Generating 3D images with PHP

Dmitry Matiouchenko

BCIT COMP4515 Self-Selected Topic

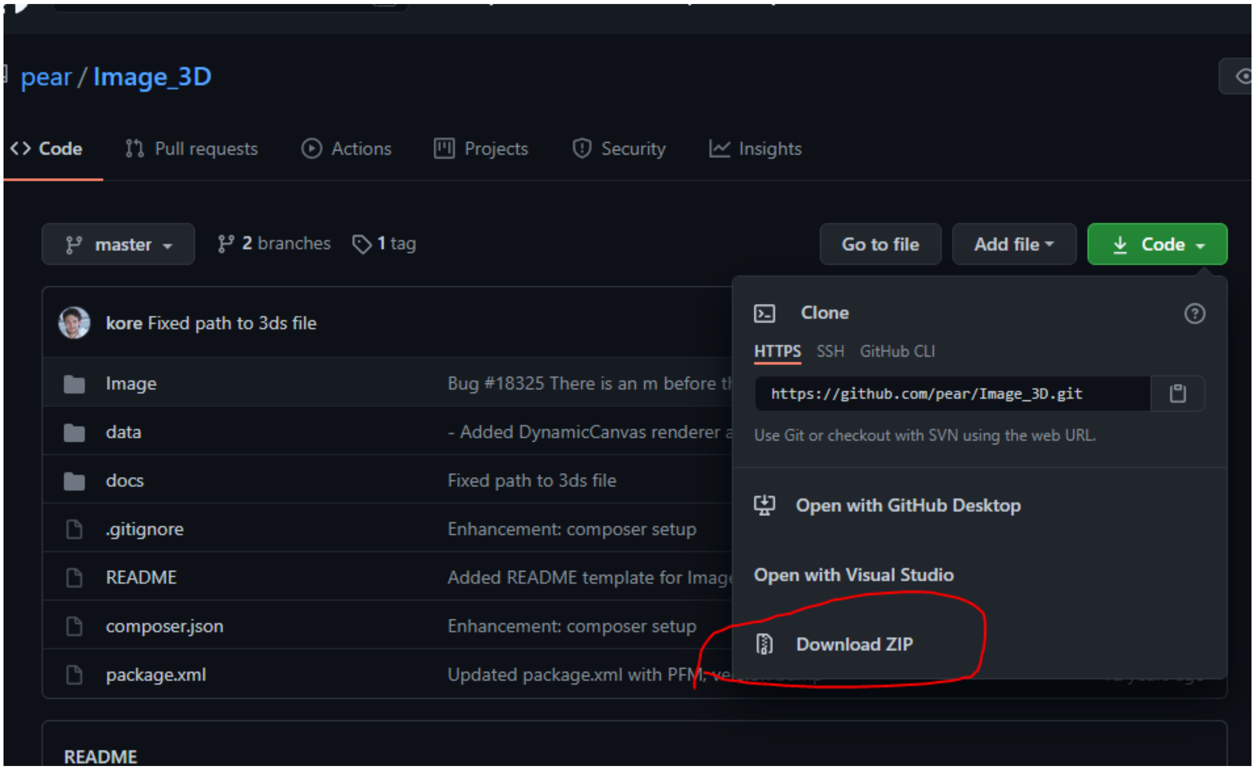
We can use PHP to generate/render 3D images, import external 3D models, and printing 3D lookalike images in ascii characters on cli. Some of them will be demod in class the rest i recommend exploring here:
https://hotexamples.com/examples/-/Image_3D/-/php-image_3d-class-examples.html
https://pear.php.net/package/Image_3D/docs/0.4.2/Image_3D/Image_3D_Color.html

To do this we will use Image_3D library which is unfortunately no longer maintained and was last updated for PHP 5 but will work with later versions.

NOTE: I personally wasn't able to find any substitute libraries and neither Pear nor Composer worked for me so we will download the github repo manually.

Let's get started!

1) First we'll create a folder for the project.
2) Next we'll go to https://github.com/pear/Image_3D and download as a zipped folder.



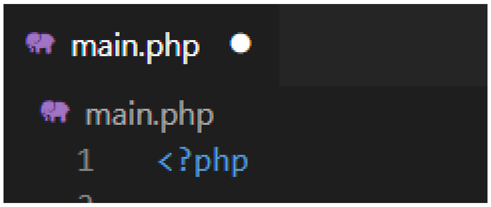
kore Fixed path to 3ds file

Image	Bug #18325 There is an m before th
data	- Added DynamicCanvas renderer a
docs	Fixed path to 3ds file
.gitignore	Enhancement: composer setup
README	Added README template for Imag
composer.json	Enhancement: composer setup
package.xml	Updated package.xml with PFM; ve

```
https://github.com/pear/Image_3D.git
```

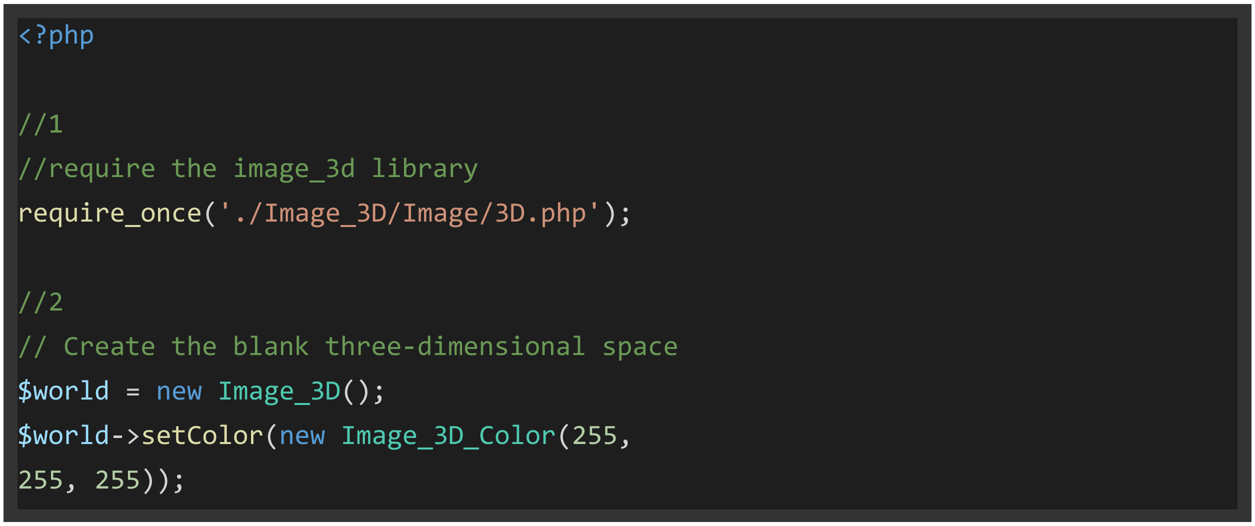
Use Git or checkout with SVN using the web URL.

3) Let's extract to the project folder and rename the new folder to "Image_3D" or simpler.
4) Next we'll create a main.php file where we will get a taste of what's possible with this library.



```
<?php
```

5) Next copy over the following code:



```
<?php

//1
//require the image_3d library
require_once('./Image_3D/Image/3D.php');

//2
// Create the blank three-dimensional space
$world = new Image_3D();
$world->setColor(new Image_3D_Color(255,
255, 255));
```

```
//3
// A blue light from the left
$light1 = $world->createLight('Light', array(-300, 0, -300));
$light1->setColor(new Image_3D_Color(252, 175, 62));
// A blue light from the upper right
$light2 = $world->createLight('Light', array(300, -300, -300));
$light2->setColor(new Image_3D_Color(164, 0, 0));

//4
// Build the sphere object
$sphere = $world->createObject('sphere', array('r' => 85, 'detail' => 5));
$sphere->setColor(new Image_3D_Color(255, 255, 255));

//5 final
// Render and save the 2-D image
$world->createRenderer('perspectively');
$world->createDriver('gd');
$world->render(400, 400, 'object.png');
```

What is all this code?

1) We start by requiring the library

```
//1
//require the image_3d library
require_once('./Image_3D/Image/3D.php');
```

2) Then we need to create a space for the 3D renders we call it the $world and we set it's background to be the color white

```
//2
// Create the blank three-dimensional space
$world = new Image_3D();
$world->setColor(new Image_3D_Color(255,
255, 255));
```

3) Then we have to create some lights or our object will be dark and detail will not be shown. We create an red and orange lights at different positions that will radiate light in 360 degrees from their positions.

```
//3
// A blue light from the left
$light1 = $world->createLight('Light', array(-300, 0, -300));
$light1->setColor(new Image_3D_Color(252, 175, 62));
// A blue light from the upper right
$light2 = $world->createLight('Light', array(300, -300, -300));
$light2->setColor(new Image_3D_Color(164, 0, 0));
```

4) We will now create our object which is a simple sphere. We can specify the detail of it and color as well and I recommend playing with those values to really understand what happens.

```
//4
// Build the sphere object
$sphere = $world->createObject('sphere', array('r' => 85, 'detail' => 5));
$sphere->setColor(new Image_3D_Color(255, 255, 255));
```

5) No to the rendering. For this example we will use PG as our renderer since we want a PNG, a snapshot of the object in space, if we wanted to view it dynamically on a page we can output a JS file or ASCII.

```
//5 final
// Render and save the 2-D image
$world->createRenderer('perspectively');
$world->createDriver('gd');
$world->render(400, 400, 'object.png');
```

That’s it!

You completed your first PHP tutorial on generating 3D images.