

A background image featuring a collection of Doraemon plush toys and a patterned wallpaper. The wallpaper has a repeating design of Doraemon's face and other characters like Peko-chan and the yellow dog. In the foreground, there are several plush figures: a large Doraemon with its mouth open, a smaller Doraemon, a character with a large orange head (likely Goro), and a small white cat-like figure.

Serious Fun Gaming in Javascript



Thai Typing Game



999
WPM



0:44

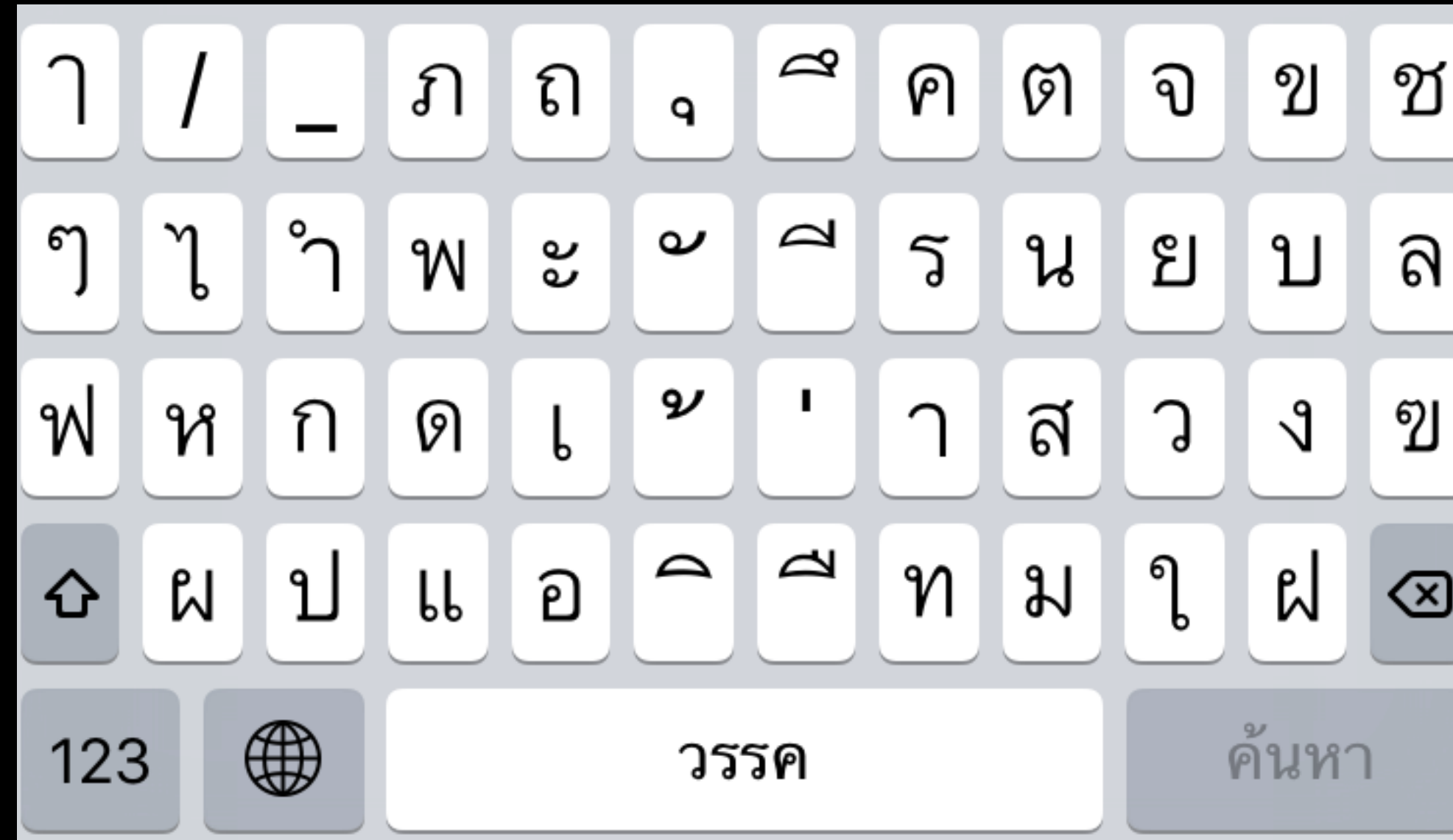
๒



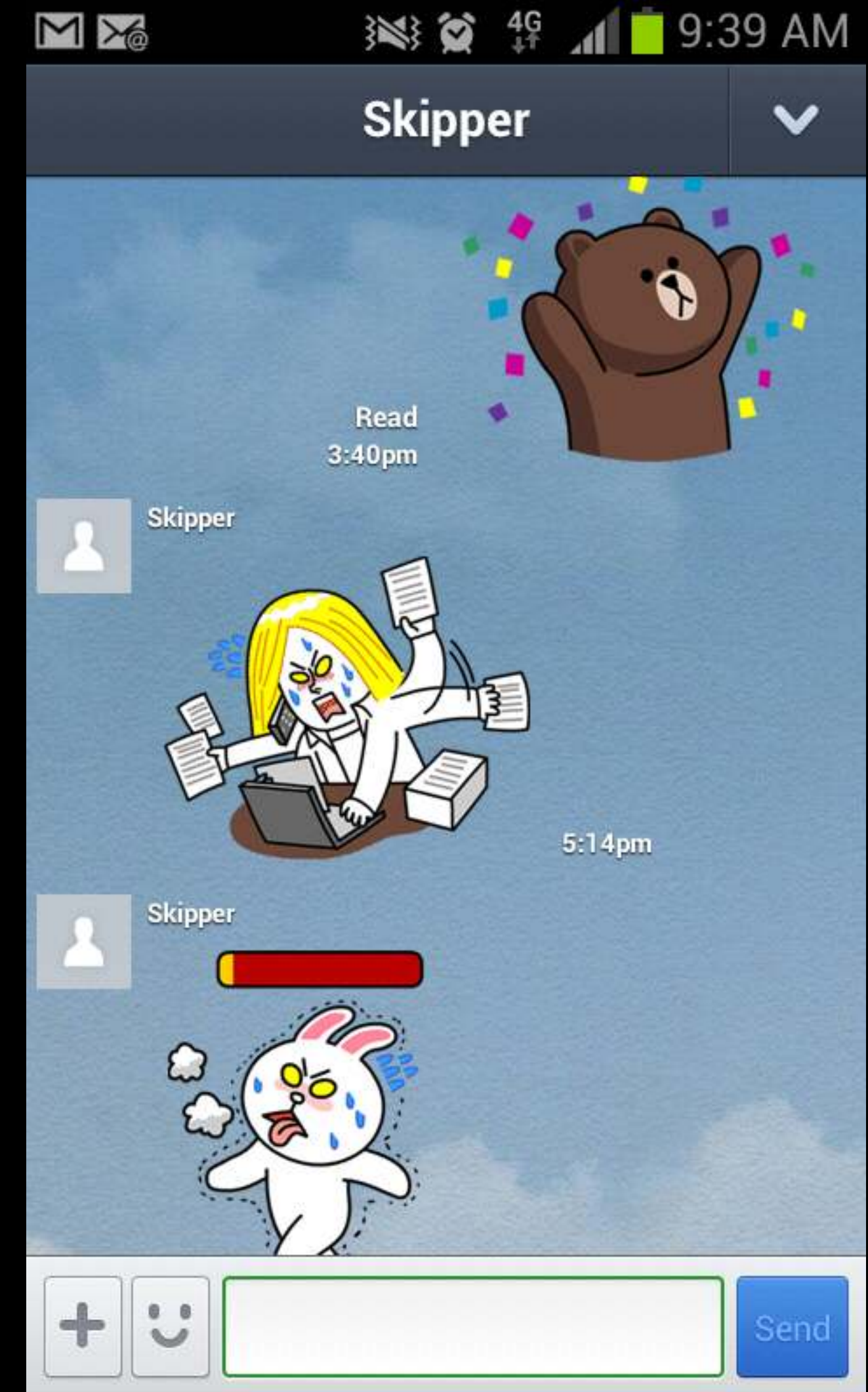
999999 x



Thai Keyboard

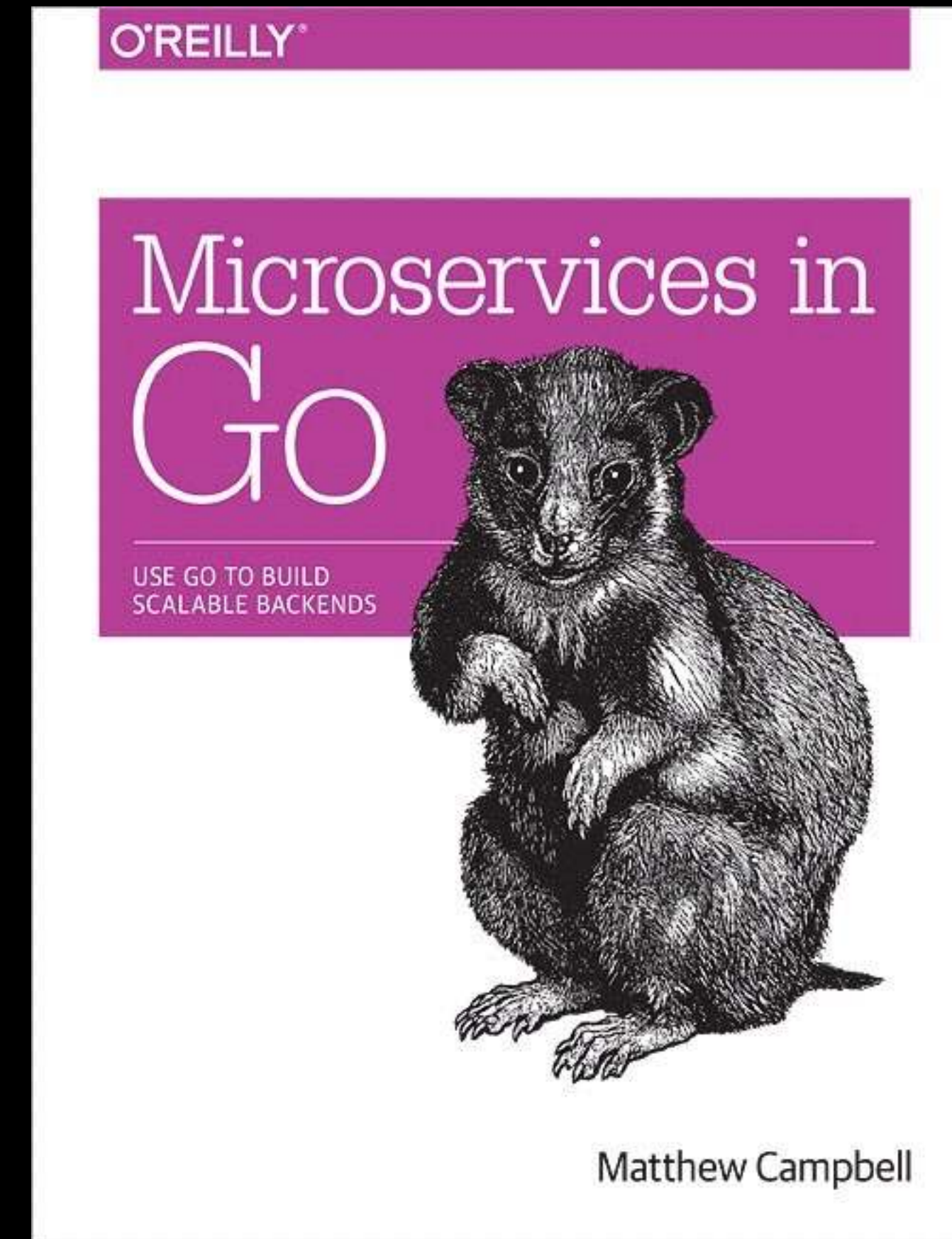


Stickers!



About me

- Lives in Bangkok
- Write GO for Digital Ocean
- Server side developer
- Book on Go





Gaming

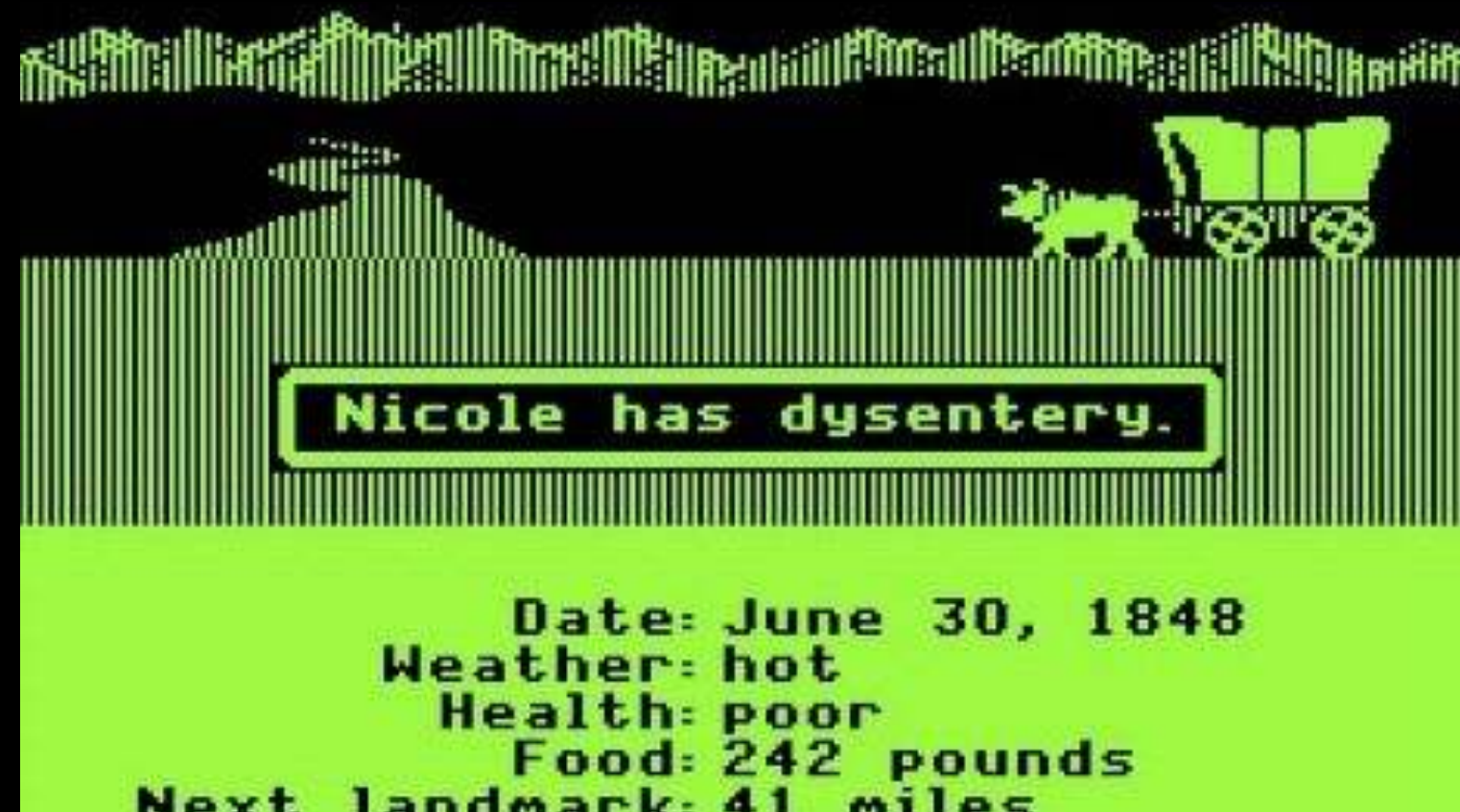


Nekoatsume



A close-up photograph of a board game. In the center, a blue circular space is labeled 'START' in black capital letters. A white arrow points from the bottom towards the 'START' space. Surrounding the 'START' space are four pawns: a black pawn on the left, a white pawn in the center, a green pawn at the top, and a brown pawn on the right. The board has a grid of squares, some of which are shaded blue. The background is a light gray surface.

What is serious gaming?

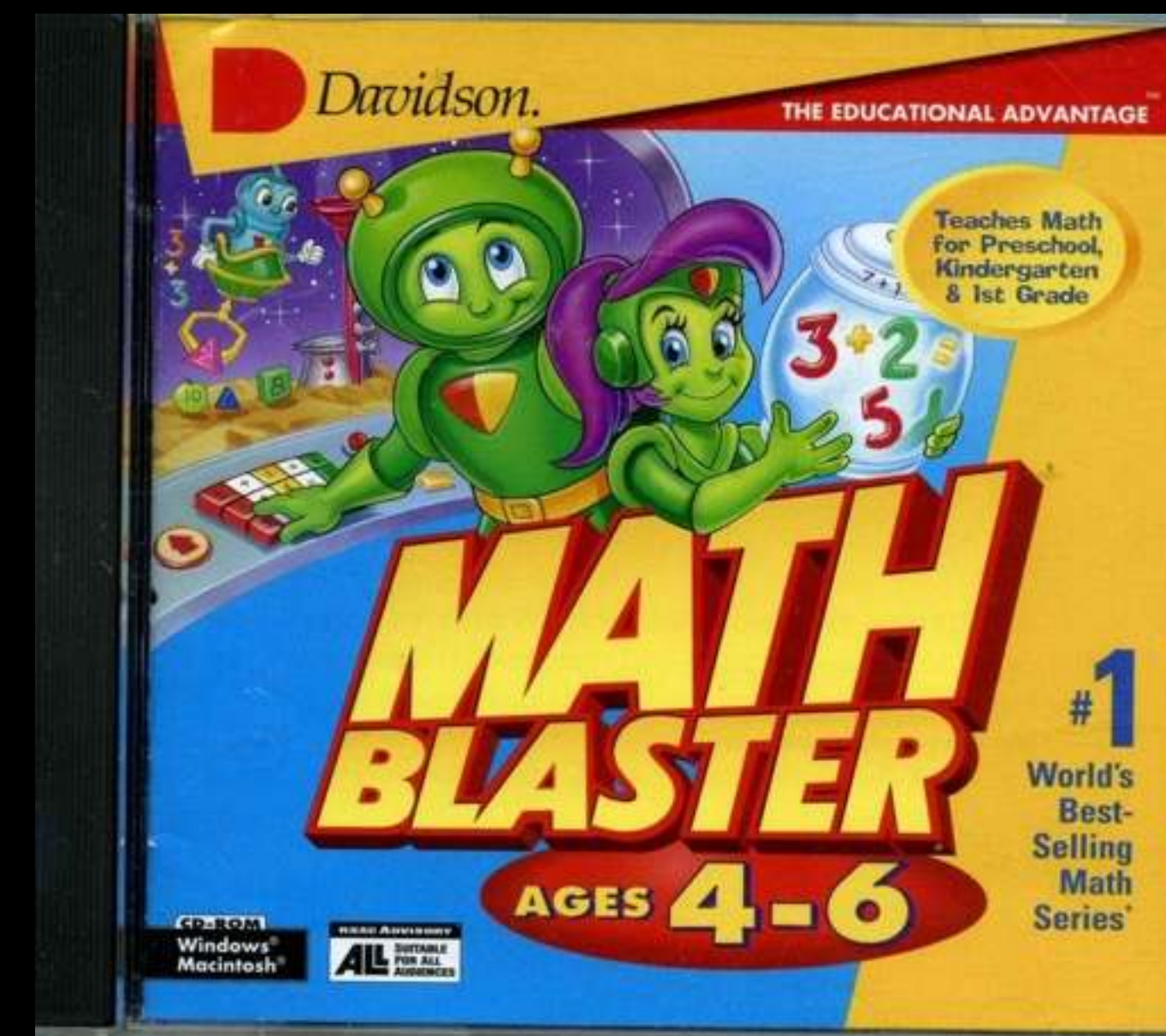


Level: 4

Equals 4

4+12	6-2	4+8	7-3		
		13-9	6-5	14-10	9+13
	11-9	4-0		12+10	8-4
7+4		10+2	6-2	5-1	7-3
		5-1	13+9	11-10	0+4

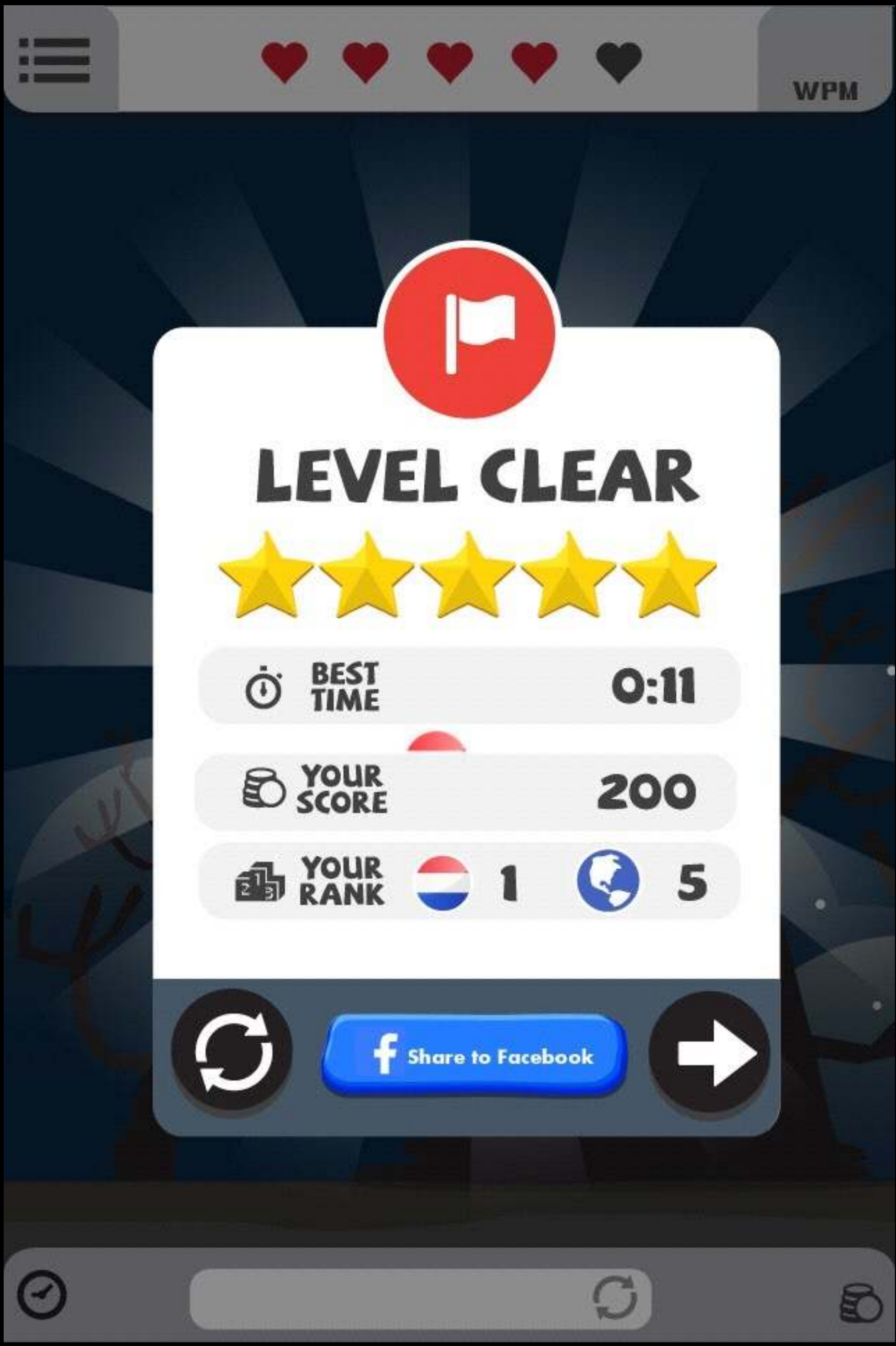
Score: 5600



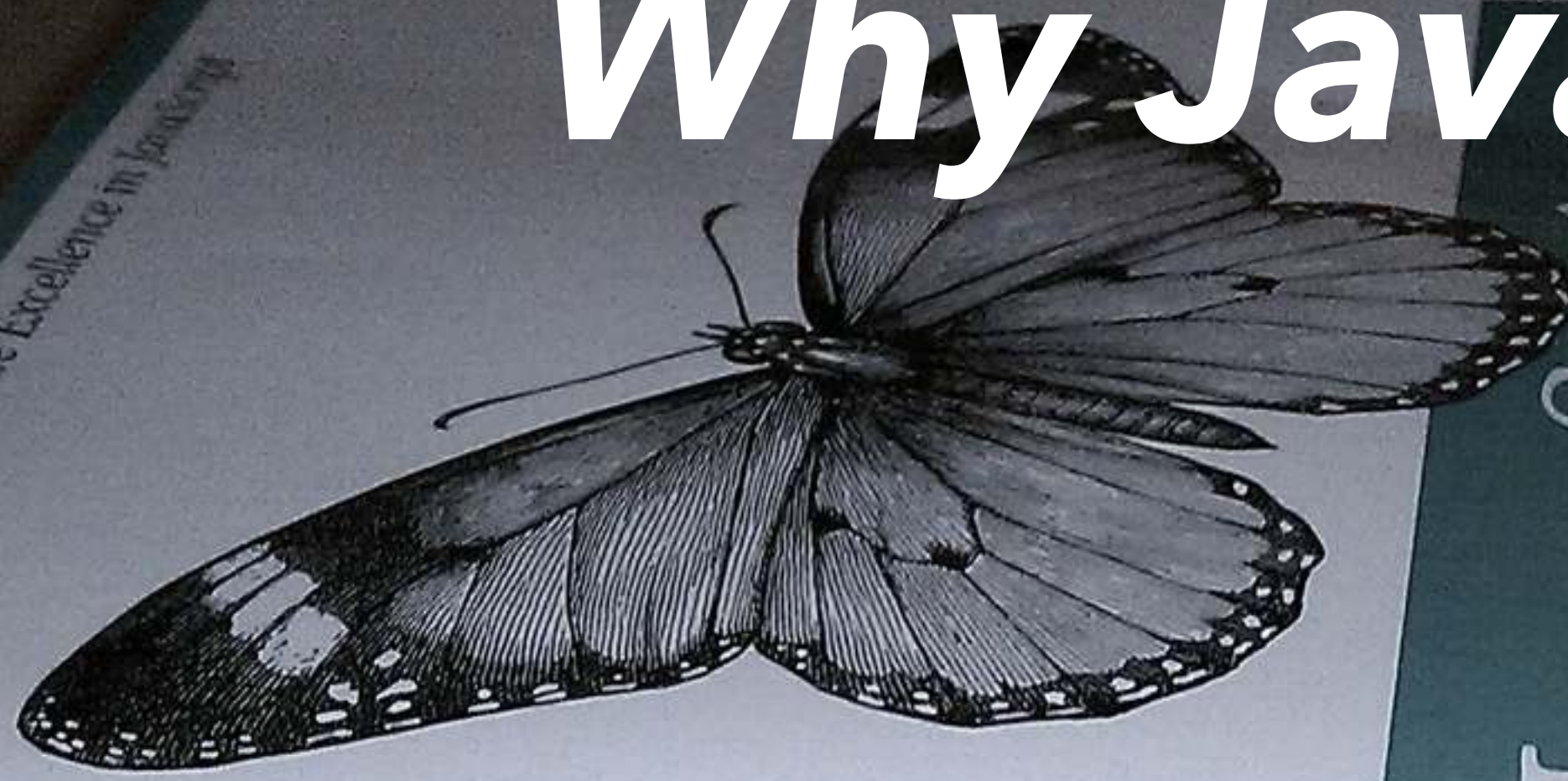


Gamification





Why Javascript?



Unearthing the Excellence in JavaScript

JavaScript The Good Parts

JavaScript: The Good Parts

Crockford



O'REILLY®



FIFTH EDITION

COVERS AJAX AND DOM SCRIPTING

JavaScript The Definitive Guide

Flanagan

O'REILLY®

Chrome File Edit View History Bookmarks People Window Help Wed 10:44 PM Matthew Campbell

localhost:3000

MonkeyType

WPM: 0

0:02

0

Elements Network Sources Timeline Profiles Resources

html body canvas

Styles Event Listeners DOM Breakpoints Properties

Filter

element.style {
outline: none;

Console Emulation Rendering

<top frame> Preserve log

words loaded level manager.js:69
["n", "u", "u", "n", "n", "u", "v", "v", "u", "u", "u", "u", "u"]

second success level manager.js:91

keydown 91 Play.js:900

keydown 16 Play.js:900

Menu.js — monkey-type

```
25 var spriteSheet = new createjs.SpriteSheet({  
26   images: [  
27     game.load.get("button_play_0"),  
28     game.load.get("button_play_1")  
29   ],  
30   frames: { width: 200, height: 200},  
31   animations: { normal: [0], hover: [1], c  
32 });  
33 var spriteSheetSetting = new createjs.SpriteSheet({  
34   images: [  
35     game.load.get("setting_btn_play_0"),  
36     game.load.get("setting_btn_play_1")  
37   ],  
38   frames: { width: 125, height: 125},  
39   animations: { normal: [0], hover: [1], c  
40 });  
41  
42 var button = new createjs.Sprite(spriteSheet,  
43   var buttonSetting = new createjs.Sprite(spriteSheet,  
44   var startBtn = new createjs.ButtonHelper(button,  
45   var settingBtn = new createjs.ButtonHelper(button,  
46  
47   button.on("click", function(evt){  
48     this.playGame();
```

Gulp Watch + BrowserSync

```
gulp.task('default', ['js'], function() {  
  browserSync.init({  
    server: {  
      baseDir: "./www"  
    }  
  });  
  
  gulp.watch("./www/**/*.js").on("change", bundle);  
  gulp.watch("./www/*.html").on("change", browserSync.reload);  
  gulp.watch("./www/assets/*").on("change", browserSync.reload);  
  gulp.watch("./www/css/*").on("change", browserSync.reload);  
});
```

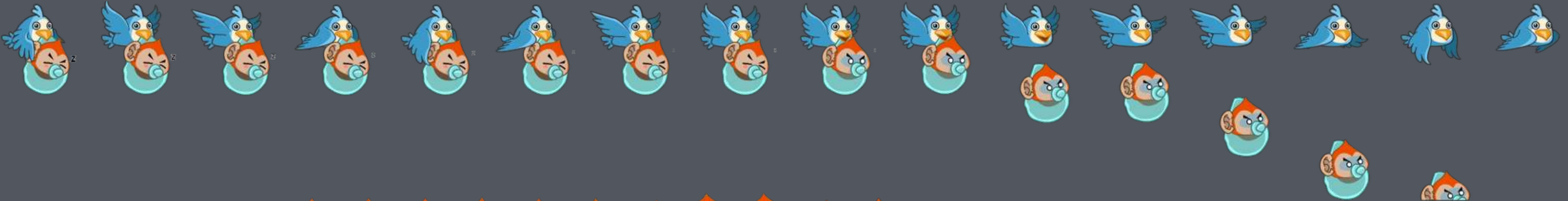
RequireJS

```
var $ = require("jquery");  
var ndgmr = require("../lib/vendor/ndgmr.utils");  
  
var ParallaxLayer = require("../lib/entities/parallax_layer.js");  
var Hero = require("../lib/entities/hero.js");  
var Monster = require("../lib/entities/monster.js");  
var ScoreLabel = require("../lib/entities/score_label.js");  
var TimeLabel = require("../lib/entities/time_label.js");  
  
var Timer = require("../lib/timer.js");  
var util = require("../lib/util");
```

A black and white photograph of a person with long hair, wearing a backpack, standing on a rocky shore and looking across a calm lake. The lake reflects the surrounding mountains and sky. The mountains are rugged and have patches of snow or ice. The overall mood is serene and contemplative.

CreateJS

Spritesheets



Creating a Spritesheet

```
var spriteSheetData = {
  animations : {
    run: {
      frames : [0, 1, 2, 3, 4, 5],
      speed: 0.2
    },
    jump: {
      frames : [6, 7, 8, 9],
      speed: 0.2
    },
    death: 10
  },
  frames : {
    width: 100, height: 100
  },
  images : ['assets/sprite/monkey_animate_100.png']
};

var ss = new createjs.SpriteSheet(spriteSheetData);

this.hero = new Hero({
  spriteSheet : ss,
  x : 0,
  y: 540,
  positionYSubtractSize: true,
  velocity : {x: 0, y:0},
  name: "hero"
});
```

Play Animation

```
this.hero.animation.gotoAndStop('run');
this.monster.animation.gotoAndStop('run');
```

Parallax Scrolling





Setup Layers

```
ground0 = that.parallaxLayer['ground0'];
ground0.graphics.y = that.gameContainer.height - that.parallaxLayer['ground0'].height;
ground0.width = width;

ground1 = that.parallaxLayer['ground1'];
ground1.graphics.y = that.gameContainer.height - ground1.height;

ground2 = that.parallaxLayer['ground2'];
ground2.graphics.y = that.gameContainer.height - ground2.height - ground0.height;
```

Parallax Implementation

```
render : function(){
    //if shapeA has moved completely off the left screen
    if(this.shapeA.x < this.outside){
        //move it to the back of shapeB
        var temp = this.shapeA;
        temp.x = this.shapeB.x + this.width;
        //switch shapeA to shapeB and shapeB to shapeA
        this.shapeA = this.shapeB;
        this.shapeB = temp;
    }

    this.shapeA.x += this.velocity.x * this.scaleFactorX;
    this.shapeB.x += this.velocity.x * this.scaleFactorX;
},
```

Game Loop

```
update: function() {
  if (this.play_paused) {
    return;
  }

  switch(this.state)
  {
    case State.MONSTER_DYING:
      if(this.monster.animation.currentAnimation == "death")
      {
        this.state = State.LEVEL_RUNNING;
        this.incrementWord();
      }
      break;
  }

  //check if current state allows game rendering
  if(RUNNABLE_STATES.indexOf(this.state) == -1) return;

  this.respawnMonster();

  this.hero.update();
  this.hero.render();

  this.monster.update();
  this.monster.render();
```

```
    if(!this.monster.dealtDamage && this.monster.isCollided(this.hero))
    {
      //todo use damage sound
      this.playAudio("soundJump");

      this.lifePoints -= this.monsterDamage;
      this.monster.dealtDamage = true;
      console.log('hero received damage', this.monsterDamage, 'current life points:');

      if(this.lifePoints <= 0)
      {
        this.hero.alive = false;
        this.hero.animation.gotoAndPlay("death");
      }
    }

    //update wpm counter
    this.wpmLabel.text = "WPM: " + this.calculateWpm();
    this.resizeWpmLabel();

    this.scoreLabel.update();
    this.scoreLabel.render(); //for concept, later both should be called through loop

    this.timeLabel.update();
    this.timeLabel.render(); //for concept, later both should be called through loop
    for(var i in this.parallaxLayer){
      this.parallaxLayer[i].update();
      this.parallaxLayer[i].render();
    }
    this.highlightWrittentext();
  },
```



Creatine

Scene creation / Transition

```
|var PlayScene = tine._scene({  
    initialize: function() {  
        console.log("init play scene");
```

```
        game.levelManager.loadNextLevel(function(){  
            game.director.replace('play');  
        });
```

```
//show preload scene  
game.director.add('preload', require("../scene/Preload.js"));  
game.director.replace('preload');
```

```
//activate touch support
```

A person is standing on a wooden dock that extends from a rocky shore into a calm lake. The dock is made of weathered wooden planks. In the background, a dense forest of evergreen trees covers the hillsides, and a mountain range is visible under a cloudy sky. The water is still, reflecting the surrounding landscape. The overall mood is serene and quiet.

Cordova

Fixing Native keyboard

```
if(window.cordova){  
    if(cordova.plugins && cordova.plugins.Keyboard) {  
        cordova.plugins.Keyboard.hideKeyboardAccessoryBar(true);  
    }  
}
```

Better background sound

```
// Cordova media plugin implementation  
if(window.cordova)  
{  
    var url = document.getElementById(id).getAttribute('src');  
  
    if(id == "soundGameplay")  
    {  
        sound = new Media(url, function(){  
            //repeat sound if not stopped  
            if(typeof that.bgMusic != "undefined")  
            {  
                that.playAudio(id);  
            }  
        });  
    }  
    else  
    {  
        sound = new Media(url);  
    }  
  
    sound.play();  
}
```



Electron

```
var app = require('app'); // Module to control application life.
var BrowserWindow = require('browser-window'); // Module to create native browser window.
// Report crashes to our server.
require('crash-reporter').start();
// Keep a global reference of the window object, if you don't, the window will
// be closed automatically when the JavaScript object is garbage collected.
var mainWindow = null;
// Quit when all windows are closed.
app.on('window-all-closed', function() {
  // On OS X it is common for applications and their menu bar
  // to stay active until the user quits explicitly with Cmd + Q
  if (process.platform !== 'darwin') {
    app.quit();
  }
});
// This method will be called when Electron has finished
// initialization and is ready to create browser windows.
app.on('ready', function() {
  // Create the browser window.
  mainWindow = new BrowserWindow({width: 800, height: 600});

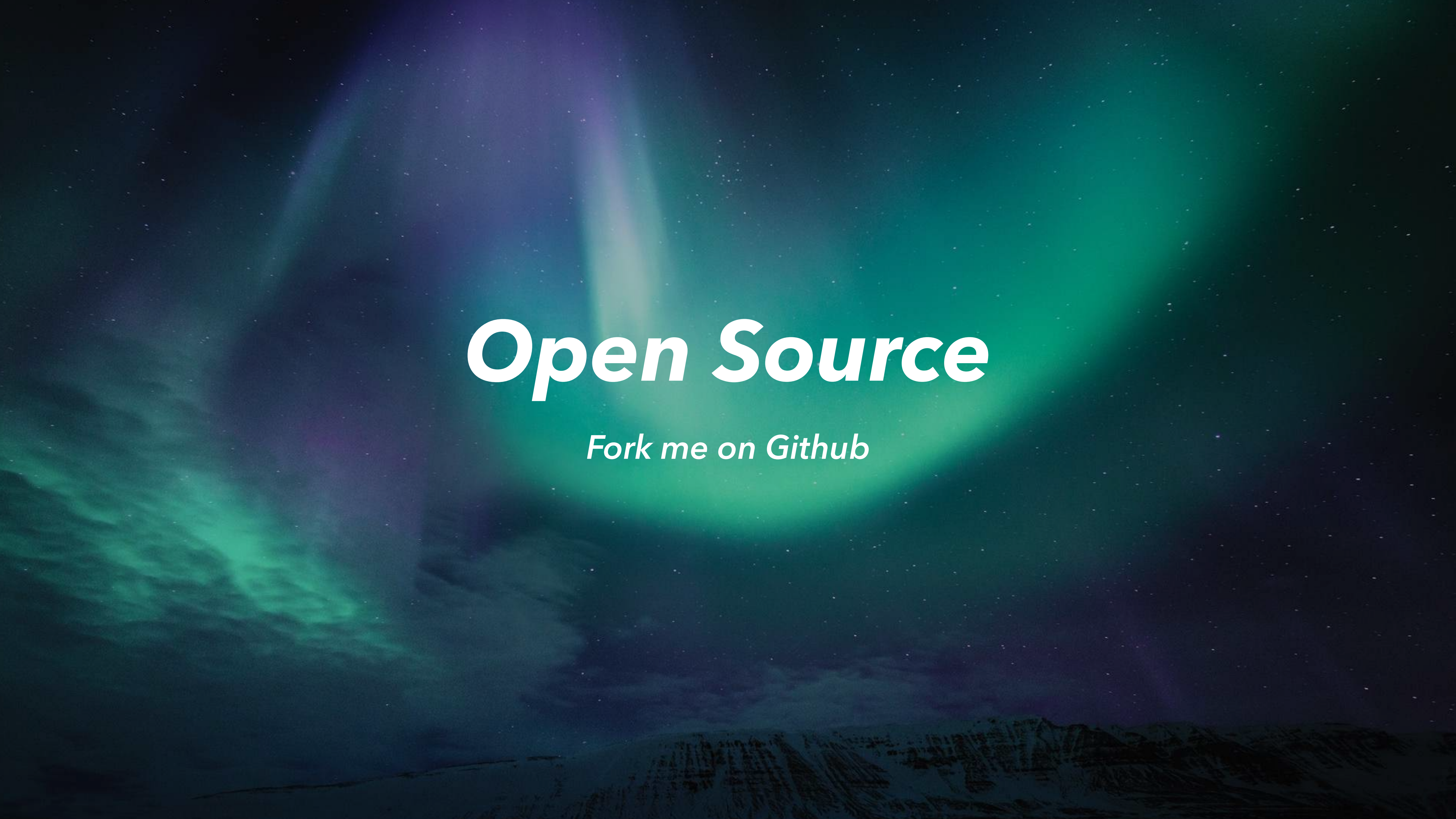
  // and load the index.html of the app.
  mainWindow.loadUrl('file://' + __dirname + '/www/index.html');

  // Open the DevTools.
  mainWindow.openDevTools();

  // Emitted when the window is closed.
  mainWindow.on('closed', function() {
    // Dereference the window object, usually you would store windows
    // in an array if your app supports multi windows, this is the time
    // when you should delete the corresponding element.
    mainWindow = null;
  });
});
```

Conclusion

- *Make games for learning*
- *Writing games in Javascript is easy*
- *Avoid cross platform tools if possible*

The background of the image is a photograph of the Aurora Borealis (Northern Lights) over a snowy mountain range. The aurora displays vibrant green and purple hues against a dark, star-filled night sky. The mountains in the foreground are covered in snow and appear dark under the low light.

Open Source

Fork me on Github

Questions?

Matthew Campbell

hyper@hyperworks.nu

@kanwisher

github.com/mattkanwisher

We're Hiring!

