Appendix: Conversion Cheat-sheet (1st Draft)

Here is ***the original Phaser v2.4.4 cheat sheet*** (aka ***"crib sheet"***).

I've updated this content to Phaser CE (v2.16.1). I've added a comparison chart to Phaser III API syntax. You can use this chart to convert your Phaser CE games ***into or from*** Phaser III.

%% RAW - original below

Starting a new game

Reference: http://docs.phaser.io/Phaser.Game.html#Game

```
var game = new Phaser.Game(width, height, renderer, "parent");
//All parameters are optional but you usually want to set width and height
//Remember that the game object inherits many properties and methods!
```

Creating a game state object

Reference: http://docs.phaser.io/Phaser.State.html

```
playState = {

    init: function() {
    //Called as soon as we enter this state
    },

    preload: function() {
    //Assets to be loaded before create() is called
    },

    create: function() {
    //Adding sprites, sounds, etc...
    },

    update: function() {
    //Game logic, collision, movement, etc...
    }
};
```

Managing game states

Reference: http://docs.phaser.io/Phaser.StateManager.html

```
game.state.add('play', playState);
game.state.start('play');

//It also has functions useful for debugging and whatnot
```

```
console.log("Currently at the "+ game.state.getCurrentState() +" game state!"
);
```

Adjusting the game to any screen size

Reference: http://phaser.io/docs/2.4.4/Phaser.ScaleManager.html

```
this.scale.scaleMode = Phaser.ScaleManager.SHOW_ALL;
this.scale.pageAlignHorizontally = true;
this.scale.pageAlignVertically = true;
this.scale.setScreenSize(true); // setScreenSize doesn't seem to exist anymor
e in the current documentation
```

Working with globals

Reference: http://www.html5gamedevs.com/topic/4247-where-to-put-global-var-in-the-basic-template/

```
//Declare it outside of any functions
//This way they persist through state changes
game.global = {
    mute: false,
    score: 0,
    bestScore: 100
};

//Then we can change them anywhere
game.global.mute = true;
game.global.bestScore = game.global.score;
```

Using local storage

Reference: http://www.w3schools.com/html/html5_webstorage.asp

```
//It can only store strings
localStorage.setItem('itemKey', 'myContent');

//You can store entire objects by doing this:
localStorage.setItem('myObject', JSON.stringify(myObject));

//Then you just get the string you stored!
localStorage.getItem('itemKey');
```

Loading an image/music/asset

Reference: http://docs.phaser.io/Phaser.Loader.html

```
function preload() {
    game.load.image('key', 'path/to/file.png');
    game.load.audio('key', ['path/to/file.mp3', 'path/to/file.ogg']);
```

```
    game.load.spritesheet('key', 'path/to/file', frameWidth, frameHeight);
}
```

Setting a background color

Reference: http://docs.phaser.io/Phaser.Stage.html

```
//Setting it to a nice, greyish blue
game.stage.backgroundColor = '#6d94b5';
```

Generating random numbers

Reference: http://docs.phaser.io/Phaser.RandomDataGenerator.html

```
var num = game.rnd.integerInRange(120, 480);
var intNum = game.rnd.integer();
var fracNum = game.rnd.frac();

//Spawn a sprite at a random position
game.add.sprite(game.world.randomX, game.world.randomY, 'mysprite');
```

Adding game objects

Reference: http://docs.phaser.io/Phaser.GameObjectFactory.html

```
function create() {
    //image, sprite, audio and others are all methods of the factory
    game.add.image(x, y, 'key');
    var player = game.add.sprite(x, y, 'key', frame, group);

    //You can add existing objects too
    var sprite = new Phaser.Sprite(game, x, y, 'key');
    game.add.existing(sprite);
}
```

Repositioning an object's anchor

Reference: https://github.com/photonstorm/phaser/wiki/Graphics

```
//Objects have an anchor property that goes from 0 (top left) to 1 (bottom ri
ght)
//It defaults to 0,0 but it can be changed easily
image.anchor.x = 0.2;
image.anchor.y = 1;

//This sets it in the middle
image.anchor.setTo(0.5,0.5);
```

Scaling an object

Reference: https://github.com/photonstorm/phaser/wiki/Graphics

```
//Objects have a scale property that defaults to 1
//Negative values essentially mirror it on the affected axis
image.scale.x = -1;

//This doubles the size of the object
image.scale.setTo(2,2);
```

Displaying an image

Reference: http://docs.phaser.io/Phaser.Image.html

```
function create() {
    game.add.image(x, y, 'key');
}
```

Working with sprites

Reference: http://docs.phaser.io/Phaser.Sprite.html

```
function create() {
    //Assign it to a variable so we can reference it
    var sprite = game.add.sprite(x, y, 'key');

    //Now we can access its properties and methods
    sprite.x = 200;
    sprite.y = 300;
}
```

Adding & playing animations

Reference: http://docs.phaser.io/Phaser.AnimationManager.html

```
function create() {
    sprite.animations.add('name', [frames], frameRate, loop);
    sprite.animations.play('name', frameRate, loop, killOnComplete);
}
```

Working with animations

Reference: http://docs.phaser.io/Phaser.Animation.html

```
function create() {
    //Assign it so we can reference it later
    var run = sprite.animations.add('name', [frames], frameRate, loop);

    //Second parameter is the context, usually 'this'
    run.onStart.add(listener, this);
}

function listener() {
```

```
    console.log("You just started running!");
}
```

Displaying text

Reference: http://docs.phaser.io/Phaser.Text.html

```
function create() {
    //Assigned for later use
    var label = game.add.text(x, y, "text", {style}, group);
    label.text = "I'm changing the text inside the label var!";
    //Center the text
    var txt = game.add.text(game.world.centerX, game.world.centerY, "My Text"
);
    txt.anchor.set(0.5, 0.5);
}
```

Tweening

Reference: http://docs.phaser.io/Phaser.Tween.html

```
//Adding an example sprite but you can tween pretty much anything
var player = game.add.sprite(100, 100, 'player');

game.add.tween(player)
    .to({x:500}, 400) //change player.x to 500 over 400ms
    .start();
```

Playing music

Reference: http://docs.phaser.io/Phaser.Sound.html

```
function create() {
    //Assign it so we can reference it
    var music = game.add.audio('key', volume, loop);
    music.loop = true;
    music.play();
}
```

Working with timers

Reference: http://docs.phaser.io/Phaser.Timer.html

```
//Three types of timers: looping, one-time event, repeat.
var looping = game.time.events.loop(delay, callback, context);
var once = game.time.events.add(delay, callback, context);
var repeat = game.time.events.repeat(delay, repeatCount, callback, context);
//You can also pass one last argument with the callback arguments

game.time.events.pause(loopingTimer);
game.time.events.remove(once);
```

Calculating elapsed time

Reference: http://docs.phaser.io/Phaser.Time.html

```
//You can get the current time
var currentTime = game.time.time;

//You can get the elapsed time (milliseconds) since the last update
var elapsedTime = game.time.elapsed;

//Or you can get the elapsed time (seconds) since the last event tracked
var lastEventTrackedTime = game.time.time;
//Do something...
var elapsedTime = game.time.elapsedSecondsSince(lastEventTrackedTime);
```

Input

Reference: http://docs.phaser.io/Phaser.Input.html

```
//Input can come from mouse, touch, or keyboard
//This is the parent object, with properties for setting how input works

//Allows up to a second between taps for a double click
game.input.doubleTapRate = 1000;

//Increases the hitbox for touch
game.input.circle = 66;
```

Mouse & touch input

Reference: http://docs.phaser.io/Phaser.Pointer.html

```
if (game.input.mousePointer.isDown) {
    console.log("Mouse X when you clicked was: "+game.input.mousePointer.x);
}

//Assign a callback and a context to a click event
game.input.onDown.add(callback, context);
```

Keyboard input

Reference: http://docs.phaser.io/Phaser.Keyboard.html

Keycodes: http://docs.phaser.io/Keyboard.js.html#sunlight-1-line-557

```
if (game.input.keyboard.justReleased(Phaser.Keyboard.SPACEBAR, 10000)) {
    console.log("Spacebar has been pressed in the last 10 seconds.");
}

//Assigning Up, Down, Left, and Right to a variable
var arrow =  game.input.keyboard.createCursorKeys();
```

```
if (arrow.up.isDown) {
    console.log("You are pressing the up arrow!");
}

//This will stop the arrow keys from scrolling the page
game.input.keyboard.addKeyCapture(arrow);
```

Working with groups

Reference: http://docs.phaser.io/Phaser.Group.html

```
//Remember to assign it so we can reference it
var enemies = game.add.group();

//We can add an already created object
enemies.add(button);

//Or we can create a new object in the group
enemies.create(x, y, 'enemy_sprite');

//You can use setAll to modify properties across all children
enemies.setAll('x', 500);
```

Adding physics to Sprites

Reference: http://docs.phaser.io/Phaser.Physics.Arcade.html

```
function create() {
    //First we start the system
    game.physics.startSystem(Phaser.Physics.ARCADE);

    //We then create our sprite & enable physics on it
    sprite = game.add.sprite(x, y, 'key');
    game.physics.enable(sprite, Phaser.Physics.ARCADE);

    //Now our sprite has an object body as a property
    sprite.body.velocity.setTo(x, y);
    sprite.body.bounce.set(0.8);
}
```

Handling Collision

Reference: http://docs.phaser.io/Phaser.Physics.Arcade.html#collide

```
function update() {
    //You can collide a group with itself
    game.physics.arcade.collide(sprites);

    //You can call a function when a collision happens
    game.physics.arcade.collide(sprites, monsters, callback);
```

```
    //You can perform additional checks with a process callback
    //If it returns false the collision will not happen
    game.physics.arcade.collide(sprites, monsters, null, processCallback);

    //Or you can check if two bodies overlap. This method avoids the impact
    //between then, keeping their velocities and properties
    game.physics.arcade.overlap(sprites, monsters, callback);

    //You can perform the following collisions:
    //Sprite vs Sprite or
    //Sprite vs Group or
    //Group  vs Group or
    //Sprite vs Tilemap Layer or
    //Group  vs Tilemap Layer
}
```

Camera

Reference: http://docs.phaser.io/Phaser.Camera.html

```
//Let's follow a sprite named 'missile'
game.camera.follow(missile);

//Once the missile explodes, maybe we want to reset the camera
game.camera.reset();

//Something cool is happening, let's pan the camera there
game.camera.x = 500;
```

Particles

Reference: http://docs.phaser.io/Phaser.Particles.Arcade.Emitter.html

```
emitter = game.add.emitter(x, y, maxParticles);
emitter.makeParticles('image');
emitter.setAlpha(min, max, rate, easing, yoyo);

//To use gravity on the emitter, start the physics system
game.physics.startSystem(Phaser.Physics.ARCADE);
emitter.gravity = 200;
emitter.start();
```

Debugging

Reference: http://docs.phaser.io/Phaser.Utils.Debug.html

```
//Print debug text
game.debug.text(game.time.physicsElapsed, 10, 20);
```

```
//Print debug body information
game.debug.bodyInfo(player, 10, 20);

//Show the sprite's hitbox as a green rectangle
game.debug.body(player);
```

Contribute!

Any recommendations are welcome. I noticed that the old-rumored cheat-sheet pointed to a “dead website” so I’ve made it available again here.