

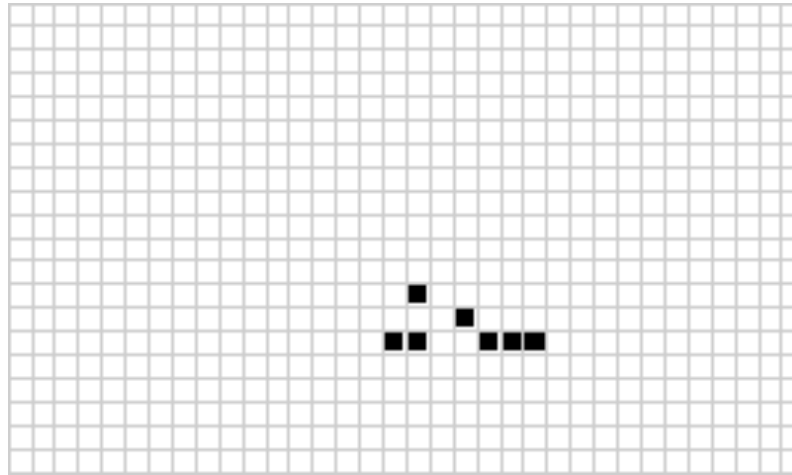
# GAME OF LIFE

*Skynet alpha?*



ERICK OH

# GAME OF LIFE



[HTTP://WWW.ERICWEISSTEIN.COM/ENCYCLOPEDIAS/LIFE/ACORN.HTML](http://www.ericweisstein.com/encyclopedias/life/acorn.html)

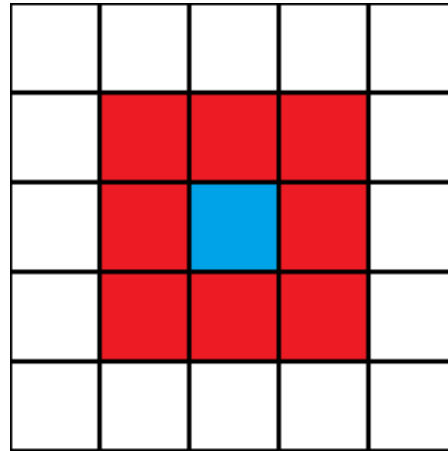
# GAME OF LIFE

- “Zero-player” game (see *animation*)
- Rooted in Von Neumann’s quest for artificial/simulated life
- Created by Jon Conway in 1970
- Sparked niche field: **cellular automaton**
- Simple rules can produce complex behavior

# RULES

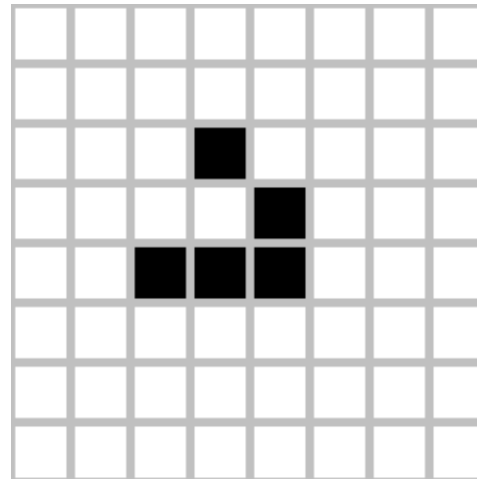
- 2D grid of cells that are currently on or off (dead or alive)
- Each step, grid updates all-at-once
- Currently alive cell
  - “Underpopulation”: dies given fewer than 2 live neighbors
  - “Overcrowding”: dies given greater than 3 live neighbors
  - Otherwise, lives on
- Currently dead cell
  - “Birth”: comes to life given exactly 3 live neighbors
  - Otherwise, remains dead

# NEIGHBORS



[HTTPS://EN.WIKIPEDIA.ORG/WIKI/CELLULAR\\_AUTOMATON](https://en.wikipedia.org/wiki/Cellular_automaton)

# GAME OF LIFE



[HTTP://WWW.HOMESCHOOLSON.COM/CONWAYS-GAME-OF-LIFE/](http://www.homeschoolson.com/conways-game-of-life/)

# PAIRPROJECT

A SLIGHTLY LESS DIRECTIVE, LONGER AND MORE DIFFICULT WORKSHOP.

# MANIPULATING THE DOM

- Changing Attributes for Style
- Making Elements
- Putting them into the DOM
- Remove Elements



# CHANGING STYLE ATTRIBUTES

```
element.style.backgroundColor = "blue";
```

## CSS

background-color

border-radius

font-size

list-style-type

word-spacing

z-index

## JavaScript

backgroundColor

borderRadius

fontSize

listStyleType

wordSpacing

zIndex

## CHANGING CSS CLASSES

- *classList* is HTML5 way to modify which classes are on an

```
document.getElementById("MyElement").classList.add('class');  
document.getElementById("MyElement").classList.remove('class');  
if ( document.getElementById("MyElement").classList.contains('class') )  
document.getElementById("MyElement").classList.toggle('class');
```

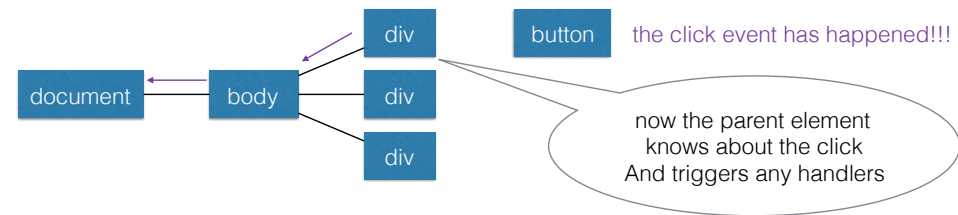
# EVENT HANDLERS

```
element.addEventListener('click', function(event) {  
  // Run this code on click  
});
```

- JS that handles things that happen in the DOM
- Event examples:
  - click
  - (form) submit
  - hover
  - mouseover

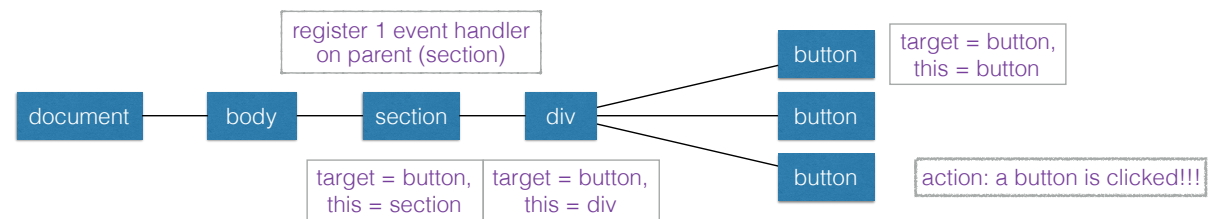
# EVENT PROPAGATION/BUBBLING

- An event is directed to its intended target
- If there is an event handler it is triggered
- From here, the **event** *bubbles* up to the containing elements
- This continues to the document element itself



# EVENT DELEGATION

- The process of using event propagation to handle events at a higher level in the DOM
- Allows for a single event listener



# WORKSHOP TIME