CS74.42A Game Development

Fall 2018 ~ Ethan Wilde

Week 4

SANTA ROSA JUNIOR COLLEGE

Welcome

- Course Outline: This Week
- Textbook Reading This Week
- Software This Week
- Planning the Flow: Flowcharts to Map Systems
- Introduction to JavaScript

Course Outline

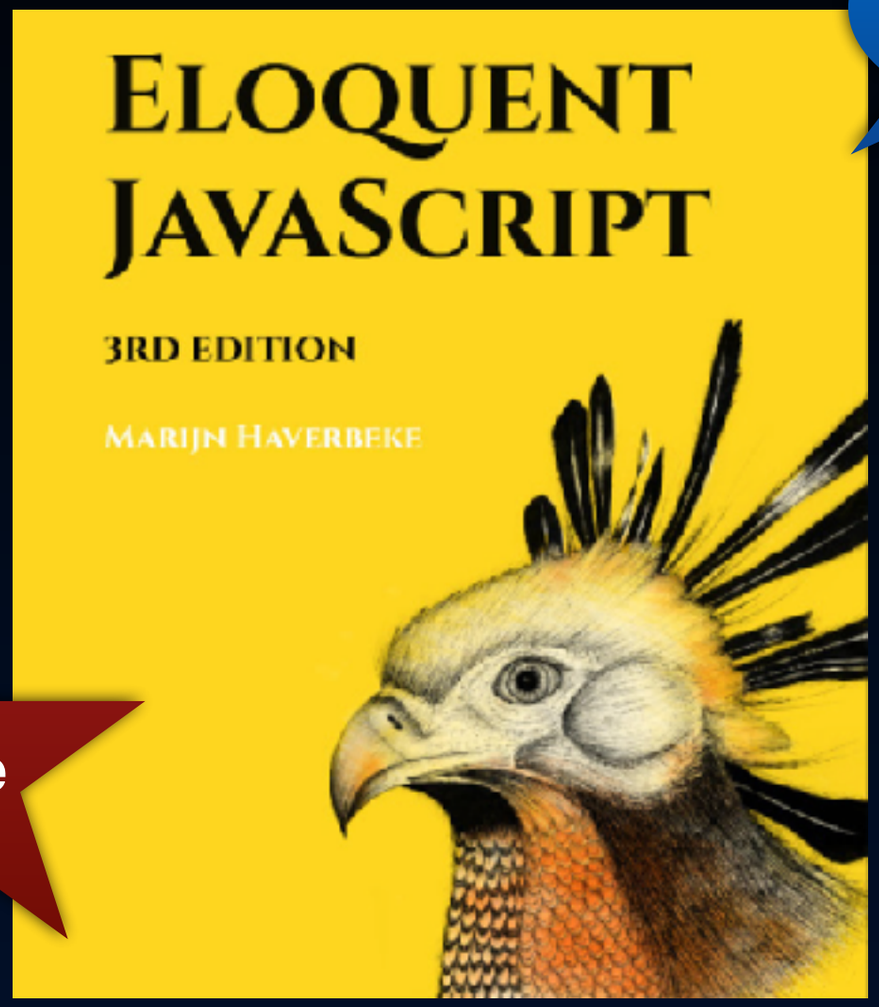
1 World of Game Development	10 Physics, Particles + Effects
2 Play a Game, Learn to Code 1	11 Midterm Review / Draft GDD
3 Play a Game, Learn to Code 2	12 Prefabs + Classes / Build Sys
4 Intro to JavaScript + Systems	13 Final Project: Design Game
5 Browser-Based Games	14 Adv Development Techniques
6 Working with Sprites + Controls	15 Build + Playtest Sprint 1
7 Level Maps, Atlases + Tiles	16 Build + Playtest Sprint 2
8 UI + Sound	17 Build + Playtest Sprint 3
9 Simulating the Physical World	18 Final Exam (online)

Get all of the details in the complete syllabus on Canvas.

**Weeks 11-17 include extra credit coverage of Unity3D.*

Textbook: JavaScript

Intro
+ Ch. 1



Free

Eloquent JavaScript (3nd)
Marijn Haverbeke
ISBN 978-1593279509

Textbook: Phaser Game Engine

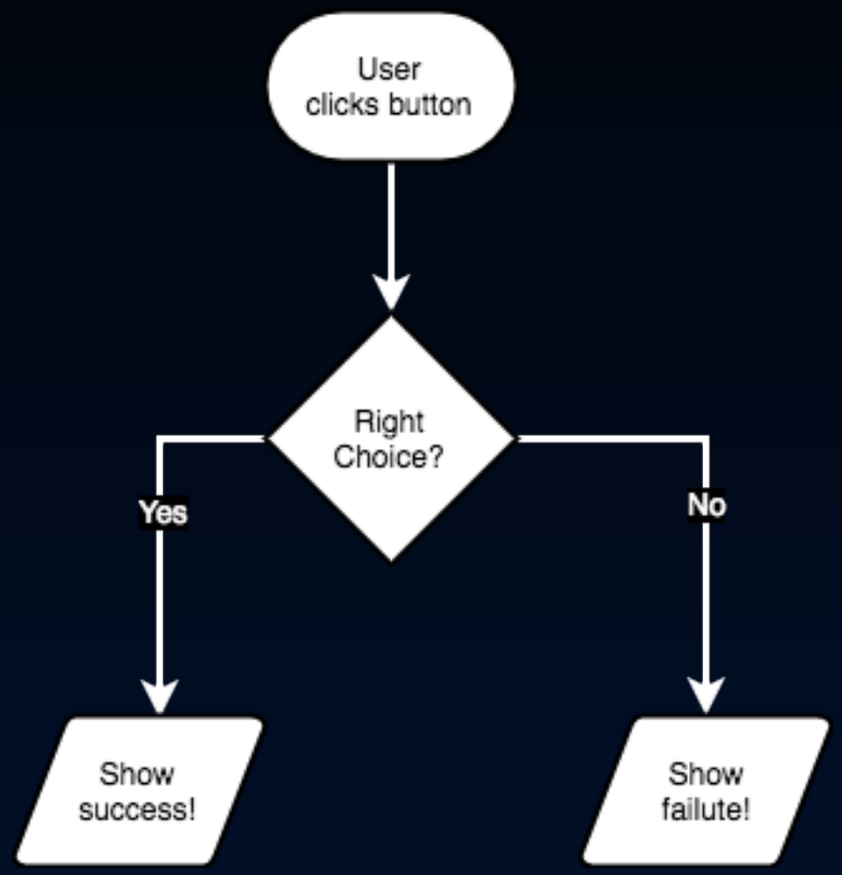
An Introduction to HTML5 Game Development with Phaser.JS

Travis Faas, CRC Press, 2016
ISBN 978-1-138-92184-9 print
ISBN 978-1-315-31921-6 ebook

Software This Week

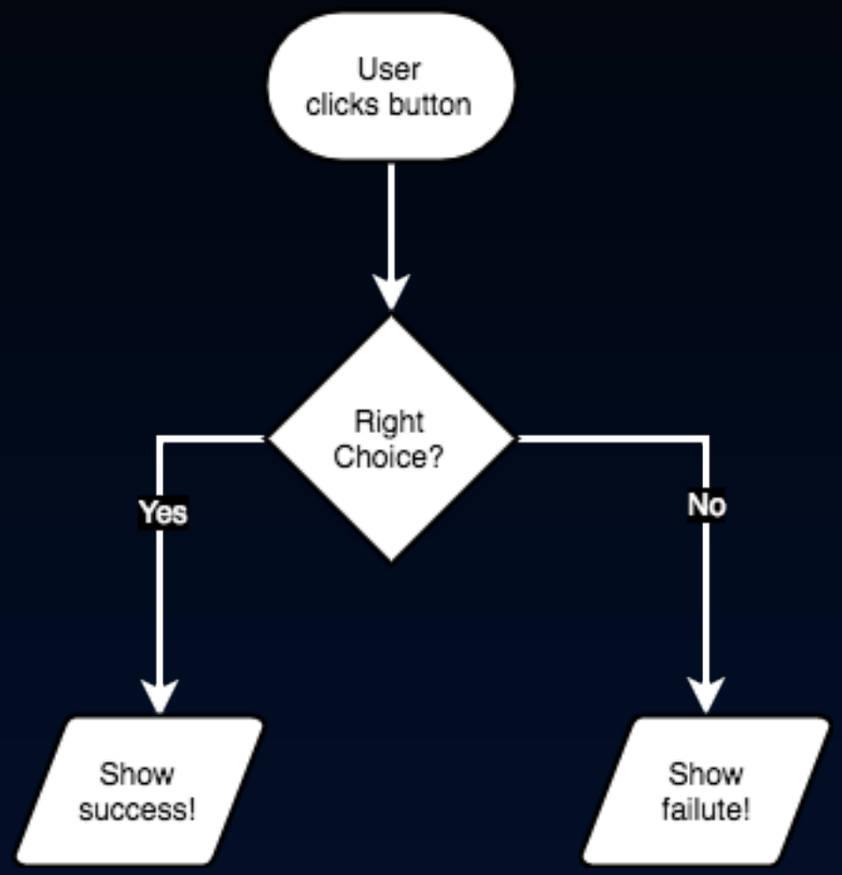
Text Editor + File Transfer	**Cloud9** (Browser-based, Mac + Win)
Web Browser	**Google Chrome** (Preferred for Cloud9)
Vector / Diagram Graphics Editor	**draw.io** (Browser-based, Mac + Win)
Bitmap Graphics Editor	**pixlr.com** (Browser-based, Mac + Win, Flash plugin)
PDF Reader	**Adobe Reader** (free)

Planning the Flow: Flowcharts to Map Systems



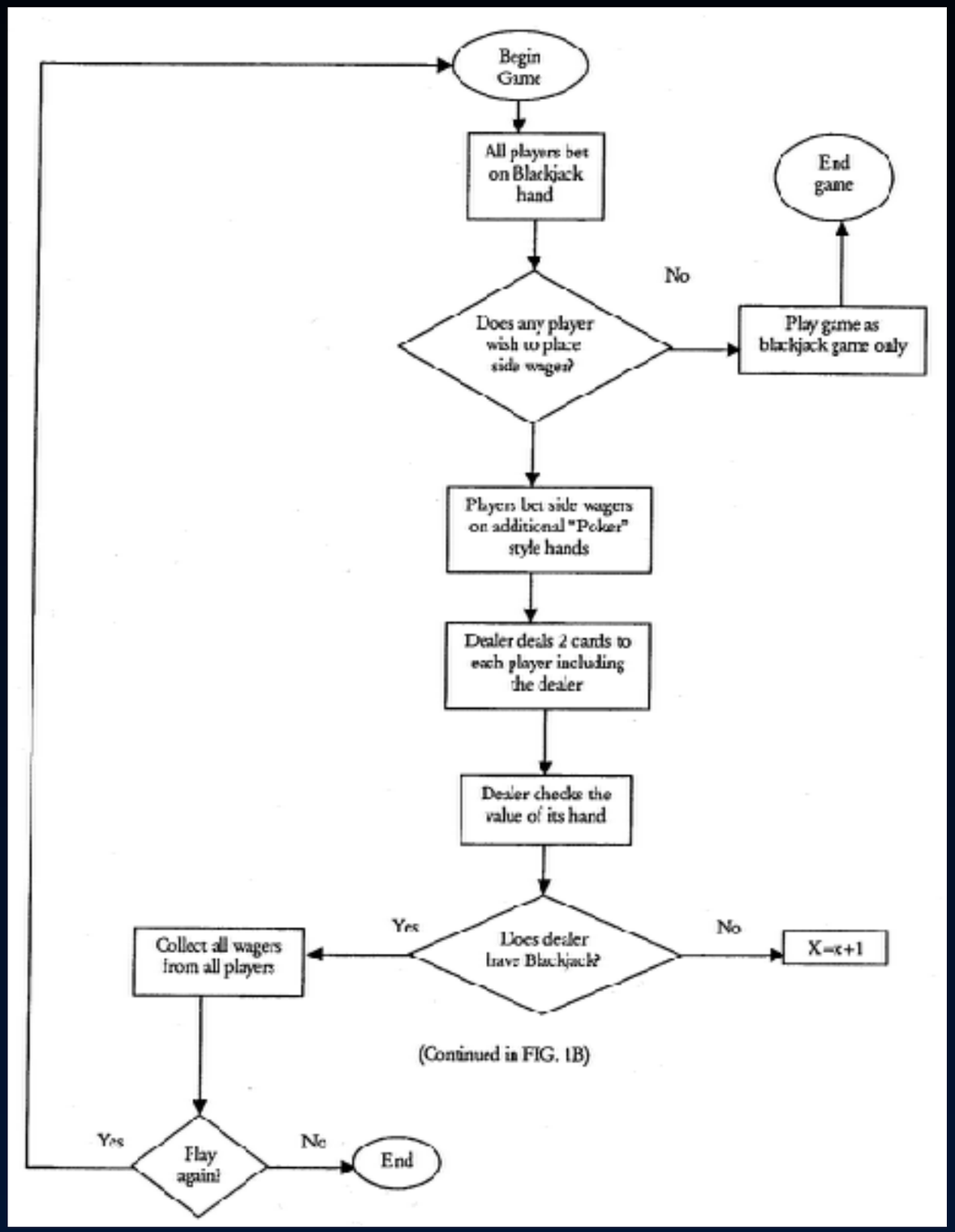
“A video game that is well-designed is like a system with interacting and interrelated parts that influence one another.” – Kelly Czarnecki

Planning the Flow: Flowcharts to Map Systems



draw.io offers browser-based diagram editing.

Planning the Flow: Flowcharts to Map Systems

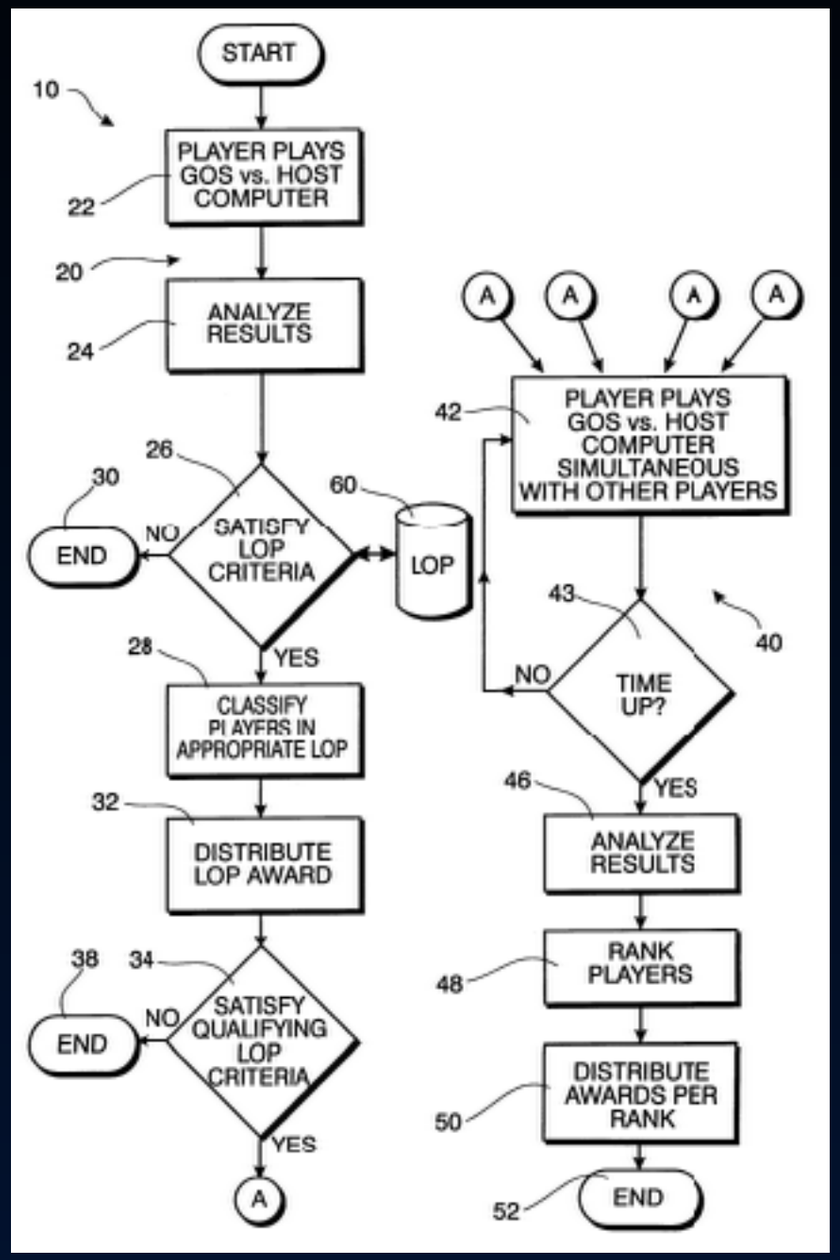


"Method of playing blackjack"
US Patent 20080067747 A1

https://www.google.com/patents/US20080067747

Gameplay is a defined system that can be mapped

Planning the Flow: Flowcharts to Map Systems



"Method for a game of skill tournament"
US Patent 6174237 B1

https://www.google.com/patents/US6174237

Gameplay is a defined system that can be mapped

Introduction to JavaScript

“It became clear around 2010 or so that JS and HTML would be the future of interactive content on the web.”

— *Travis Faas*

JavaScript Basics

1. **Instructions (Lexical Structures)**
2. **Comments**
3. **Values + Variables**
4. **Expressions + Operators**
5. **Statements + Control Structures**
6. **Functions**
7. **JavaScript, the Web + Cloud9**

JavaScript Basics

1. **Instructions (Lexical Structures)**
2. **Comments**
3. **Values + Variables**
4. **Expressions + Operators**
5. **Statements + Control Structures**
6. **Functions**
7. **JavaScript, the Web + Cloud9**

Instructions
(Lexical Structures)

break
case
catch
continue
debugger
default
delete
do
else
false
finally
for
function
if
in
instanceof
new
null
return
switch
this
throw
true
try
typeof
var
void
while
with

RESERVED WORDS

Instructions
(Lexical Structures)

let a = 0;

let a = 0

END OF LINE (OPTIONAL)

JavaScript Basics

1. **Instructions (Lexical Structures)**
2. **Comments**
3. **Values + Variables**
4. **Expressions + Operators**
5. **Statements + Control Structures**
6. **Functions**
7. **JavaScript, the Web + Cloud9**

Comments

```
// a single line comment
```

```
/*
a multi-line
comment
*/
```

COMMENTS

JavaScript Basics

1. **Instructions (Lexical Structures)**
2. **Comments**
3. **Values + Variables**
4. **Expressions + Operators**
5. **Statements + Control Structures**
6. **Functions**
7. **JavaScript, the Web + Cloud9**

Values + Variables

Number
String
Boolean

Null
Undefined

Symbol

PRIMITIVE TYPES AND OBJECTS

Values + Variables

Number
String
Boolean

Array Objects

Object Values

Null
Undefined

Symbol

PRIMITIVE TYPES AND OBJECTS

Values + Variables

```
const a = 15.67;
```

```
let my_Name = 'Frank';
```

```
var my_secret = true;
```

NUMBERS, STRINGS + BOOLEANS

Values + Variables

only available in ES6

```
const a = 15.67;
```

```
let my_Name = 'Frank';
```

```
var my_secret = true;
```

NUMBERS, STRINGS + BOOLEANS

Values + Variables

```
let a = Math.round( 0.6 );
```

```
let b = Math.random( );
```

MATH OBJECT FOR ARITHMETIC

Values + Variables

```
let q = [ 'Zero', 'One', 'Two' ];
```

ARRAYS A.K.A. LISTS OF VALUES

Values + Variables

```
{
  sky: true
}
```

OBJECT VALUES

Values + Variables

```
{
   sky: true
}
```

property name

property value

OBJECT VALUES

Values + Variables

```
let world = {
   sky: true
};
```

OBJECT VALUES

Values + Variables

```
let world = {
    sky: true,
    land: true
};
```

OBJECT VALUES

Values + Variables

```
let world = {
        sky: true,
        land: true,
        hours: 24,
        sea : {
            color: "blue",
            actions: {
                wave: function() {
                    // do something
                }
            },
            temp: 72.5
        }
}
```

property with object value

OBJECT VALUES

Values + Variables

```
let world = {
        sky: true,
        land: true,
        hours: 24,
        sea : {
            color: "blue",
            actions: {
                wave: function() {
                    // do something
                }
            },
            temp: 72.5
        }
}
```

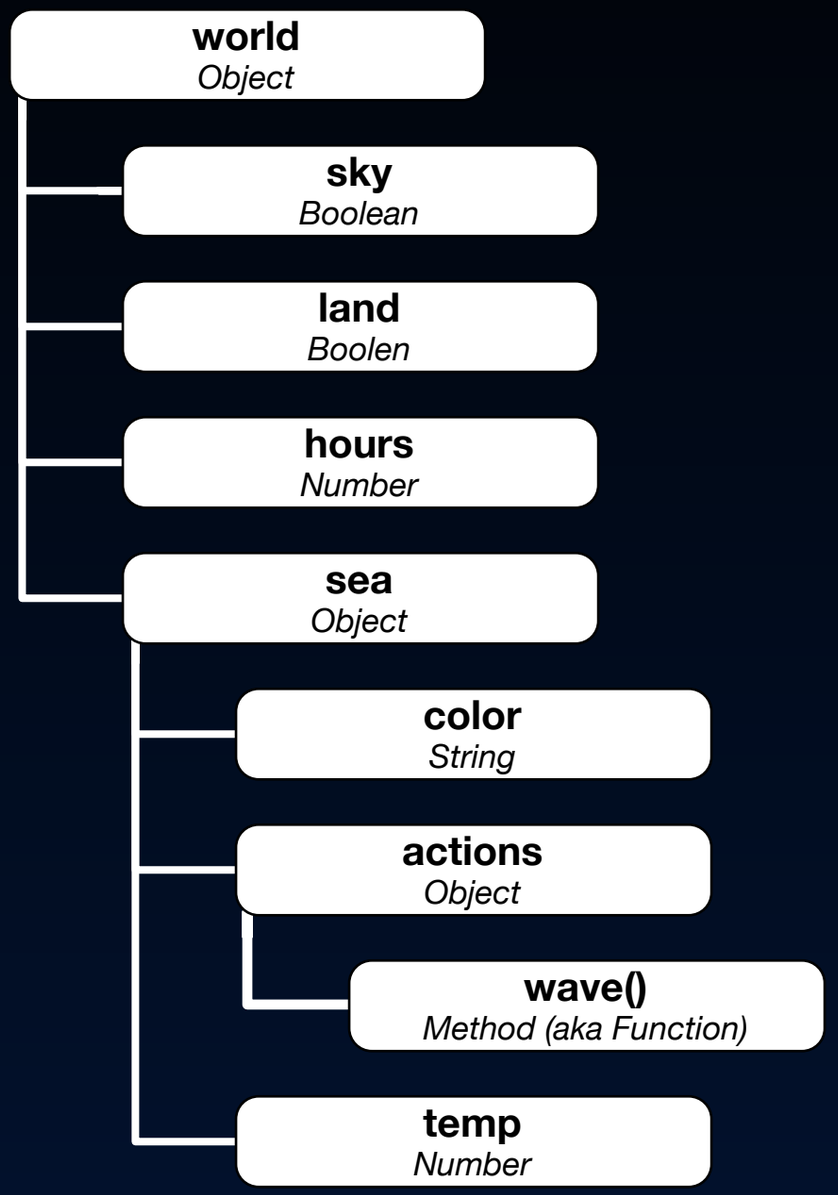
property with object value

property with function value called method

OBJECT VALUES

Values + Variables

```
let world = {
        sky: true,
        land: true,
        hours: 24,
        sea : {
            color: "blue",
            actions: {
                wave: function() {
                    // do something
                }
            },
            temp: 72.5
        }
}
```



OBJECT VALUES AS STRUCTURES

Values + Variables

```
let world = {
        sky: true,
        land: true,
        hours: 24,
        sea : {
            color: "blue",
            actions: {
                wave: function() {
                    // do something
                }
            },
            temp: 72.5
        }
}
```

world.sky

world.sea.color

world.sea.actions.wave()

OBJECT VALUES + DOT NOTATION

JavaScript Basics

1. **Instructions (Lexical Structures)**
2. **Comments**
3. **Values + Variables**
4. **Expressions + Operators**
5. **Statements + Control Structures**
6. **Functions**
7. **JavaScript, the Web + Cloud9**

Expressions + Operators

```
let a = ( 4 + 6 ) * 200;
```

EXPRESSIONS EVALUATE TO A VALUE

Expressions + Operators

++ increment	=== equals
-- decrement	!== not equals
- subtract	< less than
+ add or concat	> greater than
* multiply	<= lt or equal
/ divide	>= gt or equal
! not	= assignment

COMMON OPERATORS

JavaScript Basics

1. **Instructions (Lexical Structures)**
2. **Comments**
3. **Values + Variables**
4. **Expressions + Operators**
5. **Statements + Control Structures**
6. **Functions**
7. **JavaScript, the Web + Cloud9**

Statements + Control Structures

```
var y = 965;
let x = 100.25;
const z = 12;
```

only available in ES6

```
function my_guy()  {
    // code block
}
```

DECLARATION STATEMENTS

Statements + Control Structures

```
if ( x === 1 ) {
    // block of code if true
} else {
    // block of code if false
}
```

CONDITIONAL STATEMENTS

Statements + Control Structures

```
switch ( x ) {
    case 1:
        // do if x is 1
        break;
    default:
        // do if nothing else
}
```

CONDITIONAL STATEMENTS

Statements + Control Structures

```
while ( c < 10 ) {
    c = c + 2;
}
```

LOOPS

Statements + Control Structures

```
var i;
for ( i = 1;  i <= 10;  i++ ) {
    alert( i );
}
```

LOOPS

JavaScript Basics

1. **Instructions (Lexical Structures)**
2. **Comments**
3. **Values + Variables**
4. **Expressions + Operators**
5. **Statements + Control Structures**
6. **Functions**
7. **JavaScript, the Web + Cloud9**

Functions

```
function sneeze( x )  {
    let y;
    if ( x < 100 ) {
        y = x * 2;
    } else {
        y = x * 3;
    }
    return y;
}
```

ACCEPT PARAMETERS + RETURN VALUES

Functions

```
alert ( sneeze( 50 ) );
```

```
function sneeze( x )  {
    let y;
    if ( x < 100 ) {
        y = x * 2;
    } else {
        y = x * 3;
    }
    return y;
}
```

WHAT NUMBER WILL BE DISPLAYED?

JavaScript Basics

1. **Instructions (Lexical Structures)**
2. **Comments**
3. **Values + Variables**
4. **Expressions + Operators**
5. **Statements + Control Structures**
6. **Functions**
7. **JavaScript, the Web + Cloud9**

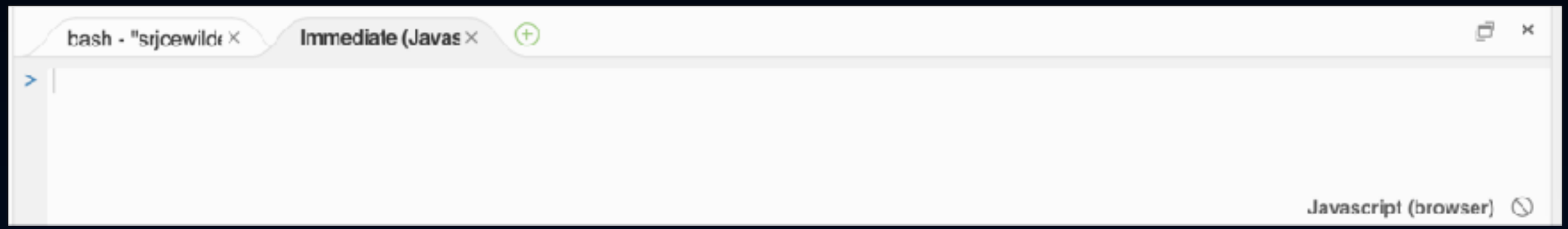
JavaScript in Web Browsers

```
document.write( 'Frank' );
```

```
window.location.href =
'http://santarosa.edu';
```

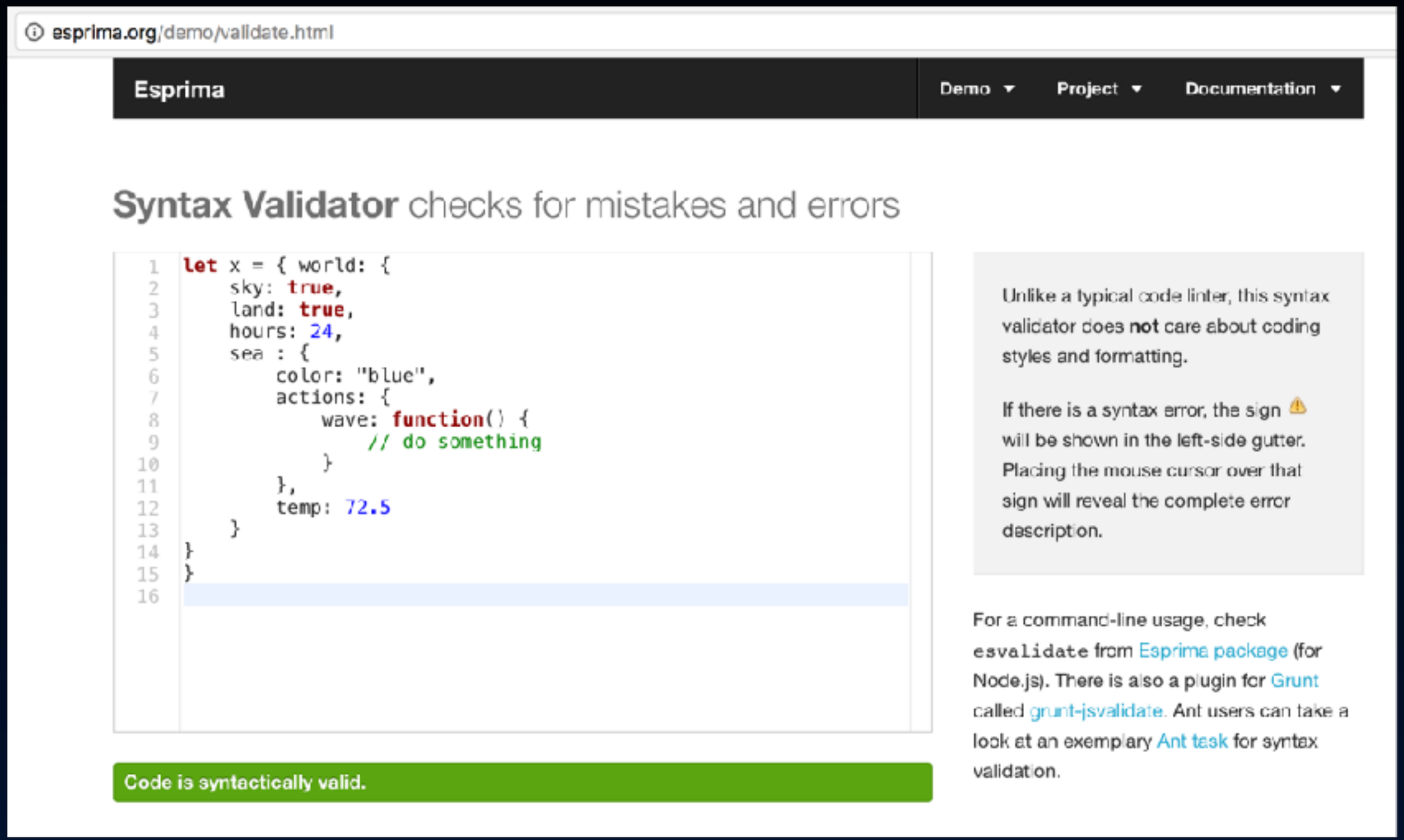
DOT NOTATION + THE OBJECT MODEL

JavaScript Coding in Cloud9



Cloud9 offers a built-in JavaScript console, where we can test out code in real time.

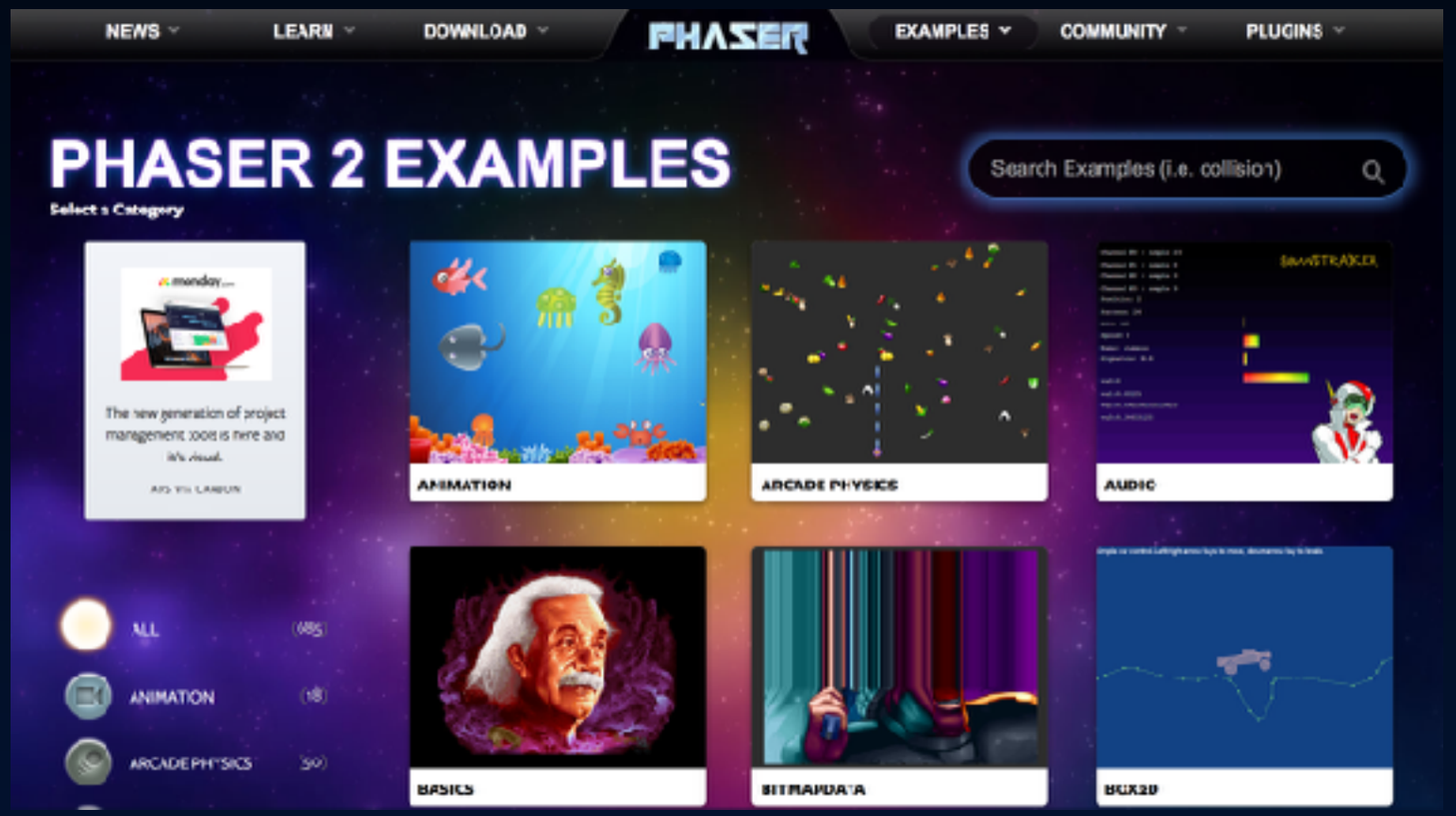
Validating JavaScript Code



esprima.org offers live JavaScript code validation.

Check Out Phaser CE / 2.x

“Take a peek at our JavaScript game engine: Phaser CE / 2.x”



https://phaser.io/examples

What to Do Next

- **Reading + Watching + Doing**
  - Read *Eloquent JavaScript*, Introduction + Chapter 1
  - Review JavaScript for Cats online tutorial
    - *http://srjc.ethan.com/js4cats/*
- **Homework**
  - **Assignment 4: First Game**
  - Homework due to Canvas by **11:59pm Thurs 9/20**
- **Canvas Site**
  - All materials available there
  - ***canvas.santarosa.edu/courses/33387***