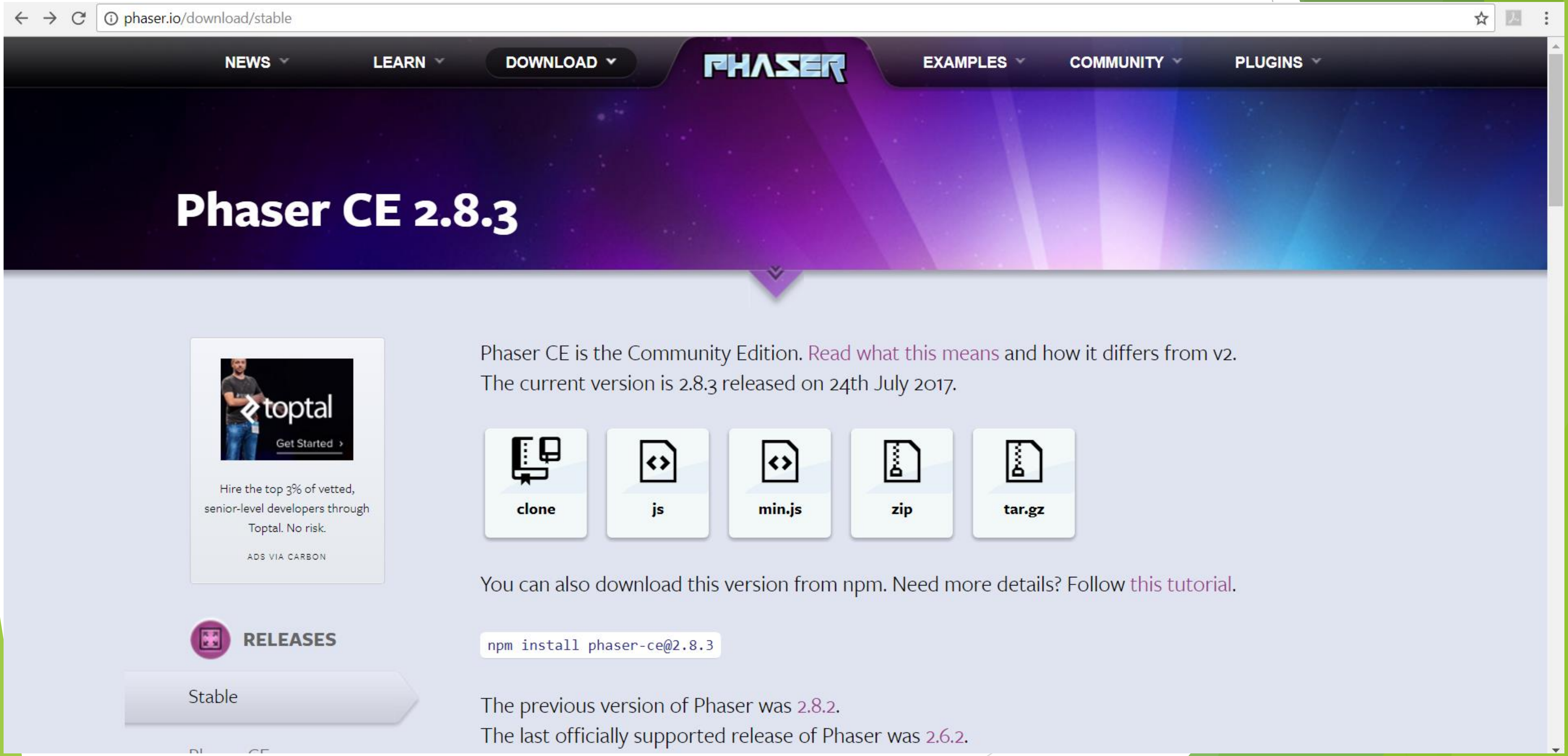
HTML5 and JavaScript Games Programming Week 5

Mario.Soflano@uws.ac.uk

Initialisation - Phaser

- Phaser is one of the most popular JavaScript game framework
- Phaser renders using both WebGL and Canvas internally for performance efficiency
- Phaser has a loader function to make assets loading easier and the loader can also track the assets loading
- Phaser has 3 different physics systems for lightweight to complex game
- Phaser has animation and particle systems
- Phaser has camera system which can be manipulated and allows various inputs

Initialisation – Phaser



Phaser CE 2.8.3

Phaser CE is the Community Edition. Read what this means and how it differs from v2.
The current version is 2.8.3 released on 24th July 2017.

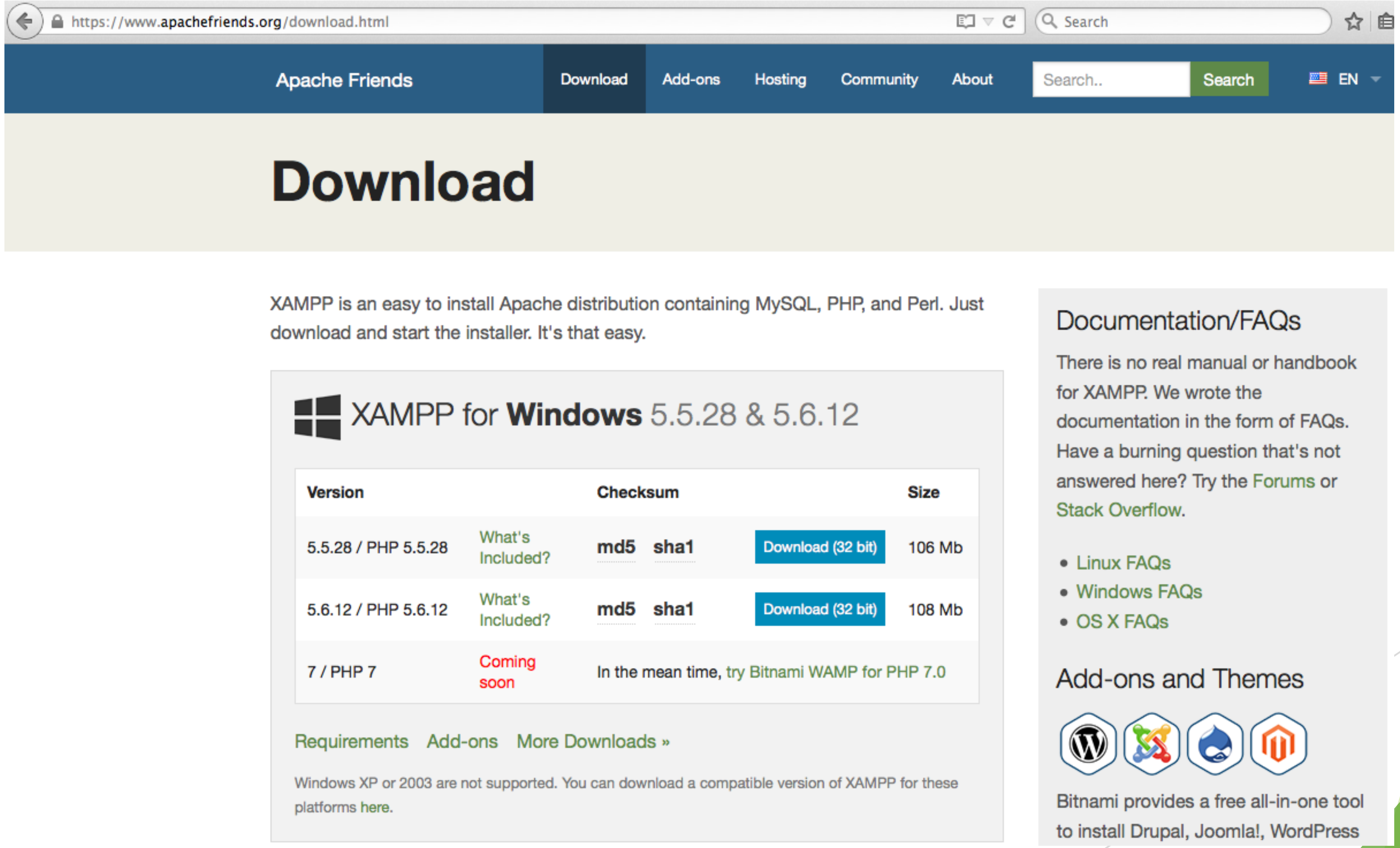
You can also download this version from npm. Need more details? Follow this tutorial.

```
npm install phaser-ce@2.8.3
```

The previous version of Phaser was 2.8.2.
The last officially supported release of Phaser was 2.6.2.

Initialisation - XAMPP

- Install XAMPP web server package (https://www.apachefriends.org/download.html)



Download

XAMPP is an easy to install Apache distribution containing MySQL, PHP, and Perl. Just download and start the installer. It's that easy.

XAMPP for **Windows** 5.5.28 & 5.6.12

Version		Checksum			Size
5.5.28 / PHP 5.5.28	What's Included?	md5	sha1	Download (32 bit)	106 Mb
5.6.12 / PHP 5.6.12	What's Included?	md5	sha1	Download (32 bit)	108 Mb
7 / PHP 7	Coming soon	In the mean time, try Bitnami WAMP for PHP 7.0			

Requirements Add-ons More Downloads »

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

Documentation/FAQs

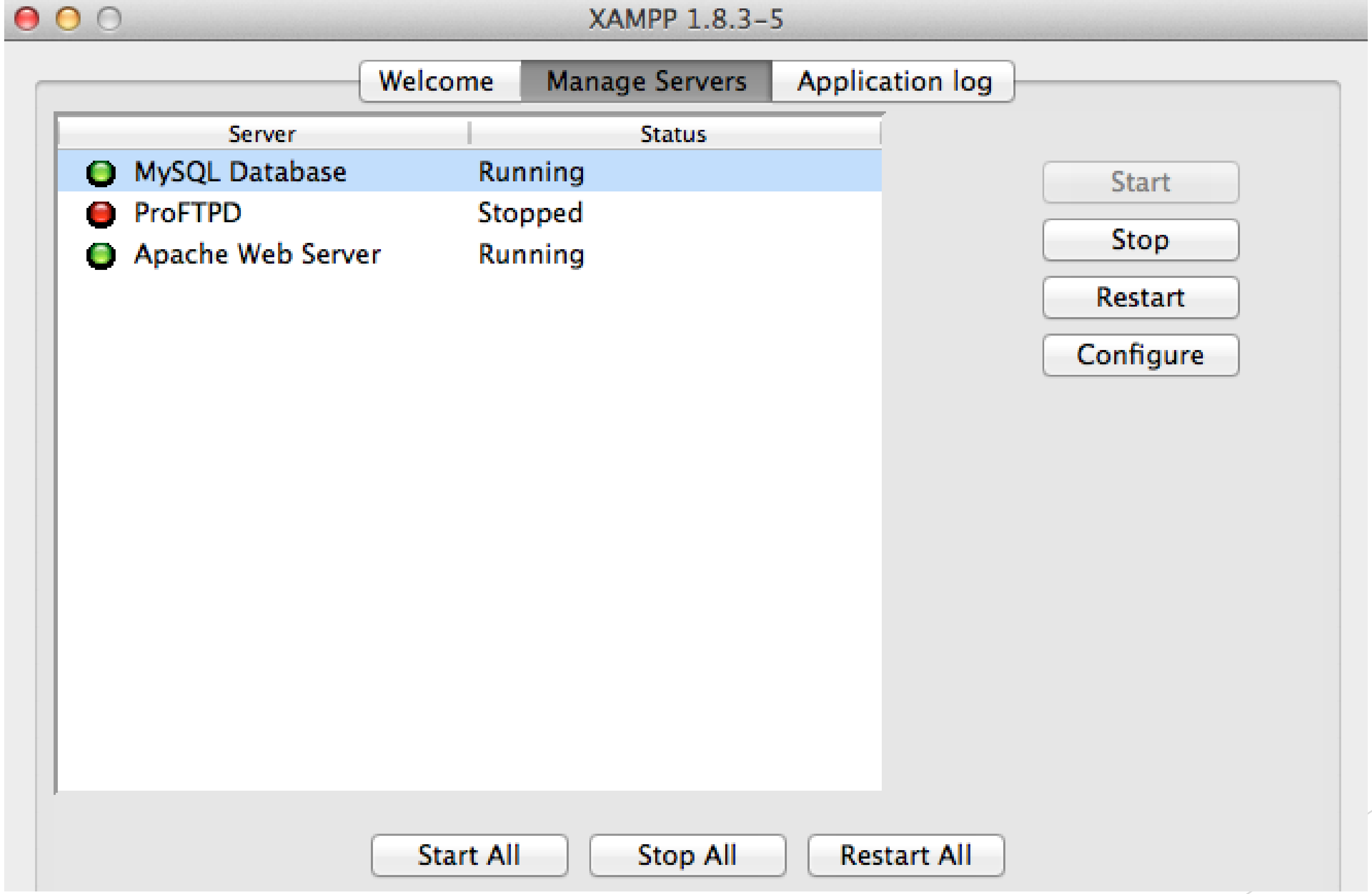
There is no real manual or handbook for XAMPP. We wrote the documentation in the form of FAQs. Have a burning question that's not answered here? Try the Forums or Stack Overflow.

- Linux FAQs
- Windows FAQs
- OS X FAQs

Add-ons and Themes

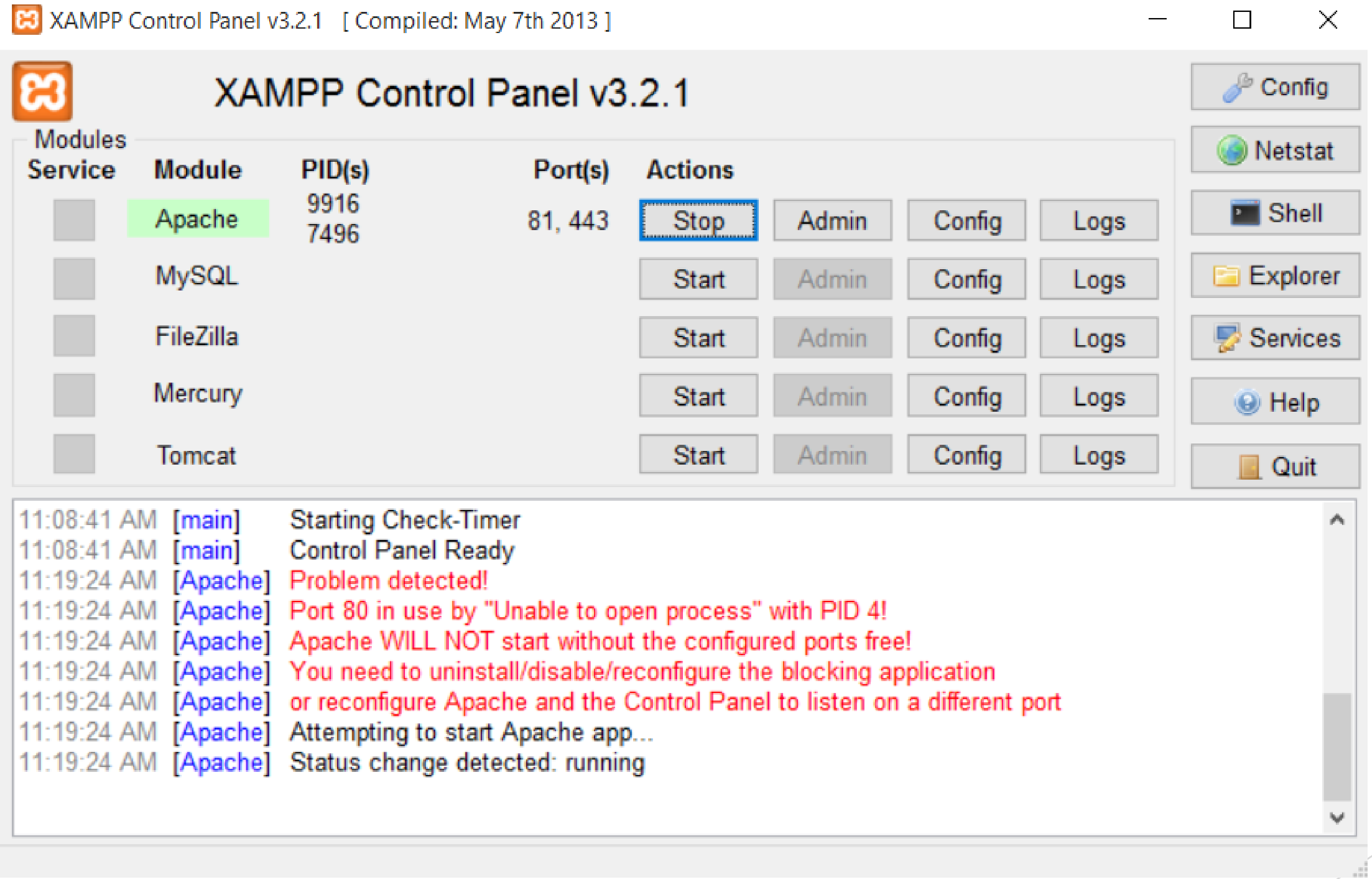
Bitnami provides a free all-in-one tool to install Drupal, Joomla!, WordPress

Initialisation - XAMPP



Server	Status
● MySQL Database	Running
● ProFTPD	Stopped
● Apache Web Server	Running

Initialisation - XAMPP



XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]
XAMPP Control Panel v3.2.1
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

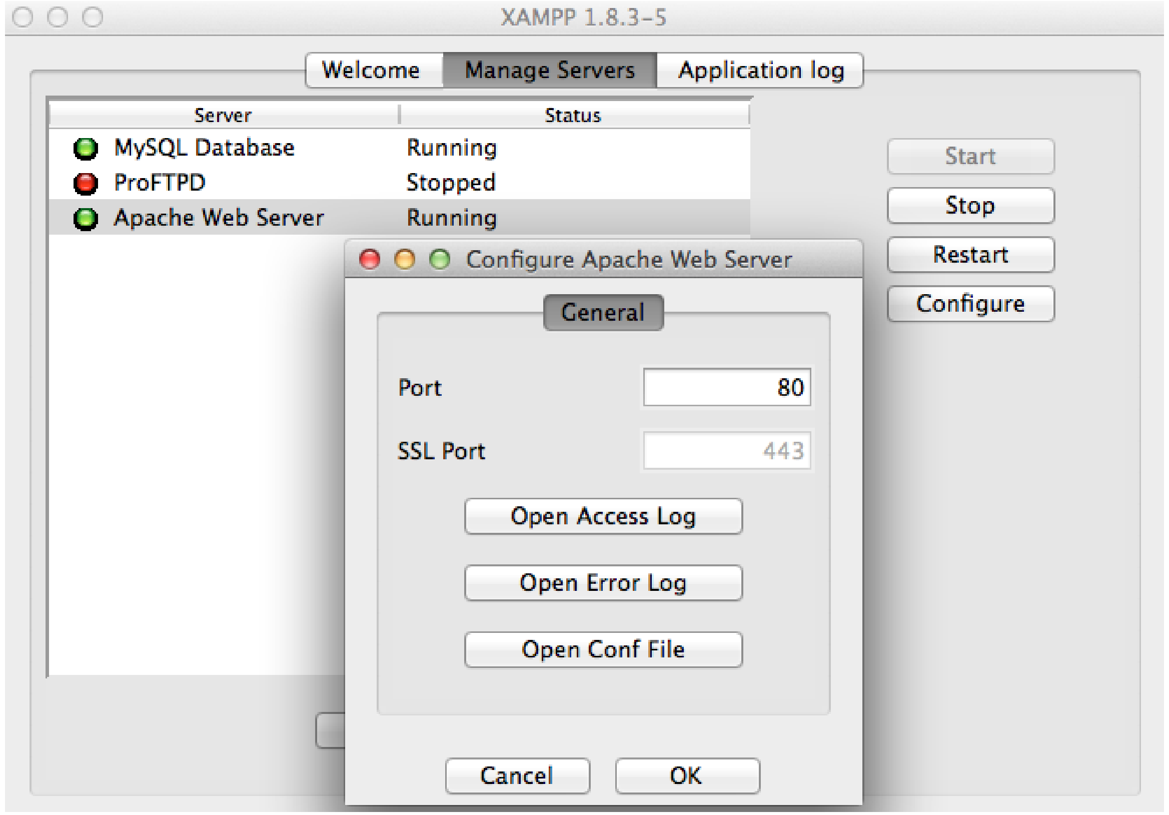
Service	Module	PID(s)	Port(s)	Actions			
□	Apache	9916 7496	81, 443	Stop	Admin	Config	Logs
□	MySQL			Start	Admin	Config	Logs
□	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
□	Tomcat			Start	Admin	Config	Logs

```
11:08:41 AM  [main]     Starting Check-Timer
11:08:41 AM  [main]     Control Panel Ready
11:19:24 AM  [Apache]   Problem detected!
11:19:24 AM  [Apache]   Port 80 in use by "Unable to open process" with PID 4!
11:19:24 AM  [Apache]   Apache WILL NOT start without the configured ports free!
11:19:24 AM  [Apache]   You need to uninstall/disable/reconfigure the blocking application
11:19:24 AM  [Apache]   or reconfigure Apache and the Control Panel to listen on a different port
11:19:24 AM  [Apache]   Attempting to start Apache app...
11:19:24 AM  [Apache]   Status change detected: running
```

Save your files in a folder inside “<xampp installation folder>/htdocs/<your folder>”

Initialisation - XAMPP

- If the Apache Web Server can not start, it is most likely because the port is blocked. Change the port number in Apache configuration file or through XAMPP control panel (Mac)



XAMPP 1.8.3-5

Server	Status
● MySQL Database	Running
● ProFTPD	Stopped
● Apache Web Server	Running

Configure Apache Web Server

General

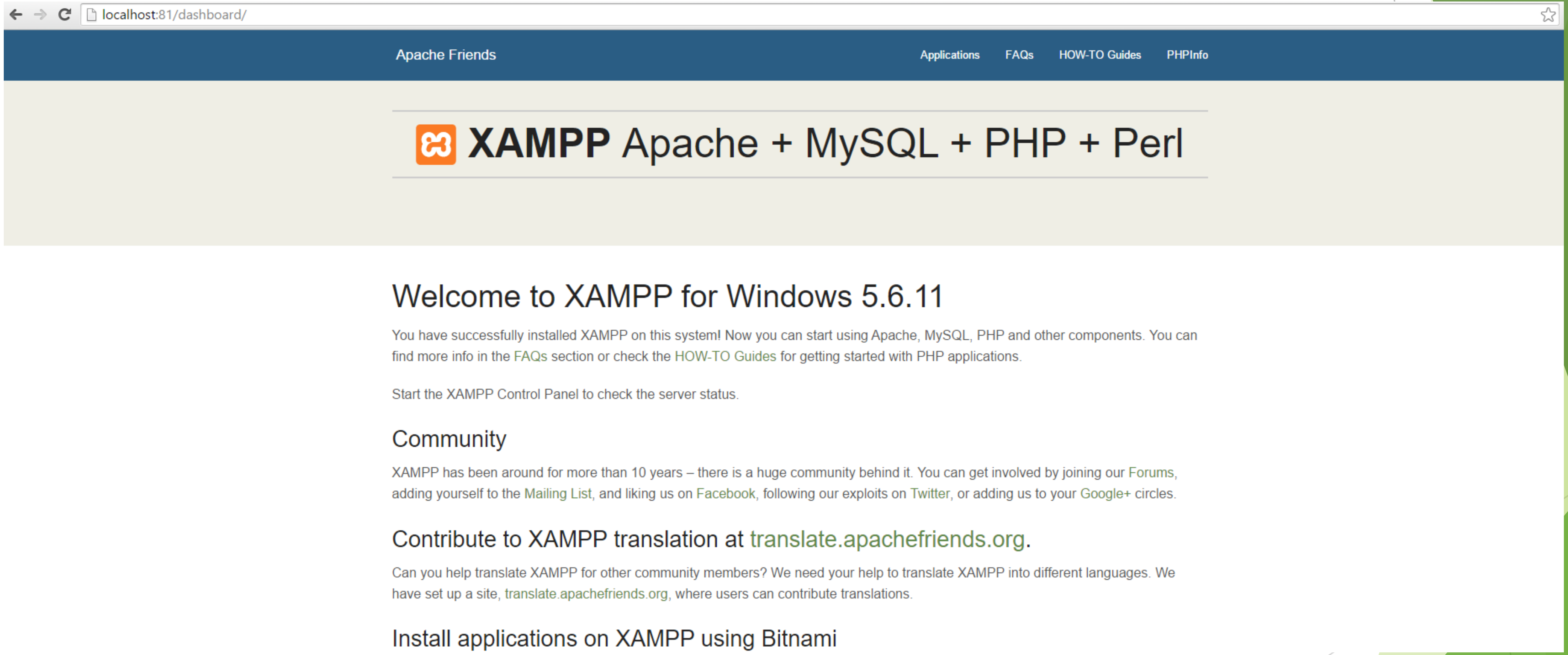
Port: 80

SSL Port: 443

Initialisation - XAMPP

- To change the port number through a web server config file, click “Config” on Xampp’s control panel and choose “Apache (httpd.conf)” to open the config file
- In the httpd.conf, change “Listen 80” to “Listen 81” and “ServerName localhost:80” to “ServerName localhost:81”
- Once the port is changed (for example to port 81), use “//localhost:81/<your folder>” to access the page in the web server

Initialisation - XAMPP



XAMPP Apache + MySQL + PHP + Perl

Welcome to XAMPP for Windows 5.6.11

You have successfully installed XAMPP on this system! Now you can start using Apache, MySQL, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

Start the XAMPP Control Panel to check the server status.

Community

XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our Forums, adding yourself to the Mailing List, and liking us on Facebook, following our exploits on Twitter, or adding us to your Google+ circles.

Contribute to XAMPP translation at translate.apachefriends.org.

Can you help translate XAMPP for other community members? We need your help to translate XAMPP into different languages. We have set up a site, translate.apachefriends.org, where users can contribute translations.

Install applications on XAMPP using Bitnami

Initialisation - XAMPP

- Save your files in a folder inside “<xampp installation folder>/htdocs/<your folder>”
- PLEASE REMEMBER TO BACKUP YOUR WORK:
  - Since the files will be stored in the C drive, it may be accessed by whoever login to the PC
  - There is a possibility the university network system will delete any “unrecognised” files after the user is logout

Setup

```
<!doctype html>
<html>
    <head>
        <script src="library/phaser.min.js"></script>
  </head>
  <body>
  </body>
</html>
```

Setup

new Phaser.Game(width, height, renderer, parent, state, transparent, antialias, physicsConfig);

Renderer : Phaser.AUTO will auto-detect, Phaser.WEBGL, Phaser.CANVAS or Phaser.HEADLESS (no rendering at all). Default: Phaser.AUTO

Parent : The DOM element into which this games canvas will be injected. Either a DOM ID (string) or the element itself.

State : The default state object. A object consisting of Phaser.State functions (preload, create, update, render) or null.

Transparent : Use a transparent canvas background or not. Default: false

Antialias : Draw all image textures anti-aliased or not. The default is for smooth textures, but disable if your game features pixel art. Default: true

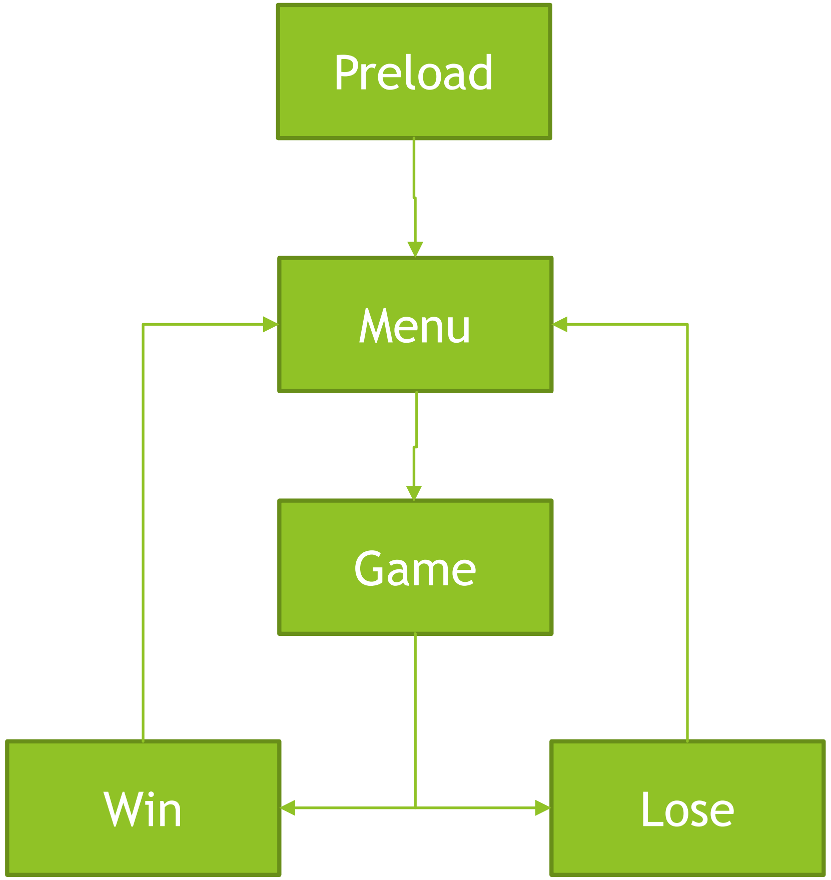
physicsConfig : A physics configuration object to pass to the Physics world on creation.

Setup

```
<!doctype html>

<html>

    <head>

    <script src="library/phaser.min.js"></script>

    <script>

        var game = new Phaser.Game(800, 600, Phaser.AUTO, “week1” ', { preload:
preload,
create: create });

       function preload() { }

       function create() { }

    </script>

  </head>

  <body>

    <div id="week1"></div>

  </body>

</html>
```

States

- Encapsulation



States

States can accommodate object’s behavioural change without using numerous conditional statements

index.html

- Include the JS files of the states (<script src=“firststate.js"></script>)
- Add the states after the game object is created (game.state.add(”FirstState", firststate))
- Start a state (game.state.start("FirstState”))

firststate.js

- Create a state object (var firststate= function(){ };)
- Create prototypes for the state (firststate.prototype = { preload: function(){ }, create: function(){ } })
- Start the next state (game.state.start(”SecondState”))

States

index.html

```
<!doctype html>
<html>
    <head>
        <script src="library/phaser.min.js"></script>
        <script src="src/preloadFile.js"></script>
        <script src="src/menuFile.js"></script>
        <script src="src/gameFile.js"></script>
        <script src="src/winFile.js"></script>
        <script src="src/loseFile.js"></script>
```

States

index.html

```
        <script>
                var game = new Phaser.Game(800, 600, Phaser.AUTO, "game");
                game.state.add(“preloadState",preloadFunc);
                game.state.add(“menuState",menuFunc);
                game.state.add(“gameState",gameFunc);
                game.state.add(“winState",winFunc);
                game.state.add(“loseState",loseFunc);
                game.state.start(" preloadState ");
        </script>
    </head>
    <body>
    </body>
</html>
```

States

preloadFile.js

```
var preloadFunc = function(){ };
preloadFunc.prototype = {
    preload: function(){
            console.log(”On Preload function of preloadFunc”);
    },
    create: function(){
        console.log(”On Create function of preloadFunc”);
        this.game.state.start("menuState");
    }
}
```

Reserved Functions

- preload: the first function to be called. Normally used to load all the assets
- loadUpdate: normally used to create a progress bar
- create: the function called right after Preload. Normally used to create objects such as sprites, particles, etc
- update: the function which is called on every frame. Normally used to create animations, character movements, collisions, inputs, etc
- render: similar to the update function however this function is called after the canvas / WebGL render. Normally used to debug and post-production effects

Geometry

```
var circle = new Phaser.Circle(x, y, diameter);
var ellipse = new Phaser.Ellipse(x, y, width, height);
var line = new Phaser.Line(x1, y1, x2, y2);
var point = new Phaser.Point(x1, y1);
var rectangle = new Phaser.Rectangle(x, y, width, height)
```

Loading Resources

Loading Image

```
load.image(uniqueKey, fileName);
```

Displaying Image

```
add.sprite(x, y, uniqueKey);
```

Displaying Text

```
add.text(x, y, text, style);
```

Loading Audio

```
load.audio(uniqueKey, audioFileName);
```

Displaying Resources

```
<!doctype html>
<html>
    <head>
        <script src="library/phaser.min.js"></script>
        <script>
            var music;
            var game = new Phaser.Game(800, 600, Phaser.AUTO, "week3", { preload: preloadstate, create: createstate });
            function preloadstate () {
                game.load.image('myImage', 'assets/theimagefile.png');
                game.load.audio('myBgMusic', ['assets/audio1.mp3', 'assets/audio1.ogg']);
            }
            function createstate () {
                game.add.sprite(0, 0, 'myImage');
                game.add.text(150, 150, 'Hello World', { font: "20px Arial", fill: "#ffffff", align: "left" });
                music = game.add.audio('myBgMusic');
                music.play();
            }
        </script>
    </head>
    <body><div id="week3"></div></body>
</html>
```

Loader

Create a “loading page” when loading many assets. Instantiate loader:

- game.load.onLoadStart.add(loadStart, this) : when the loader starts to load the assets
- game.load.onFileComplete.add(fileComplete, this) : this callback shows the progress of the loader. The parameters on this function are: *progress* (the progress in percentage), *cacheKey* (the assets being loaded), *success* (a boolean of the status of the loader), totalLoaded (the number of the assets loaded), totalFiles (the total number of assets to be loaded)
- game.load.onLoadComplete.add(loadComplete, this) : when the loader has finished loading the assets
- onLoadStart : can put message such as “Loading…”
- onLoadComplete : can put message such as “Loading Completed” and start the game

(http://phaser.io/examples/v2/loader/load-events)

Resources

- http://phaser.io/