For Game Development students – Buu Nguyen – v0.2

TexturePacker User Guide

Table of Contents

Introduction

This guide is for anyone who needs some help on using TexturePacker. In this guide I will show you how to export animation from Photoshop and put them into TexturePacker step by step.

I would like to thank Raven Mortimer, a wonderful artist and project partner who inspired me to write this guide. I also wanted to thank Matthew Doucette, Sean Morrow and Sam Robichaud – my instructor at NSCC Truro – for their advices and mentorship while I was writing this guide.

If you, readers, would like to give me any feedback on this guide, feel free to contact me at

Buu.nguyenquoc@gmail. You can also look for the newest updates of this guide on https://www.seromyr.com/download-archive

Getting Started

1. Download and install Adobe Photoshop

Get it at https://www.adobe.com/ca

Please note that the Render Video function mentioned in this guide will not working properly on Photoshop CC version 20.0.0 due to software's error. This issue has been fixed on version 20.0.1. It is my recommendation to use the newest version.

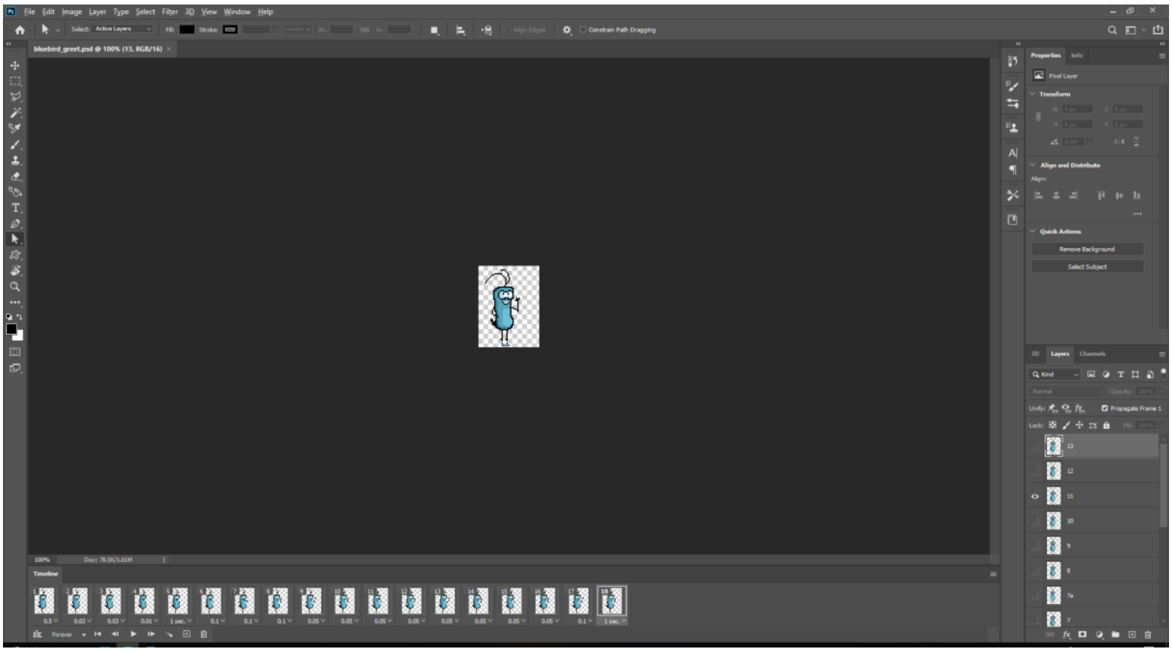
In this guide, I used Photoshop version 21.1.2.

2. Download and install TexturePacker

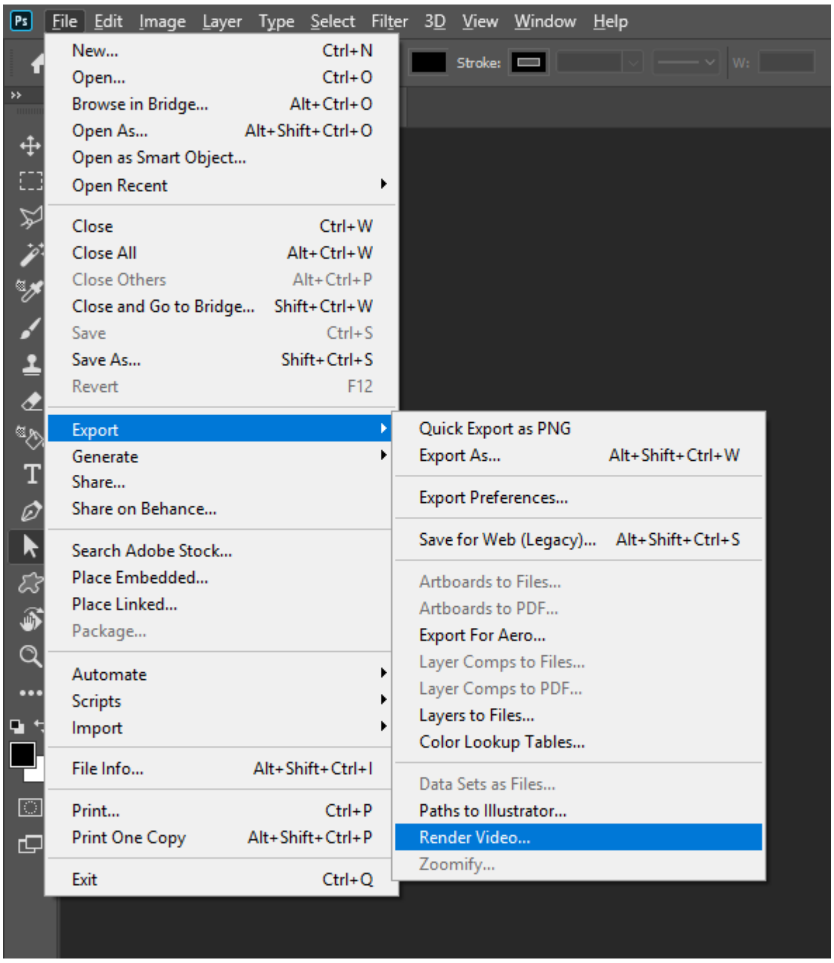
Get it at https://www.codeandweb.com/texturepacker

Export Animation from Photoshop

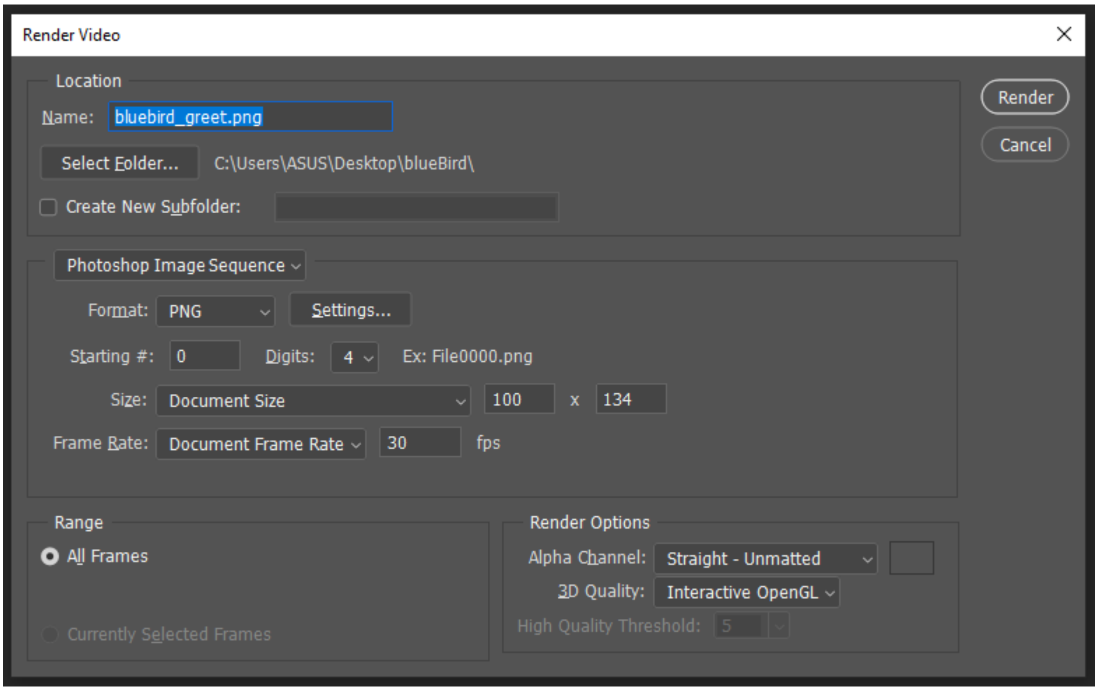
1. Open the animation file in Photoshop



2. Go to **File** > **Export** > **Render Video**



3. A Render Video window will pop up



Render Video

Location

Name: bluebird_greet.png

Select Folder... C:\Users\ASUS\Desktop\blueBird\

□ Create New Subfolder:

Photoshop Image Sequence

Format: PNG

Settings...

Starting #: 0 Digits: 4 Ex: File0000.png

Size: Document Size 100 x 134

Frame Rate: Document Frame Rate 30 fps

Range

● All Frames

○ Currently Selected Frames

Render Options

Alpha Channel: Straight - Unmatted

3D Quality: Interactive OpenGL

High Quality Threshold: 5

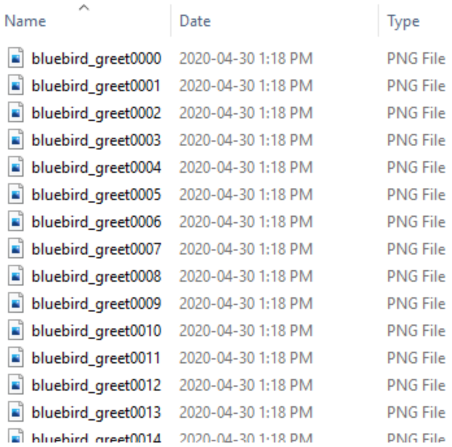
From top to bottom:

Name

- The output name suffix of the exporting files if your animation had more than 1 frame.

- Look at the name I had in the example: *bluebird_greet.png*
  “*bluebird*” is the name of the character.
  “*greet*” is the action of the current animation. In our project, it is likely that one character would have multiple actions.

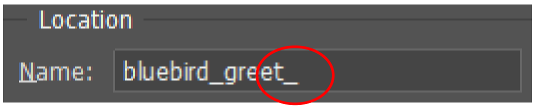
- If I left as it is, Photoshop will export it to multiple files as shown in this image below:



Name	Date	Type
bluebird_greet0000	2020-04-30 1:18 PM	PNG File
bluebird_greet0001	2020-04-30 1:18 PM	PNG File
bluebird_greet0002	2020-04-30 1:18 PM	PNG File
bluebird_greet0003	2020-04-30 1:18 PM	PNG File
bluebird_greet0004	2020-04-30 1:18 PM	PNG File
bluebird_greet0005	2020-04-30 1:18 PM	PNG File
bluebird_greet0006	2020-04-30 1:18 PM	PNG File
bluebird_greet0007	2020-04-30 1:18 PM	PNG File
bluebird_greet0008	2020-04-30 1:18 PM	PNG File
bluebird_greet0009	2020-04-30 1:18 PM	PNG File
bluebird_greet0010	2020-04-30 1:18 PM	PNG File
bluebird_greet0011	2020-04-30 1:18 PM	PNG File
bluebird_greet0012	2020-04-30 1:18 PM	PNG File
bluebird_greet0013	2020-04-30 1:18 PM	PNG File
bluebird_greet0014	2020-04-30 1:18 PM	PNG File

You can see that it put 4-digit number right next to the name. That is okay but not quite well organized.

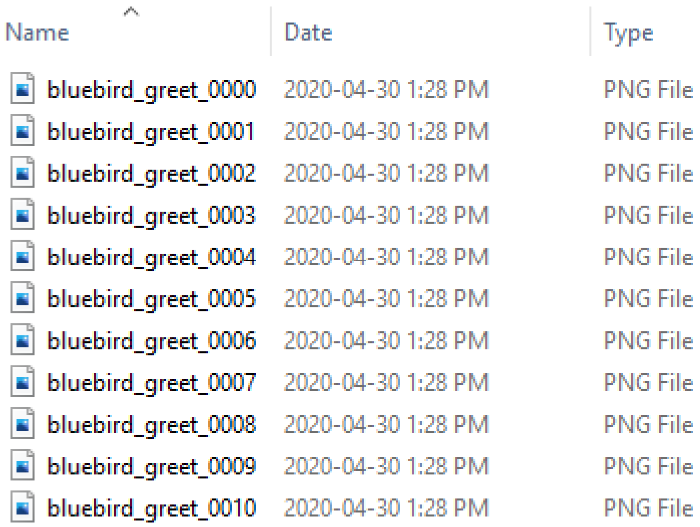
I suggest you put an extra underscore next to the file name which should look like this:



Location

Name: bluebird_greet_

It would separate the name and the number, which is very easy to read and keep track.

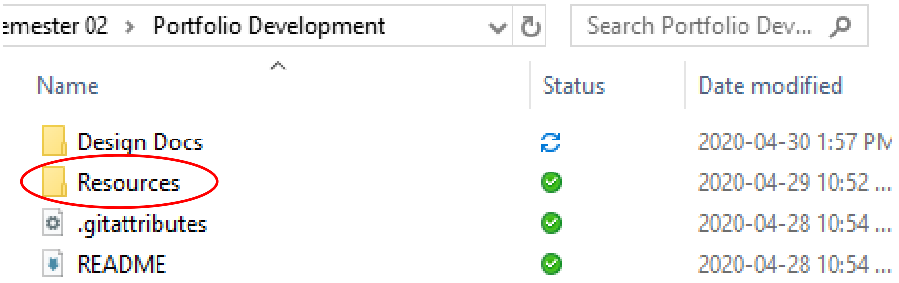


Name	Date	Type
bluebird_greet_0000	2020-04-30 1:28 PM	PNG File
bluebird_greet_0001	2020-04-30 1:28 PM	PNG File
bluebird_greet_0002	2020-04-30 1:28 PM	PNG File
bluebird_greet_0003	2020-04-30 1:28 PM	PNG File
bluebird_greet_0004	2020-04-30 1:28 PM	PNG File
bluebird_greet_0005	2020-04-30 1:28 PM	PNG File
bluebird_greet_0006	2020-04-30 1:28 PM	PNG File
bluebird_greet_0007	2020-04-30 1:28 PM	PNG File
bluebird_greet_0008	2020-04-30 1:28 PM	PNG File
bluebird_greet_0009	2020-04-30 1:28 PM	PNG File
bluebird_greet_0010	2020-04-30 1:28 PM	PNG File

This looks better, don't you agree?

Select Folder

- This button allows you to change the location for exported files.
- I recommended you put all your exported animation in Resources folder, which where programmer would look for data implementation.

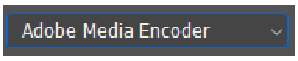


Name	Status	Date modified
Design Docs	⟳	2020-04-30 1:57 PM
Resources	✓	2020-04-29 10:52 ...
.gitattributes	✓	2020-04-28 10:54 ...
README	✓	2020-04-28 10:54 ...

Create New Subfolder checkbox

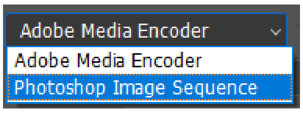
- This button allows you to create a new subfolder and put all the exported files inside it.
- I recommend you create a subfolder for every animation of the same character. It is a tidier way to organize and would not accidentally override your other animation data during the process.
- Just simply check the checkbox and type in the name of the folder of your choice.

Encoder

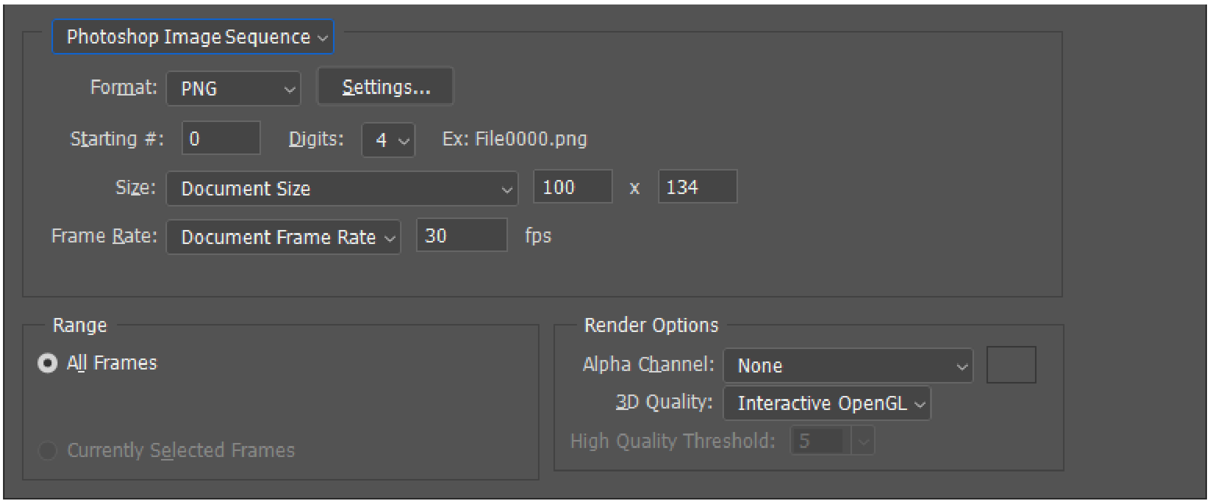


Adobe Media Encoder

- Photoshop probably uses this encoding method by default the first time you opened it. This is not what we are going to use.
- Click on it and select Photoshop Image Sequence.



- You will see new settings for this encoding method. We will make some changes here.



Photoshop Image Sequence

Format: PNG   Settings...

Starting #: 0   Digits: 4   Ex: File0000.png

Size: Document Size   100 x 134

Frame Rate: Document Frame Rate   30 fps

Range

● All Frames

○ Currently Selected Frames

Render Options

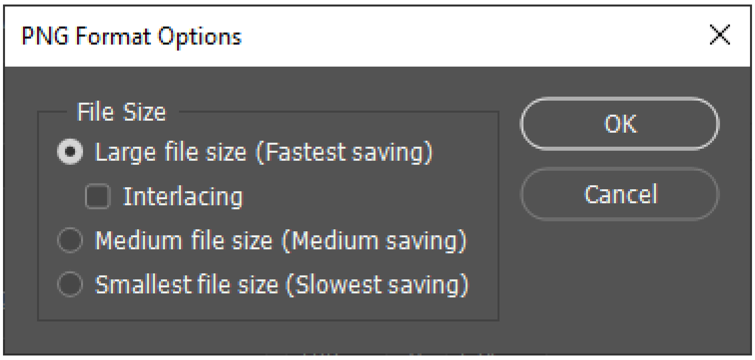
Alpha Channel: None

3D Quality: Interactive OpenGL

High Quality Threshold: 5

File format

- Select PNG
- In the Settings button next to it, choose Large file size to keep the best image quality.



PNG Format Options

File Size

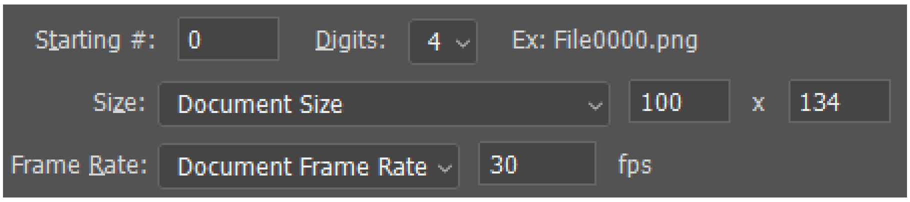
● Large file size (Fastest saving)

□ Interlacing

○ Medium file size (Medium saving)

○ Smallest file size (Slowest saving)

Other formatting options



Starting #: 0 Digits: 4 Ex: File0000.png

Size: Document Size 100 x 134

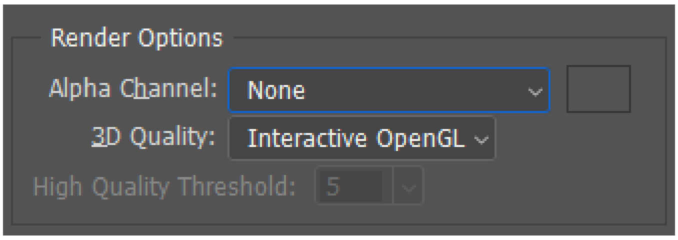
Frame Rate: Document Frame Rate 30 fps

- Starting #: This affect the starting number in the naming prefix of the exported files. Just leave it be.
- Digits: As you have seen in the first demonstration above, the exported files have 4-digit number. If you were certain that your animation would never exceed 100 frames, you could change it down to 2-digit. I personally prefer 3-digit. It is your choice.
- Size: Just leave it be, it will use the dimensions of the document canvas.
- Frame Rate: Just leave it be, our game and most of other games out there are using 30 fps.

Range

- Do not touch it.

Render Options



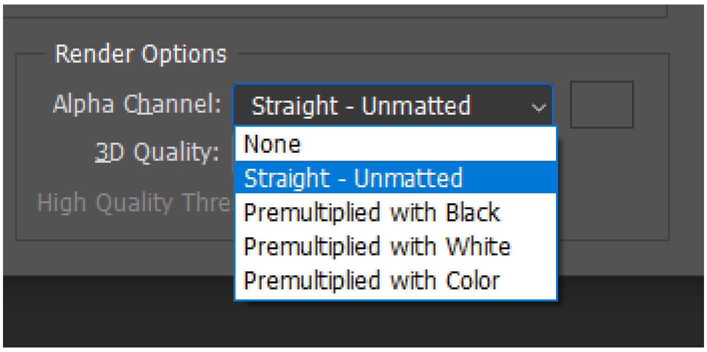
Render Options

Alpha Channel: None

3D Quality: Interactive OpenGL

High Quality Threshold: 5

- Alpha Chanel: Default is None, change it to Straight – Unmatted. It would allow exported images have transparent background.



Render Options

Alpha Channel: Straight - Unmatted

3D Quality:

High Quality Thre

- 3D Quality: Just leave it be.

Congratulations! You can hit Render and switch to TexturePacker now.

4. Summary

A few things to remember before hitting **Render** button.

1. Place an extra underscore at the end of the file name (if there was not any).
2. Export animation into Resources folder of the common working folders.
3. Use separate folder for each animation.
4. Make sure encoding method is Photoshop Image Sequence.
5. Format type is PNG with Large File Size for highest image quality.
6. Make sure the Alpha Chanel uses Straight – Unmatted for transparent background.

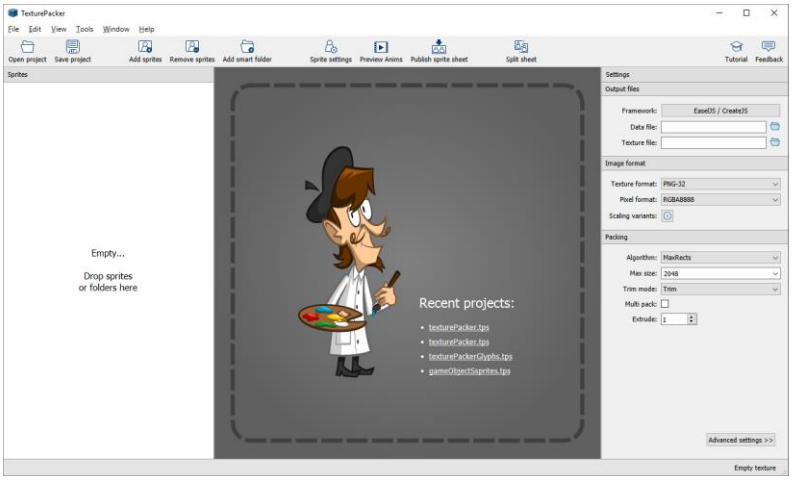
Working with TexturePacker

1. Creating a static asset

Static assets are sprites that has no animation, like background images, buttons, banners, splash arts.

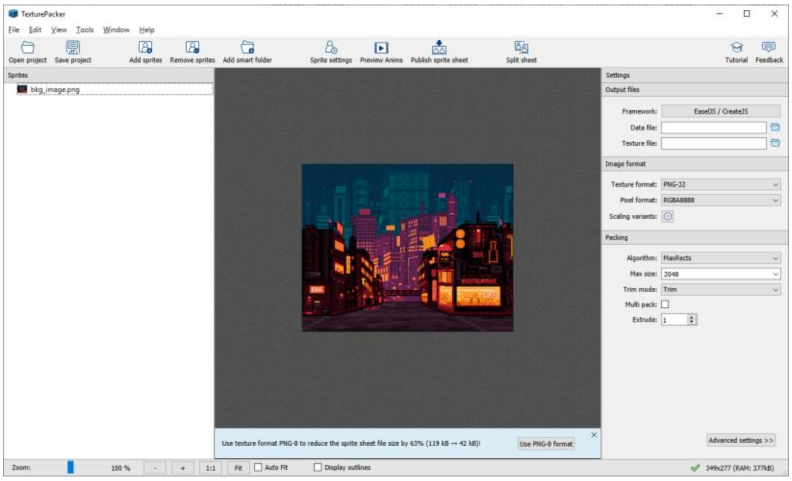
In this example, I'm going to use a splash image taken from here. Thank you Ansimuz for this amazing piece of art.

- Open TexturePacker.



Framework: EaselJS / CreateJS
Data file:
Texture file:
Texture format: PNG-32
Pixel format: RGBA8888
Scaling variants:
Algorithm: MaxRects
Max size: 2048
Trim mode: Trim
Multi pack: □
Extrude: 1

- Drag the image or image folder into TexturePacker.



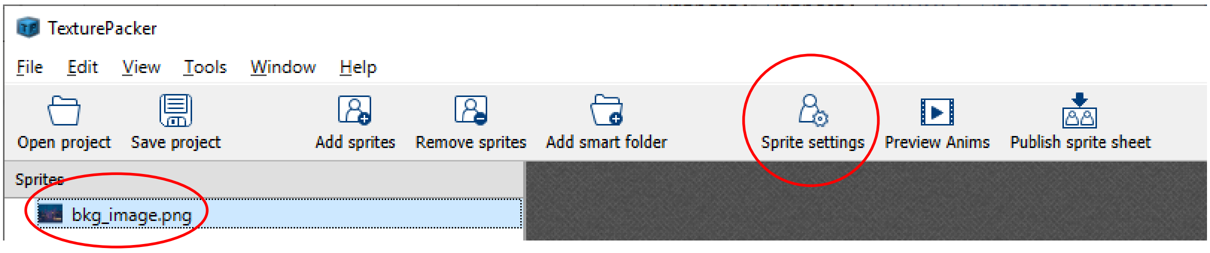
Framework: EaselJS / CreateJS
Data file:
Texture file:
Texture format: PNG-32
Pixel format: RGBA8888
Scaling variants:
Algorithm: MaxRects
Max size: 2048
Trim mode: Trim
Multi pack: □
Extrude: 1

Use texture format PNG-8 to reduce the sprite sheet file size by 63% (119 kB → 42 kB)!

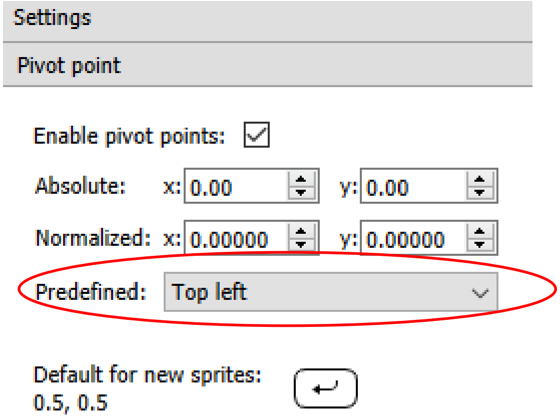
□ Auto Fit □ Display outlines

349x277 (RAM: 377kB)

- Select the image and click on Sprite Settings to change it Pivot Point (very important).



- By default, the pivot is at the center of the sprite. Change it to Top Left.



Settings

Pivot point

Enable pivot points: ☑

Absolute: x: 0.00 y: 0.00

Normalized: x: 0.00000 y: 0.00000

Predefined: Top left

Default for new sprites: 0.5, 0.5

All the static assets' pivot should be Top Left, unless stated otherwise by programmer.

- Hit Back to sheet to make changes to other settings

✕ Back to sheet

Now Look at the Settings Panel on the right side

- Change Framework to EaselJS / CreateJS (or anything else depend on your project).
- Set the output data file path and file name. I would recommend you put it inside the Resources folder. Save as type JSON file. TexturePacker would also create a new image file for this JSON file.
- Other settings should look like this image below

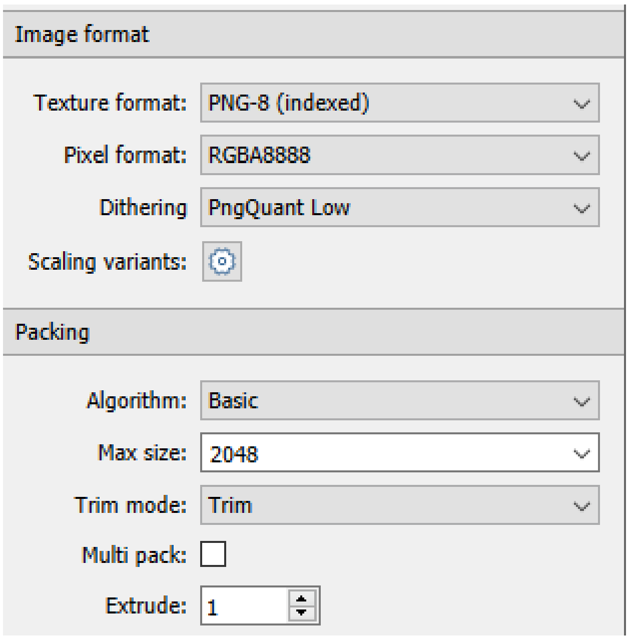


Image format

Texture format: PNG-8 (indexed)

Pixel format: RGBA8888

Dithering PngQuant Low

Scaling variants:

Packing

Algorithm: Basic

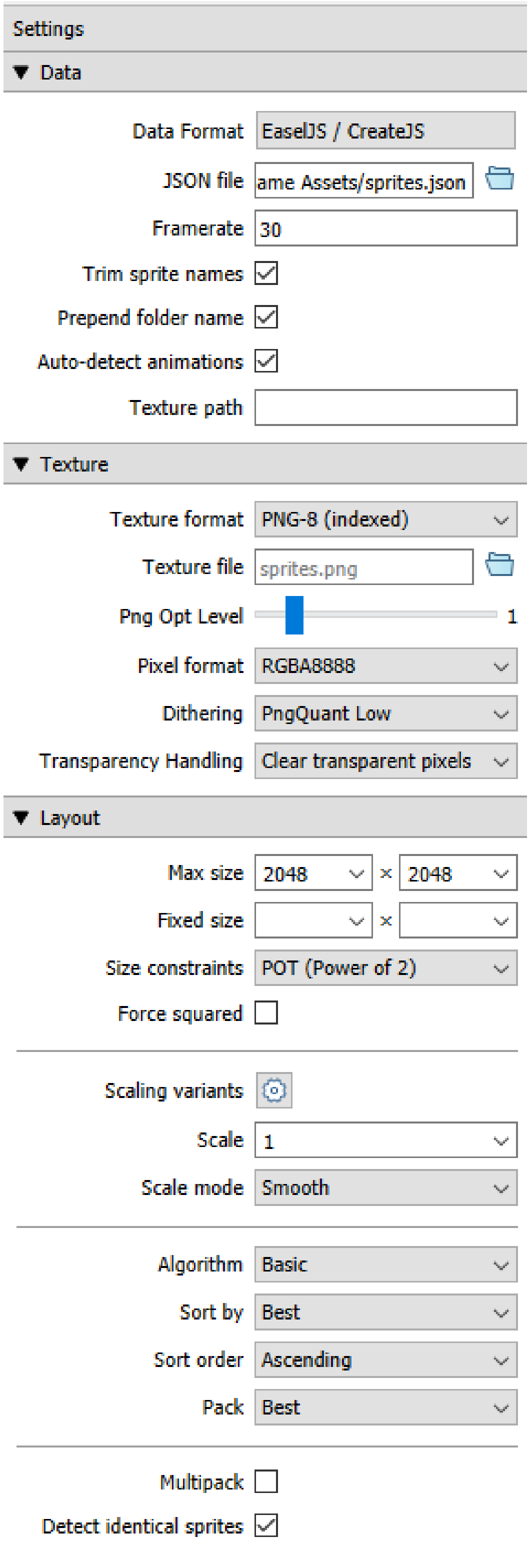
Max size: 2048

Trim mode: Trim

Multi pack: □

Extrude: 1

- Advanced settings should like these images below



Settings

▼ Data

Data Format: EaselJS / CreateJS

JSON file: ame Assets/sprites.json

Framerate: 30

Trim sprite names ☑

Prepend folder name ☑

Auto-detect animations ☑

Texture path:

▼ Texture

Texture format: PNG-8 (indexed)

Texture file: sprites.png

Png Opt Level: 1

Pixel format: RGBA8888

Dithering: PngQuant Low

Transparency Handling: Clear transparent pixels

▼ Layout

Max size: 2048 × 2048

Fixed size: ×

Size constraints: POT (Power of 2)

Force squared □

Scaling variants

Scale: 1

Scale mode: Smooth

Algorithm: Basic

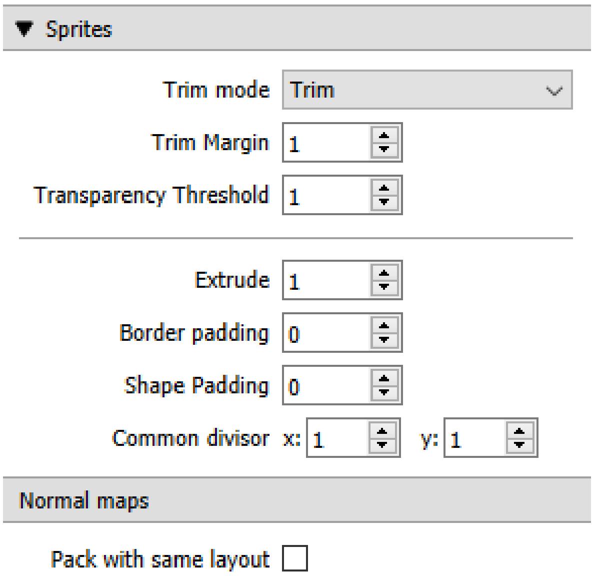
Sort by: Best

Sort order: Ascending

Pack: Best

Multipack □

Detect identical sprites ☑



▼ Sprites

Trim mode: Trim

Trim Margin: 1

Transparency Threshold: 1

Extrude: 1

Border padding: 0

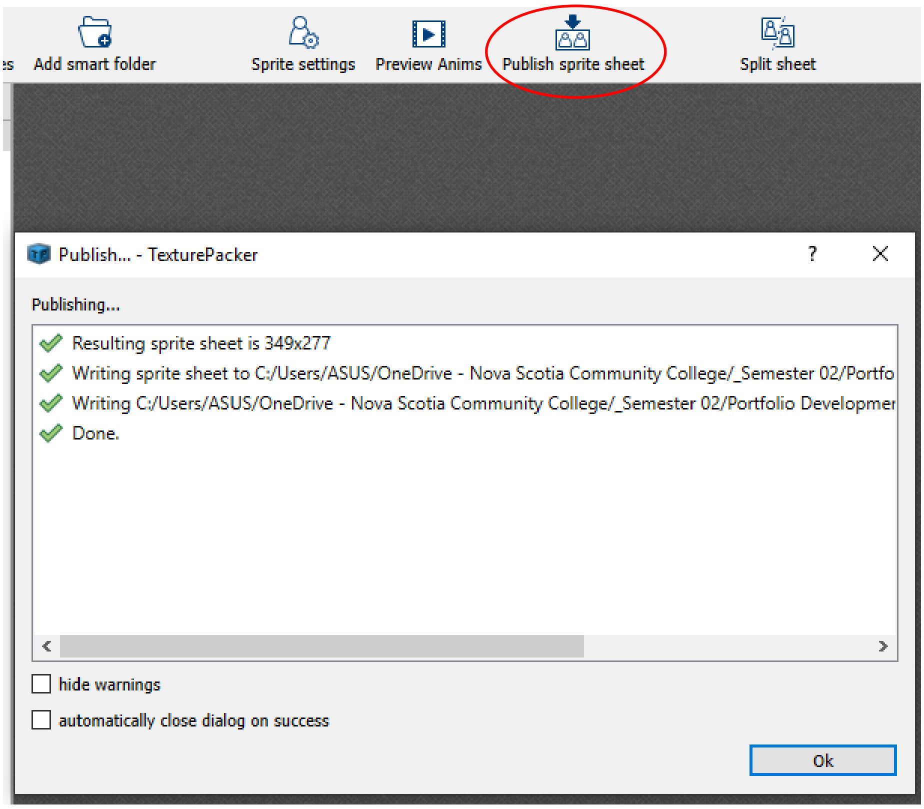
Shape Padding: 0

Common divisor x: 1 y: 1

Normal maps

Pack with same layout □

- When everything is ready press the Publish Sprite Sheet button, a popup would appear and notify you that the export process has finished. It is time to send the exported JSON and PNG to your programmer or commit to GitHub.



Publishing...

```
✓ Resulting sprite sheet is 349x277
✓ Writing sprite sheet to C:/Users/ASUS/OneDrive - Nova Scotia Community College/_Semester 02/Portfo
✓ Writing C:/Users/ASUS/OneDrive - Nova Scotia Community College/_Semester 02/Portfolio Developmer
✓ Done.
```

□ hide warnings

□ automatically close dialog on success

- Do not forget to save your TexturePacker project.

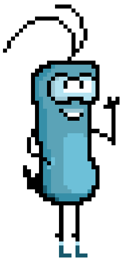
2. Creating dynamic assets

Dynamic assets are sprites that has at least one animation, like a game character with walking, jumping animation.

Casually, you could put all sprites of all artworks into one big JSON file. However, I would recommend you put them in separate JSON files. This helps a lot in preventing potential bugs that could slow your project down.

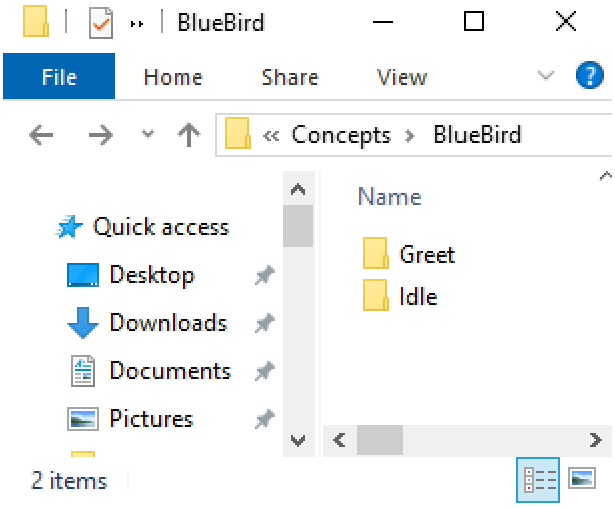
The process would be the same as creating static asset. In this section, I will demonstrate how to create a character sprite that has 2 different animations.

In this example, I am going to use my BlueBird character.

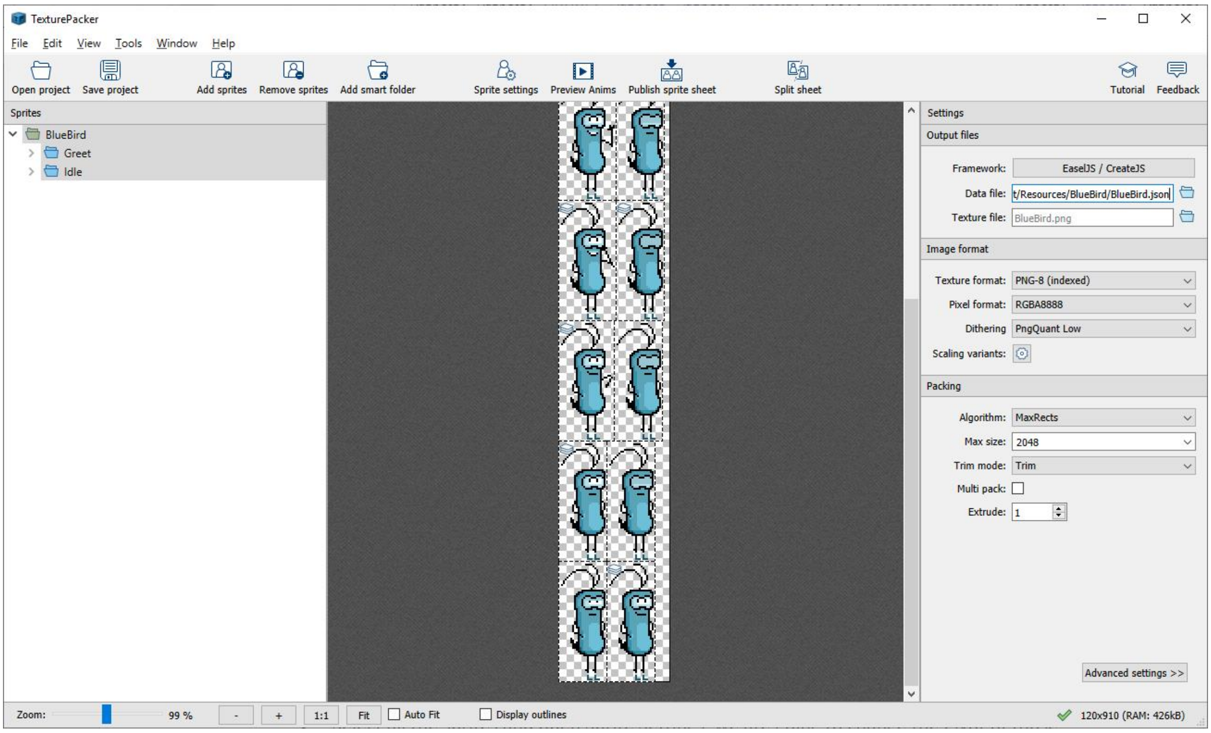


On a second thought, he does not really look like a bird. Welp! ¯_(ツ)_/¯

- I have two folders contain two different animation files exported from Photoshop using Render Video function.



- Open TexturePacker and drag the parent BlueBird folder that holds the animations into it.



Settings

Output files

Framework: EaselJS / CreateJS

Data file: t/Resources/BlueBird/BlueBird.json

Texture file: BlueBird.png

Image format

Texture format: PNG-8 (indexed)

Pixel format: RGBA8888

Dithering: PngQuant Low

Scaling variants:

Packing

Algorithm: MaxRects

Max size: 2048

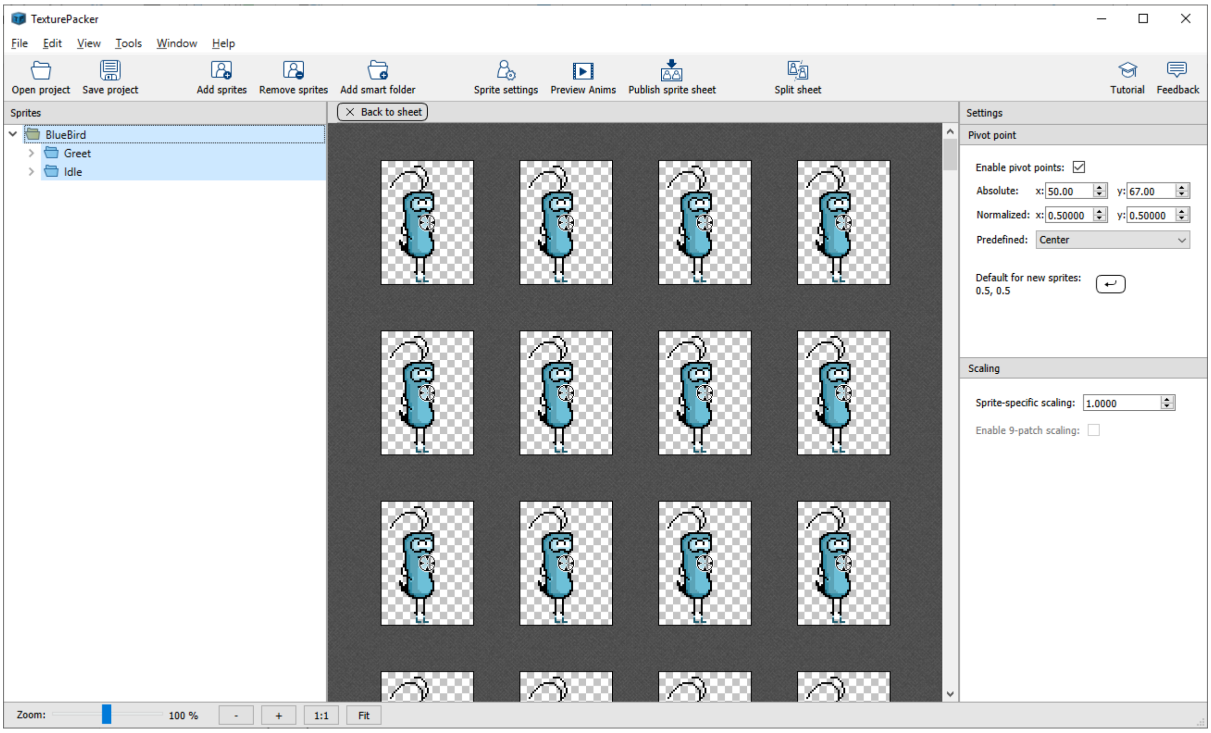
Trim mode: Trim

Multi pack: □

Extrude: 1

✓ 120x910 (RAM: 426kB)

- Specify a new name and path for this sprite.
- Use the same settings described in Creating static assets section.
- Select all the sprites and open Sprite Settings, we are going to change the Pivot of those sprites a little bit different this time.



Settings

Pivot point

Enable pivot points: ☑

Absolute: x: 50.00 y: 67.00

Normalized: x: 0.50000 y: 0.50000

Predefined: Center

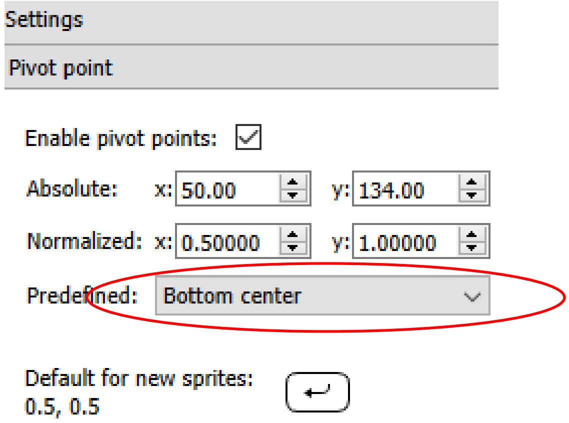
Default for new sprites: 0.5, 0.5

Scaling

Sprite-specific scaling: 1.0000

Enable 9-patch scaling: □

- Use Bottom Center for the Pivot Point



Settings

Pivot point

Enable pivot points: ☑

Absolute: x: 50.00 y: 134.00

Normalized: x: 0.50000 y: 1.00000

Predefined: Bottom center

Default for new sprites: 0.5, 0.5

Explanation: This is an animated character that can be controlled by player. The feet are the most frequent things that collides with other objects during gameplay. Putting the pivot at the feet helps programmer a lot in coding. We could manually do it ourselves, but with the acknowledgement and help from artists would benefit all of us and make the work goes faster overall.

If you were unclear where you should set the pivot point of a certain sprite, just simply ask the programmer. If they are not available now and you need to finish your task anyways, set it to Top Left.

- You could hit Publish Sprite Sheet at this point.
- Do not forget to save your TexturePacker project file. You have been warned twice.

3. Summary

Things to remember before **publishing the sprite sheet**

1. Make sure all settings and pivot points are set correctly.
2. Uses different JSON file for each character, do not make an All-In-One file.
3. If anything is unclear, give your programmer a nudge.
4. Save you TexturePacker project.

Creating Glyph assets aka. Graphic fonts for games (writing in progress)