

CheapBirds

White Games

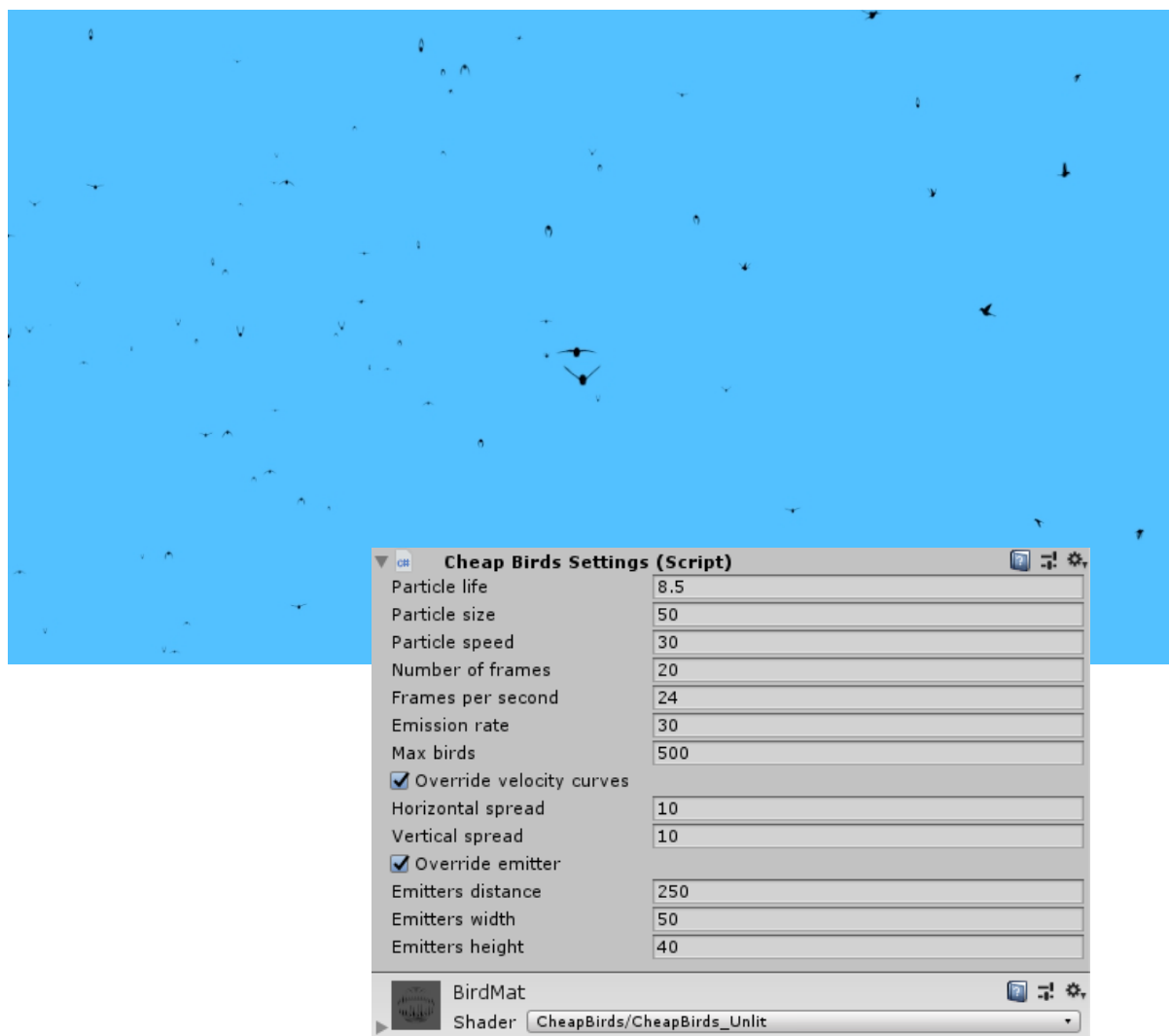


Table of Contents

Table of Contents.....	2
Overview.....	3
Release Notes.....	4
How to use.....	5
Adding Custom Spritesheets.....	6
Under the hood.....	6
Automatic Spritesheet creation.....	6
Custom Spritesheet creation.....	7
HDRP/URP.....	9
Known Issues/Troubleshooting.....	10

Overview

CheapBirds is intended to provide a computationally-cheap animated birds effect. The effect is achieved via a particle system and an animation spritesheet, together with a custom particle mesh and shader which allow the particles to be viewed from different angles and giving a 3d effect.

Also provided is a custom emission mesh which allows particles to be fired in 2 opposing directions, intended to allow the birds to fly in different directions. Of course, you can also define your own emission shapes by editing the Particle System as per normal.

Release Notes

V1.1

- Added 6-sided mesh with lit material
- Added tool to generate spritesheets
- Added fish model/spritesheet and spritesheet generation example
- Added HDRP/URP support from Unity 2020.1
- Undo/Redo now supported in custom Inspector

V1.0

- Initial release

How to use

1. Insert the CheapBirds prefab into your scene. The prefab consists of a particle system and a script which makes it easier to make common changes.
2. In the Scene window you will be able to preview the effect of the CheapBirds particle system. Place the prefab in the desired location.
3. If using the 2-way emission mesh, set the emitter distance, width, and height to the desired values.
4. Adjust the “Particle Life” value as needed
5. If required, adjust the remaining values (described below)

Particle Life	Lifetime of particles
Particle Size	Size of particles
Particle Speed	Speed of particles
Number of Frames	Number of frames in the spritesheet. You may need to alter this if you use a custom spritesheet.
Frames per Second	Alters the speed of the animation. Adjust to your preference.
Emission rate	How many new particles are created per second
Max birds	The maximum number of particles
Override velocity curves	Whether to use the below “spread” values to alter the Particle System's “velocity over lifetime” curves. Disable this to set your own velocity values.
Horizontal spread	How far the particles horizontally deviate from a straight path
Vertical spread	How far the particles vertically deviate from a straight path
Override emitter	Whether to use the bidirectional custom emitter mesh (causes birds to fly in opposing directions)
Emitters distance	The distance between the opposing emitters, in metres
Emitters width	The width of the opposing emitters
Emitters height	The height of the opposing emitters

You can also edit any other values directly in the Particle System. If you change one of the values that CheapBirds uses, CheapBirds will update its setting the next time the GameObject is selected.

If you change the value type in the Particle System (for example, changing the “Start Lifetime” setting in the particle system from “Constant” to “Between two curves”) then CheapBirds' “Particle Life” value will no longer have an effect until it is reset to “Constant” mode.

Adding Custom Spritesheets

Under the hood

The 3-sided Particle mesh has its UV coordinates laid out to expect the spritesheet texture to have 3 rows in the order:

1. Front/back view
2. Side view, facing left
3. Top view, facing down

The 6-sided mesh works in a similar way.



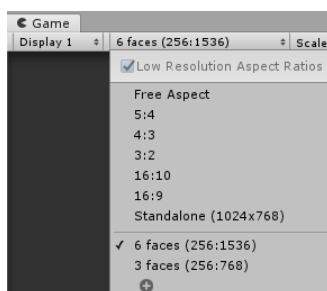
(The birds in our included spritesheet are white with a transparent background, shown here as black for illustrative purposes. The birds could be any colour, but white allows us to apply colour tints later in the material properties.)

Automatic Spritesheet creation

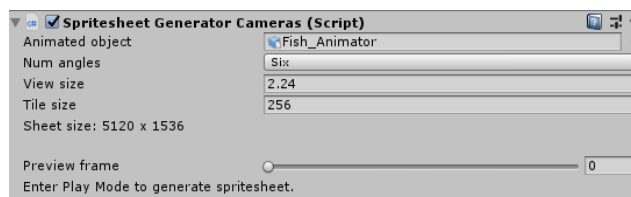
In v1.1, a tool was added to generate spritesheets based off a looping animation inside Unity. In the *CheapBirds\Scenes* folder there is a scene called *SpritesheetGeneration*. An example with an animated fish model is provided.

1. Set the Game view's resolution

The Game View in the *SpritesheetGeneration* scene should have 2 options for the desired spritesheet type (3 or 6-sided). These modes ensure no stretching occurs of the resulting renders and you get a better idea of what the final spritesheet will look like. If these options aren't present, you should create them (select "Fixed Resolution" instead of "Aspect Ratio").



The object called *SceneCameras* has a *SpritesheetGeneratorCameras* component that does the work:



Drag your animated object into the AnimatedObject slot. This object must have either an Animator or Animation component with the desired animation loop. Animation components are slightly easier to set up as you don't need to create an Animator Controller, but may not be suitable for your model as they rely on the Legacy animation system.

2. Select the **number of angles** you want your cameras to render – three or six.

Three-angle spritesheets are suitable for silhouette particles, as the same texture is shown on both sides (left/right, top/bottom, front/back). After rendering a three-sided spreadsheet it is recommended to take the spritesheet into an image editing tool such as Photoshop to ensure the silhouettes in the spritesheets are solid white in colour (and possibly with an outline for effect) – the colour tint of the particles can be edited later in the material.

Six-sided particles will look better when viewed in more detail than a silhouette, as the textures will be different on each side of the particle instead of for example the front/back sharing the same texture which may not look correct.

3. Adjust the **View size**

Adjusting this value adjusts the view size of the 6 cameras used for Rendering. You should look at the Game view while adjusting this value and ensure that the animated model fits within the rendered images. Use the **Preview Frame** slider to check the model stays within view on all frames.

4. Set the **Tile size**

Adjust the tile size as needed. The **Sheet size** readout tells you how big the resulting spritesheet will be.

5. Generate the spritesheet

Enter Play mode and click **Generate Spritesheet**.

If the **Preview Frame** slider does not change the animation time or the generated spritesheet's frames are all the same, you may need to check your animated object is set up correctly. Use the Fish example as a reference.

Custom Spritesheet creation

Each of our sprites in the sheet is created by rendering a frame of a 3d model's animation. We recommend you do the same, though there is nothing stopping you from creating your own sprites by hand in 2D.

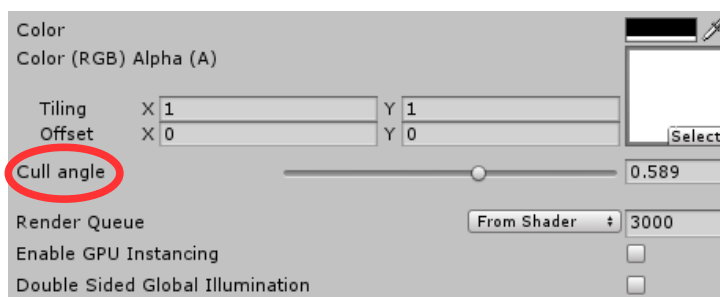
Our recommended workflow:

1. Open your animated 3d model in your favourite animation / rendering package
2. Set up 3 or 6 identical cameras at the same distance from the 3d model. Adjust the camera settings so that the model will just about fit inside the rendered image for every frame of the animation. Orthographic cameras may work best.

3. Render the whole animation with each camera, outputting each frame to an image file. If your application allows it, you may be able to render all 3 cameras into the same image, thereby creating an entire column of the spritesheet, or even the whole spritesheet at once. Make sure your rendered image includes a transparent background (alpha channel).
4. Use an image-editing application to stitch the rendered images together to create the spritesheet.
5. Duplicate one of the the included BirdMat (3 sided) or FishMat (6 sided) materials, name it accordingly, and insert your spritesheet texture into the new material.
6. Insert one of the CheapBirds prefabs into your scene and assign your material to it.
7. In the CheapBirds settings, set the number of frames in your spritesheet.
8. Adjust the CheapBirds' Frames per second value as needed
9. Using the scene view, preview the particle effect, pause it, and zoom in on a single particle.



10. The CheapBirds shader includes a slider to control the angle at which the planes of the particle mesh are visible. This prevents us from seeing all views of the spritesheet simultaneously. Adjust the value to give the best result, while checking the particle from all angles.



11. Rename your edited CheapBirds object and save it as a new prefab.

HDRP/URP

HDRP and URP are supported from 2020.1. To install, unpack the relevant .unitypackage from the SRP folder. ShaderGraph is required.

The .unitypackage updates the custom shader and materials to use ShaderGraph in order to be compatible with the render pipeline.

Known Issues/Troubleshooting

If the **Preview Frame** slider does not change the animation time and/or the generated spritesheet's frames are all the same, you may need to check your animated object is set up correctly. Make sure the "Preview frame" slider works and your model is animated in Play mode. If you use the Animator component, make sure your animation is the only clip in the Controller and is the default clip. Use the Fish example as a reference.

If you have any problems, please contact WhiteGamesInfo@gmail.com