



# Catching Pokemon with Phaser

Devoxx4Kids at Devoxx US

March 19th 2017



Wojtek Wilk

[wilk.wojtek@gmail.com](mailto:wilk.wojtek@gmail.com)

<https://github.com/wwilk>



# Why JavaScript + HTML?

- simple
- about 95% of websites use JavaScript
- about 75% of websites use HTML
- easier to start than compiled languages (like C++/Java)

# Variables

- containers holding some information/  
value
- make code more clear and readable
- can be constant



# Variables

```
var caughtPokemons = 0;  
  
console.log(caughtPokemons);  
  
caughtPokemons++;  
  
console.log(caughtPokemons);
```

# Exercise

- Create variable "a" and assign value 5 to it, then create variable "b" and assign 6 to it, and then create variable "c" and assign sum of a and b to it



# Functions

- blocks of code
- reusability
- avoid duplications in code

# Functions

```
5 * (10 * 1.8 + 32)
```

```
function celsiusToFahrenheit(celsius) {  
    return celsius * 1.8 + 32;  
}
```

```
5 * celsiusToFahrenheit(10);
```



# Functions

```
function createPokemon(name) {  
  // a lot of operations that create a Pokemon  
  //and set its name, initial position, velocity etc  
};  
  
createPokemon("pikachu");  
createPokemon("charmander");  
  
...
```

# Functions

Create a function of signature:

```
function mileToKilometer(miles) {  
...  
}
```

that will convert given miles into  
kilometers (multiply by 1.609344)

# Conditionals

```
var condition = true;  
  
if (condition) {  
    console.log('yes');  
} else {  
    console.log('no');  
}
```

# Exercise

Create a function of signature:

```
function isPositiveNumber(num) {
```

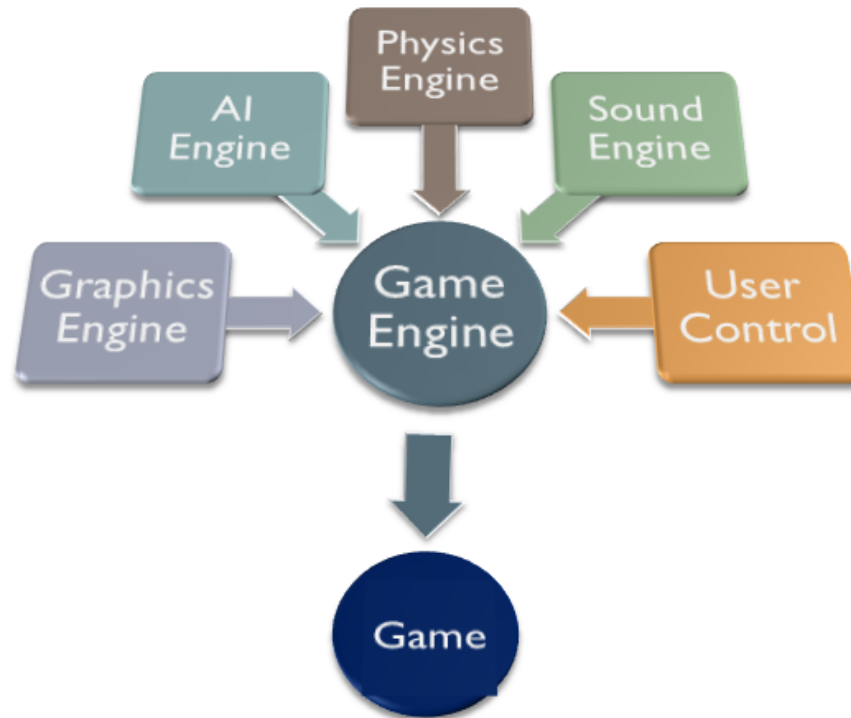
```
...
```

```
}
```

that will print "yes" if number is positive  
and "no" if negative or zero

# Game Engine

## Game Engine





# Why Game engine/Phaser?

- no point in reinventing the wheel
- quick and visible effect



# Phaser game structure

```
// Initialize Phaser, and create a 800x480px game
var game = new Phaser.Game(800, 480, Phaser.AUTO, 'gameContainer');

// Create our 'main' state that will contain the game
var mainState = {

    preload: function() {
        // called once after beginning - the best place to load images
    },

    create: function() {
        // called once after beginning, later than preload
        // here we initialize all objects that are needed at the start of the game
    },

    update: function() {
        // called 30 times per second
    }

};

// Add and start the 'main' state to start the game
game.state.add('main', mainState);
game.state.start('main');
```



# Possible changes

- background image
- pokemon image
- initial position of Pokemon
- velocity of Pokemon
- text / text color
- not destroying Pokemon after click
- frequency of Pokemon appearance