modify\*\***

1. Select "Walk\_Frame1" layer
2. `Ctrl+J` / `Cmd+J` (duplicate)
3. Rename: "Walk\_Frame2"
4. Press **\*\*V\*\*** (Move Tool)
5. Drag to the RIGHT of Frame 1 (leave small gap)

**#### \*\*Modify the Pose:\*\***

**\*\*Using Free Transform & Lasso:\*\***

1. Press **\*\*L\*\*** for **\*\*Lasso Tool\*\***

2. Draw selection around **right leg**
3. `Ctrl+T` / `Cmd+T` (Transform)
4. Move leg forward (legs should be close together)
5. Press **Enter**

**Repeat for arms:**

- Left arm moves back
- Right arm moves forward
- Body upright

**Result:** Legs close together, mid-stride position

---

**Step 5: Create Frame 3 - Contact Pose (Opposite) (3 min)**

**Same as Frame 1, but opposite foot forward**

**Quick Method: Duplicate Frame 1**

1. Duplicate "Walk\_Frame1" layer
2. Rename: "Walk\_Frame3"
3. Move to RIGHT of Frame 2

**Flip the Pose:**

**Option A: Manual Flip**

1. Select the leg/arm parts with Lasso

2. Swap their positions

**\*\*Option B: Use Transform\*\***

1. `Ctrl+T` / `Cmd+T`

2. **\*\*Right-click\*\*** → **\*\*Flip Horizontal\*\***

3. This flips the whole sprite

**\*\*💡 Tip:\*\*** If you flip the whole sprite, you might need to adjust details to keep it looking natural!

---

**### \*\*Step 6: Create Frame 4 - Passing Pose (Optional) (2 min)\*\***

**\*\*For a 4-frame walk cycle, duplicate Frame 2 and flip it\*\***

1. Duplicate "Walk\_Frame2"

2. Rename: "Walk\_Frame4"

3. Move to RIGHT of Frame 3

4. Modify leg/arm positions (opposite of Frame 2)

**\*\*Or skip this if you're short on time - 3 frames works fine!\*\***

---

**### \*\*Step 7: Create Idle Pose (3 min)\*\***

**\*\*Character standing still - this goes in same sprite sheet!\*\***

#### **\*\*Create New Document Section (or extend canvas):\*\***

**\*\*Option A: Extend Current Canvas\*\***

1. **\*\*Image → Canvas Size\*\***
2. Width: **\*\*320 pixels\*\*** (add 64px for idle frame)
3. Anchor: Left (adds space to the right)
4. Click **\*\*OK\*\***

**\*\*Option B: New Layer in Current Document\*\***

- Just add idle frame to the far right of your current layout

#### **\*\*Create Idle Frame:\*\***

**\*\*If you have a standing reference:\*\***

1. Import with **\*\*Place Embedded\*\***
2. Remove background
3. Position to RIGHT of walk frames

**\*\*If drawing:\*\***

1. New layer: "Idle\_Frame"
2. Draw character standing naturally:
  - Feet together or slightly apart
  - Arms at sides or on hips
  - Relaxed posture
  - Can have slight lean or tilt



**\*\*💡 Advanced Idle:\*\*** Make 2 idle frames that are slightly different - creates subtle breathing/swaying animation!

---

### **\*\*Your Canvas Should Look Like:\*\***

`, `

[Walk 1] [Walk 2] [Walk 3] [Walk 4] [Idle]

(Contact)(Passing)(Contact)(Passing)(Stand)

`, `

**\*\*Or minimum version:\*\***

`, `

[Walk 1] [Walk 2] [Walk 3] [Idle]

`, `

---

### **\*\*Step 8: Clean Up & Label (1 min)\*\***

**\*\*Organize your Layers panel:\*\***

`, `

Layers Panel:

Walk\_Frame1


Walk\_Frame2

Walk\_Frame3

Walk\_Frame4 (optional)

Idle\_Frame

\\

**Hide any reference layers** (  icon)

---

**Step 9: Export Animation Sprite Sheet (2 min)**

**File → Export → Export As...**

**Settings:**

| Setting | Value |

|-----|-----|

| **Format** | **PNG** |

| **Transparency** | **✓ Checked** |

| **Smaller File** | **✓ Checked** |

**Save As:**

\\

Desktop/GameArt\_Year9/exported\_sprites/player\_animated.png

\\

**Click Export**

**Also save PSD:**

\\

Desktop/GameArt\_Year9/working\_files/player\_animated.psd

\\ \

---

## ## \*\*Photoshop Checklist\*\*

\\ \

- ☐ Walk cycle created (3-4 frames minimum)
- ☐ Idle pose created (1 frame)
- ☐ All backgrounds removed (transparent)
- ☐ Frames arranged in a row left to right
- ☐ Layers properly named
- ☐ PNG exported to exported\_sprites/
- ☐ PSD saved to working\_files/
- ☐ Preview PNG shows all frames clearly

\\ \

---

## ## \*\*Part 2: Import to Godot & Animate (20 min)\*\*

### ### \*\*Step 1: Copy New Sprite to Godot (1 min)\*\*

**\*\*Copy:\*\***

\\ \

Desktop/GameArt\_Year9/exported\_sprites/player\_animated.png

\\ \

**\*\*To:\*\***

\\ \

Desktop/Year9\_Platformer/assets/sprites/player\_animated.png

\\ \

---

### \*\*Step 2: Open Godot Project (1 min)\*\*

1. Launch \*\*Godot 4.5\*\*
2. Open \*\*Year9\_Platformer\*\*
3. Open \*\*level\_1.tscn\*\*

---

### \*\*Step 3: Configure Import Settings (3 min)\*\*

**\*\*This step is CRITICAL for pixel art! If you skip this, your sprites will look blurry!\*\***

#### \*\*Part A: Select Your Sprite File (30 seconds)\*\*

1. Look at \*\*FileSystem panel\*\* (bottom-left corner)
2. Navigate to: `res://assets/sprites/`
3. **\*\*Single-click\*\*** on `player\_animated.png`
4. Filename should be **\*\*highlighted in blue\*\***

#### \*\*Part B: Change Compression Settings (1 min)\*\*

**\*\*Location: Import Tab (top of screen)\*\***

1. Look at **top of screen** for tabs
2. Click the **Import** tab (next to Scene tab)
3. Find the **Compress** section:

...

Compress

Mode: [VRAM Compressed ▼]

...

4. Click the **Mode** dropdown
5. Select: **Lossless**

#### #### **Part C: Change Texture Filter Settings (1 min)**

**Location: Inspector Panel (right side of screen)**

**IMPORTANT:** Make sure `player\_animated.png` is still selected in FileSystem!

1. Look at **Inspector panel** (right side)
2. **Scroll down** in Inspector until you find:

...

Texture (or Sampling)

Filter: [Linear ▼]

Repeat: [Disabled ▼]

...

3. Click the **Filter** dropdown
4. Select: **Nearest**

#### #### **\*\*Part D: Optional - Turn Off Mipmaps (30 seconds)\*\***

**\*\*Location: Import Tab\*\***

In the Import tab, find:

\ \ \

Mipmaps

Generate: [On]

\ \ \

1. Click to turn it **\*\*Off\*\***

#### #### **\*\*Part E: Apply Changes - CRITICAL!\*\***

1. Scroll to **\*\*bottom\*\*** of Import tab
2. Click **\*\*Reimport\*\*** button
3. Wait 1-2 seconds

#### #### **\*\*Part F: Verify Settings Worked\*\***

- Look at sprite preview in FileSystem
- Pixels should be **\*\*sharp and clear\*\*** (not blurry)

**\*\*Quick Settings Summary:\*\***

\ \ \

Import Tab (top):

✓ Compress → Mode = Lossless

✓ Mipmaps → Generate = Off

✓ Reimport clicked

Inspector Panel (right):

✓ Texture → Filter = Nearest

✓ Texture → Repeat = Disabled

...`

---

### \*\*Step 4: Replace Sprite2D with AnimatedSprite2D (3 min)\*\*

\*\*We need to upgrade from static sprite to animated sprite!\*\*

#### \*\*Delete Old Sprite2D:\*\*

1. Select \*\*Player\*\* node
2. Find \*\*Sprite2D\*\* child node
3. \*\*Right-click → Delete\*\*
4. Click \*\*OK\*\*

#### \*\*Add AnimatedSprite2D:\*\*

1. Select \*\*Player\*\* node
2. Click \*\*++\*\* (Add Child Node)
3. Search: \*\*AnimatedSprite2D\*\*
4. Click \*\*Create\*\*

**\*\*Your Player node should now have:\*\***

...

Player (CharacterBody2D)

└─ AnimatedSprite2D ← NEW!

└─ CollisionShape2D

...

---

**### \*\*Step 5: Create SpriteFrames Resource (3 min)\*\***

**\*\*This is where we store all our animations!\*\***

1. Select **\*\*AnimatedSprite2D\*\*** node
2. **\*\*Inspector → Sprite Frames\*\*** property
3. Click **\*\*[empty]\*\*** dropdown
4. Select **\*\*New SpriteFrames\*\***

**\*\*You should see:\*\*** `[SpriteFrames]` appear

5. **\*\*Double-click\*\*** `[SpriteFrames]` to open editor

**\*\*SpriteFrames Editor opens at bottom of screen!\*\***


---

**### \*\*Step 6: Create "idle" Animation (4 min)\*\***



**\*\*The SpriteFrames editor should show:\*\***

...

Animation Frames:	
default ▼ [Rename] [  Add ] [Delete]	
<div></div>	
(empty - drag frames here)	
<div></div>	

...

**#### \*\*Rename "default" to "idle":\*\***

1. Click **\*\*Rename\*\*** button
2. Type: **\*\*idle\*\***
3. Press **\*\*Enter\*\***

**#### \*\*Add Idle Frame:\*\***

1. Find `player\_animated.png` in **\*\*FileSystem\*\*** panel
2. **\*\*Drag it\*\*** into the frames area

**\*\*"Select Frames" dialog appears - shows your sprite sheet in a grid\*\***

3. **\*\*Click ONLY the idle frame\*\*** (rightmost frame)
4. Click **\*\*Add X Frame(s)\*\*** button

✓ Idle frame appears in frames list!

#### \*\*Set Animation Speed:\*\*

- \*\*Speed (FPS):\*\* 5 (idle is slow)

- \*\*Loop:\*\* ✓ Checked

---

### \*\*Step 7: Create "walk" Animation (4 min)\*\*

#### \*\*Add New Animation:\*\*

1. Click **+** Add Animation button

2. Name: walk

3. Press Enter

#### \*\*Add Walk Frames:\*\*

1. Drag `player\_animated.png` into frames area

2. "Select Frames" dialog appears

**Select your walk cycle frames:**

- Click Frame 1 (first walk frame)

- Hold Shift + Click Frame 3 or 4 (last walk frame)

- This selects all walk frames!

3. Click Add X Frame(s)

✓ You should see 3-4 walk frames in sequence!

#### **\*\*Set Walk Animation Speed:\*\***

- **\*\*Speed (FPS):\*\*** **\*\*8\*\*** (smooth walking)

- **\*\*Loop:\*\*** ✓ Checked

**\*\*Your SpriteFrames Should Have:\*\***

...

Animations:

└ idle [1 frame, 5 FPS, loop]

└ walk [3-4 frames, 8 FPS, loop]

**Close SpriteFrames editor** (or keep open to preview!)

---

## Step 8: Test Animations in Editor (1 min)

**Preview without running the game!**

**With AnimatedSprite2D selected:**

1. **Inspector → Animation:** Select "walk"
2. **Playing:** ✓ Check this box

**Character should walk in viewport!** 

Try "idle" - character stands still!


**Uncheck Playing when done**

---

## Step 9: Code Animation Transitions (5 min)

**Make animations play automatically based on movement!**

**Open player.gd:**

1. Select **Player** node
2. Click **script icon** (  )

### Update Animation Code:

Find `_physics_process` function.

### REMOVE this (from Lesson 1):

```
gdscript
```

```
# --- SPRITE FLIPPING ---
```

```
if direction > 0:
```

```
    $Sprite2D.flip_h = false
```

```
elif direction < 0:
```

```
    $Sprite2D.flip_h = true
```

### REPLACE with this:

```
gdscript
```

```
func _physics_process(delta):
```

```
    # --- APPLY GRAVITY ---
```

```
    if not is_on_floor():
```

```
        velocity += get_gravity() * delta
```

```
    # --- JUMP CODE (DISABLED FOR LESSON 1-2) ---
```

```
    # if Input.is_action_just_pressed("ui_accept") and is_on_floor():
```

```
        # velocity.y = JUMP_VELOCITY
```

```
    # --- GET PLAYER INPUT ---
```

```
    var direction = Input.get_axis("ui_left", "ui_right")
```

```
    # --- MOVE THE PLAYER ---
```

```
    if direction:
```

```
        velocity.x = direction * SPEED
```

```
    else:
```

```
        velocity.x = move_toward(velocity.x, 0, SPEED)
```

```

# === ANIMATION CONTROL === (NEW CODE!)

# Play walk when moving, idle when stopped

if direction != 0:

    $AnimatedSprite2D.play("walk")

    # Flip sprite to face movement direction

    $AnimatedSprite2D.flip_h = (direction < 0)

else:

    $AnimatedSprite2D.play("idle")


# --- EXECUTE THE MOVEMENT ---

move_and_slide()

```

**Save:** Ctrl+S / Cmd+S

---

### Understanding the Animation Code:

gdscript

if direction != 0:

**Meaning:** If player is pressing left OR right...

gdscript

\$AnimatedSprite2D.play("walk")

**Meaning:** Play the "walk" animation

gdscript

\$AnimatedSprite2D.flip\_h = (direction < 0)

**Meaning:**

- flip\_h = flip horizontally
- direction < 0 = moving left
- Result: Faces left when moving left, right when moving right!

gdscript

else:

```
$AnimatedSprite2D.play("idle")
```

**Meaning:** If NOT moving, play "idle" animation

---

## Step 10: Test Your Animated Character! (2 min)

Press F5 to run!

You Should See:

- ✓ Character starts in **idle** animation
- ✓ Press **Right Arrow** → Character **walks** right with animation!
- ✓ Press **Left Arrow** → Character **walks** left (flipped + animated!)
- ✓ Release arrows → Character returns to **idle**
- ✓ Smooth transitions between animations

**Success!** Your character is fully animated! 🎉

---

## Troubleshooting

Problem	Solution
No animation playing	Check animation name spelling matches code
Only first frame shows	Check "Loop" is enabled in SpriteFrames
Animation too fast/slow	Adjust FPS in SpriteFrames editor
Character slides without animating	Check play("walk") is inside if direction != 0
Wrong animation plays	Check names: "idle" and "walk" (lowercase)
Sprite flips wrong direction	Change < 0 to > 0 in flip_h line
Error: Invalid call	Check node is AnimatedSprite2D not Sprite2D
Frames in wrong order	Reorder in SpriteFrames (drag and drop)
Sprite looks blurry	Go back: Filter must = Nearest, then Reimport

---

## Complete Code Reference

## **player.gd with animations:**

gdscript

extends CharacterBody2D

*# === MOVEMENT CONSTANTS ===*

const SPEED = 300.0

const JUMP\_VELOCITY = -400.0

*# === MAIN PHYSICS FUNCTION ===*

func \_physics\_process(delta):

*# --- APPLY GRAVITY ---*

if not is\_on\_floor():

velocity += get\_gravity() \* delta

*# --- JUMP CODE (DISABLED FOR LESSON 1-2) ---*

*# if Input.is\_action\_just\_pressed("ui\_accept") and is\_on\_floor():*

*# velocity.y = JUMP\_VELOCITY*

*# --- GET PLAYER INPUT ---*

var direction = Input.get\_axis("ui\_left", "ui\_right")

*# --- MOVE THE PLAYER ---*

if direction:

velocity.x = direction \* SPEED

else:

velocity.x = move\_toward(velocity.x, 0, SPEED)

*# --- ANIMATION CONTROL ---*

```

    if direction != 0:
        $AnimatedSprite2D.play("walk")
        $AnimatedSprite2D.flip_h = (direction < 0)
    else:
        $AnimatedSprite2D.play("idle")

# --- EXECUTE THE MOVEMENT ---
move_and_slide()

# === GOAL DETECTION SETUP ===
func _ready():
    var goal_box = get_parent().get_node("GoalBox")
    goal_box.body_entered.connect(_on_goal_box_body_entered)

# === GOAL COLLISION RESPONSE ===
func _on_goal_box_body_entered(body):
    if body == self:
        print("Level Complete! Great job!")
        get_tree().change_scene_to_file("res://level_2.tscn")

```

---

## Challenge Extensions

### Challenge 1: Add Breathing to Idle

Create 2-frame idle for subtle movement:

1. Photoshop: Create 2 idle frames (slight up/down)
2. Add both to "idle" animation
3. Set FPS to 4 (slow breathing)

---

### Challenge 2: Speed-Based Animation



Animation speed matches movement:

gdscript

*# In \_physics\_process, animation section:*

if direction != 0:

    \$AnimatedSprite2D.play("walk")

    \$AnimatedSprite2D.speed\_scale = abs(velocity.x) / SPEED

    \$AnimatedSprite2D.flip\_h = (direction < 0)

else:

    \$AnimatedSprite2D.play("idle")

    \$AnimatedSprite2D.speed\_scale = 1.0

---

### Challenge 3: Jump Animation (Preview)

Add jump frame:

1. Create "jump" animation (1 frame)
2. Add code:

gdscript

*# After gravity, BEFORE walk/idle:*

if not is\_on\_floor():

    \$AnimatedSprite2D.play("jump")

elif direction != 0:

    \$AnimatedSprite2D.play("walk")

    \$AnimatedSprite2D.flip\_h = (direction < 0)

else:

    \$AnimatedSprite2D.play("idle")

---

### Challenge 4: Running Animation

Separate "run" animation:

1. Make 4 frames with exaggerated motion

## 2. Play "run" when holding Shift:

gdscript

if direction != 0:

    if Input.is\_action\_pressed("ui\_shift"):

        \$AnimatedSprite2D.play("run")

    else:

        \$AnimatedSprite2D.play("walk")

    \$AnimatedSprite2D.flip\_h = (direction < 0)

else:

    \$AnimatedSprite2D.play("idle")

...

---

## 🎨 ***\*\*Animation Best Practices\*\****

### ***\*\*Frame Count Guidelines:\*\****

| Animation Type | Frame Count | FPS |

|-----|-----|-----|

| ***\*\*Idle\*\**** | 1-2 frames | 4-5 |

| ***\*\*Walk\*\**** | 3-4 frames | 8 |

| ***\*\*Run\*\**** | 4-6 frames | 12 |

| ***\*\*Jump\*\**** | 1-2 frames | 10 |

| ***\*\*Attack\*\**** | 3-5 frames | 10-15 |

### ***\*\*Smooth Animation Tips:\*\****

1. **\*\*Keep frame count consistent\*\*** (all walk frames same size)
2. **\*\*Use even timing\*\*** (each frame shown equal time)
3. **\*\*Loop seamlessly\*\*** (last frame connects to first)
4. **\*\*Exaggerate motion\*\*** (clear, obvious movement)
5. **\*\*Test at different speeds\*\*** (adjust FPS until right)

---

## ## 🎨 **\*\*Scene Structure\*\***

**\*\*Your Player should look like:\*\***

`, ` ,`

Player (CharacterBody2D) [ 📄 SCRIPT]

```
└─ AnimatedSprite2D
  │   └─ SpriteFrames
  │       └─ idle (1 frame, 5 FPS)
  │       └─ walk (3-4 frames, 8 FPS)
  └─ CollisionShape2D
```

`, ` ,`

---

## ## ✅ **\*\*Final Success Checklist\*\***

**\*\*Photoshop:\*\***

`, ` ,`

☐ Walk cycle created (3-4 frames)

☐ Idle pose created

- ☐ All frames in one sprite sheet
- ☐ Backgrounds removed (transparent)
- ☐ PNG exported
- ☐ PSD saved

...

#### **\*\*Godot:\*\***

...

- ☐ New sprite copied to project
- ☐ Import settings: Compress = Lossless
- ☐ Inspector settings: Filter = Nearest
- ☐ Reimport clicked
- ☐ Sprite2D replaced with AnimatedSprite2D
- ☐ SpriteFrames resource created
- ☐ "idle" animation created
- ☐ "walk" animation created
- ☐ Animation code added to script
- ☐ CollisionShape2D correct size

...

#### **\*\*Testing:\*\***

...

- ☐ Game runs (F5)
- ☐ Character starts idle
- ☐ Walk animation plays when moving
- ☐ Character flips to face direction
- ☐ Returns to idle when stopped
- ☐ Smooth transitions

---

## ☀️ What You Learned Today

### Photoshop Skills:

- Creating animation sequences
- Understanding walk cycles
- Frame-by-frame animation
- Organizing animation frames

### Godot Skills:

- AnimatedSprite2D node
- SpriteFrames resource
- Animation management
- Playing animations from code
- Conditional animation logic

### Game Development Concepts:

- Animation loops
- FPS (frames per second)
- State-based animation
- Visual polish and "game feel"

---

## 🧙 Next Lesson Preview

### Lesson 3: Creating UI Icons (Hearts, Coins, Buttons)

We'll create:

- Health heart icon (full, half, empty states)
- Collectible coin icon with shine
- Button designs (normal, hover, pressed)
- Implementing UI in Godot

**Think about:** What games have clear, easy-to-read UI? What makes good icon design?

---

## Reflection Questions

Answer in your exercise book:

1. Why do walk cycles need at least 2 different poses?
  2. What FPS worked best for your walk animation? Why?
  3. How does animation make a game feel more "alive"?
  4. What was the hardest part of creating your walk cycle?
  5. Name a game with excellent character animation. What makes it good?
- 

## Homework (Optional)

1. Create a 6-frame walk cycle (smoother!)
  2. Add a "crouch" or "duck" animation
  3. Design an enemy with simple walk animation
  4. Find and study 3 professional game walk cycles
  5. Sketch ideas for attack or jump animations
- 

 **Congratulations! Your character now moves with fluid, professional-looking animation!**