

# Is there life after code?



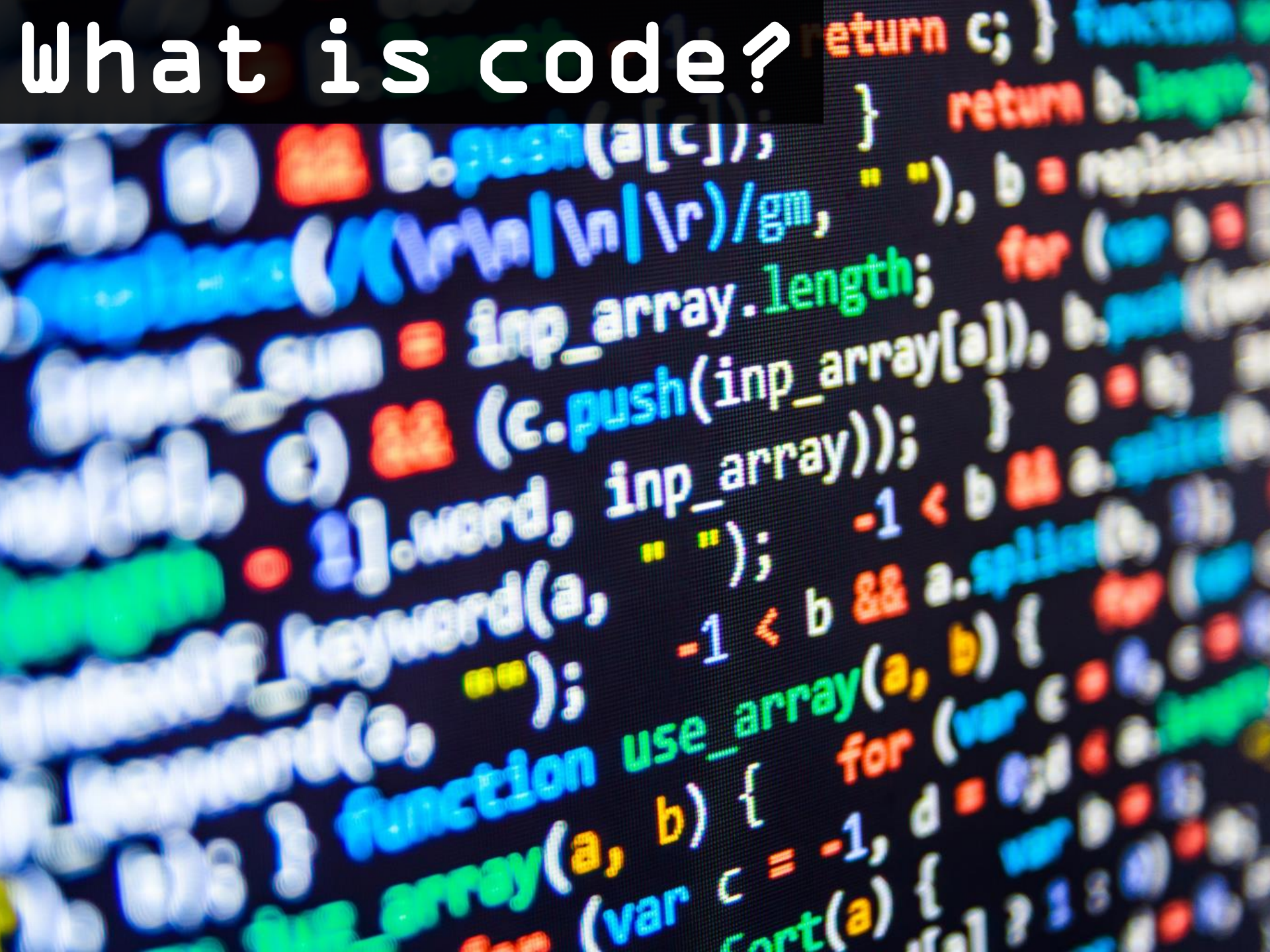
**BALL STATE  
UNIVERSITY**

*Computer Science Colloquium Series 2017-18:*

*Huseyin Ergin, PhD*

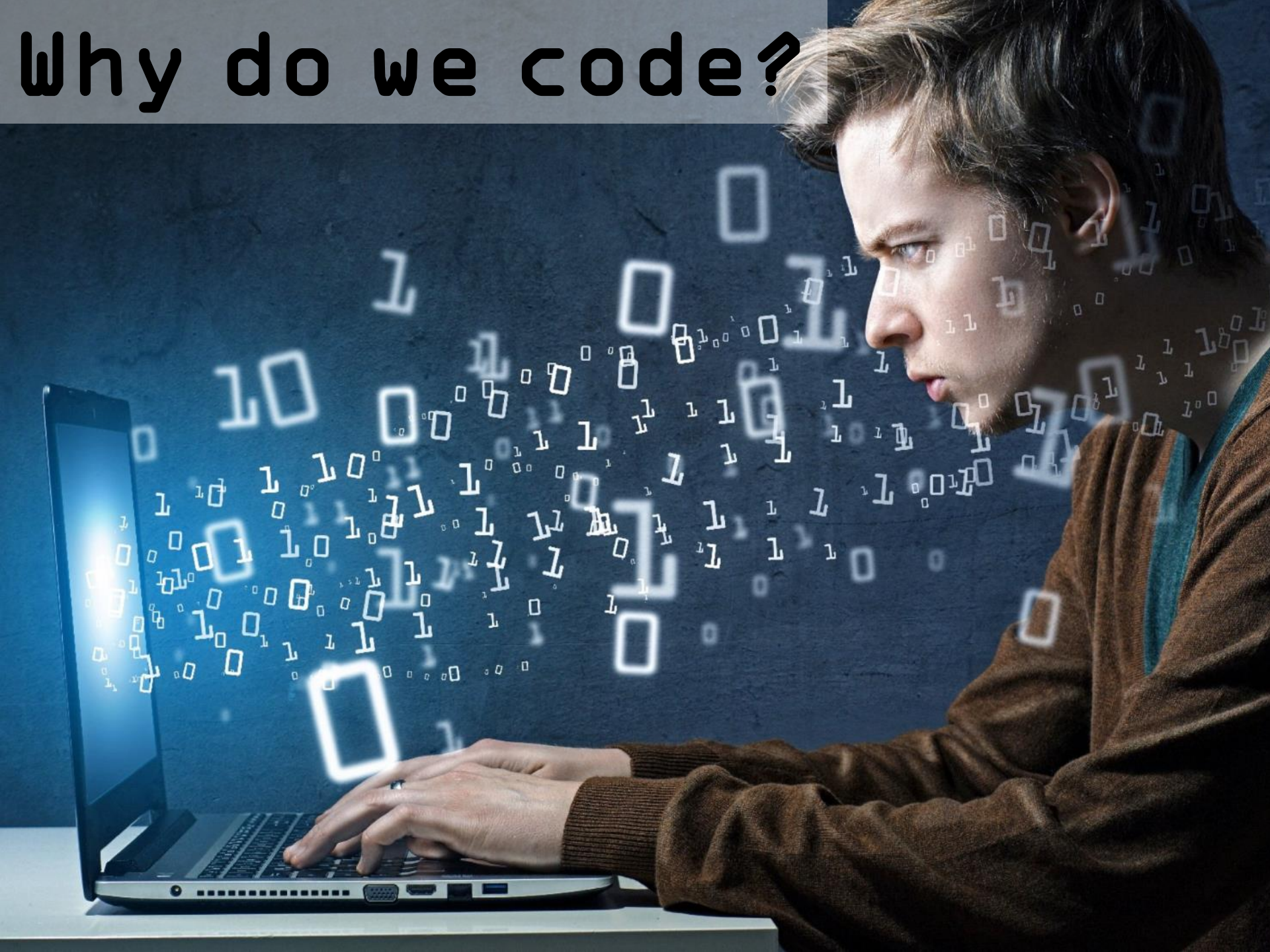


# What is code?





# Why do we code?





**We want to change the world!**



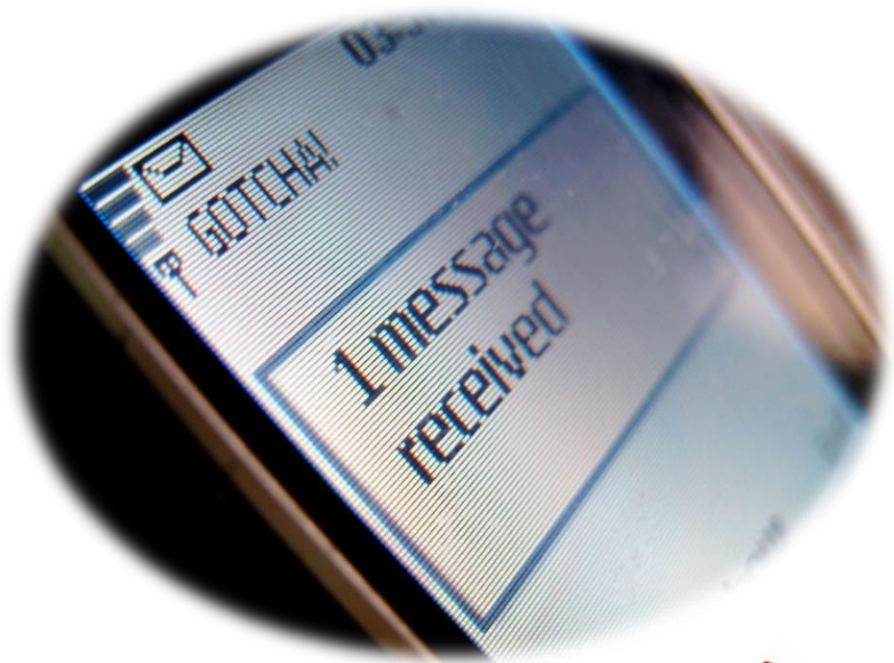




**NETFLIX**





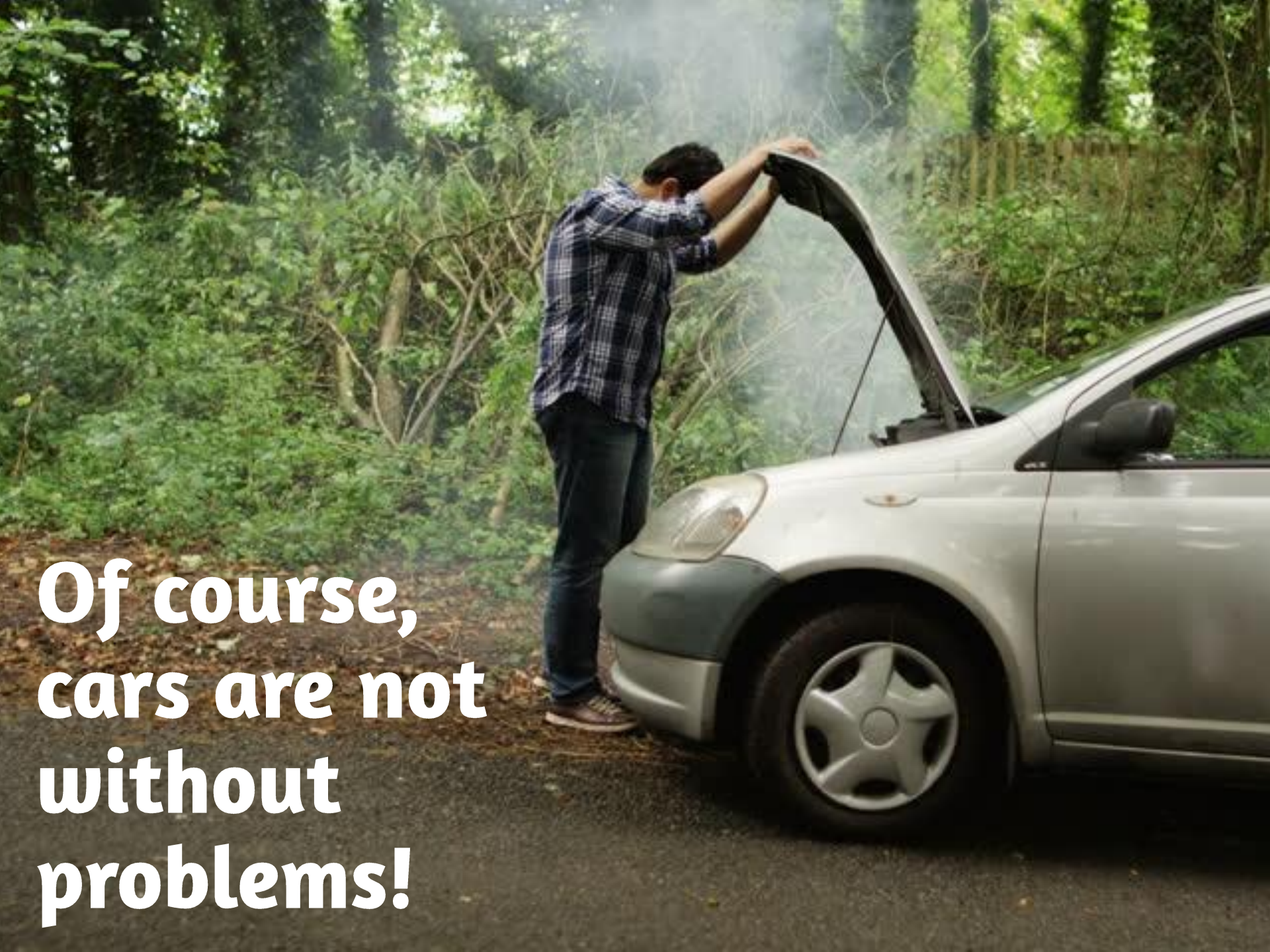












**Of course,  
cars are not  
without  
problems!**



**Also, planes have their own problems, too.**





Accidental

VS

**ESSENTIAL**







Tab



Caps  
Lock





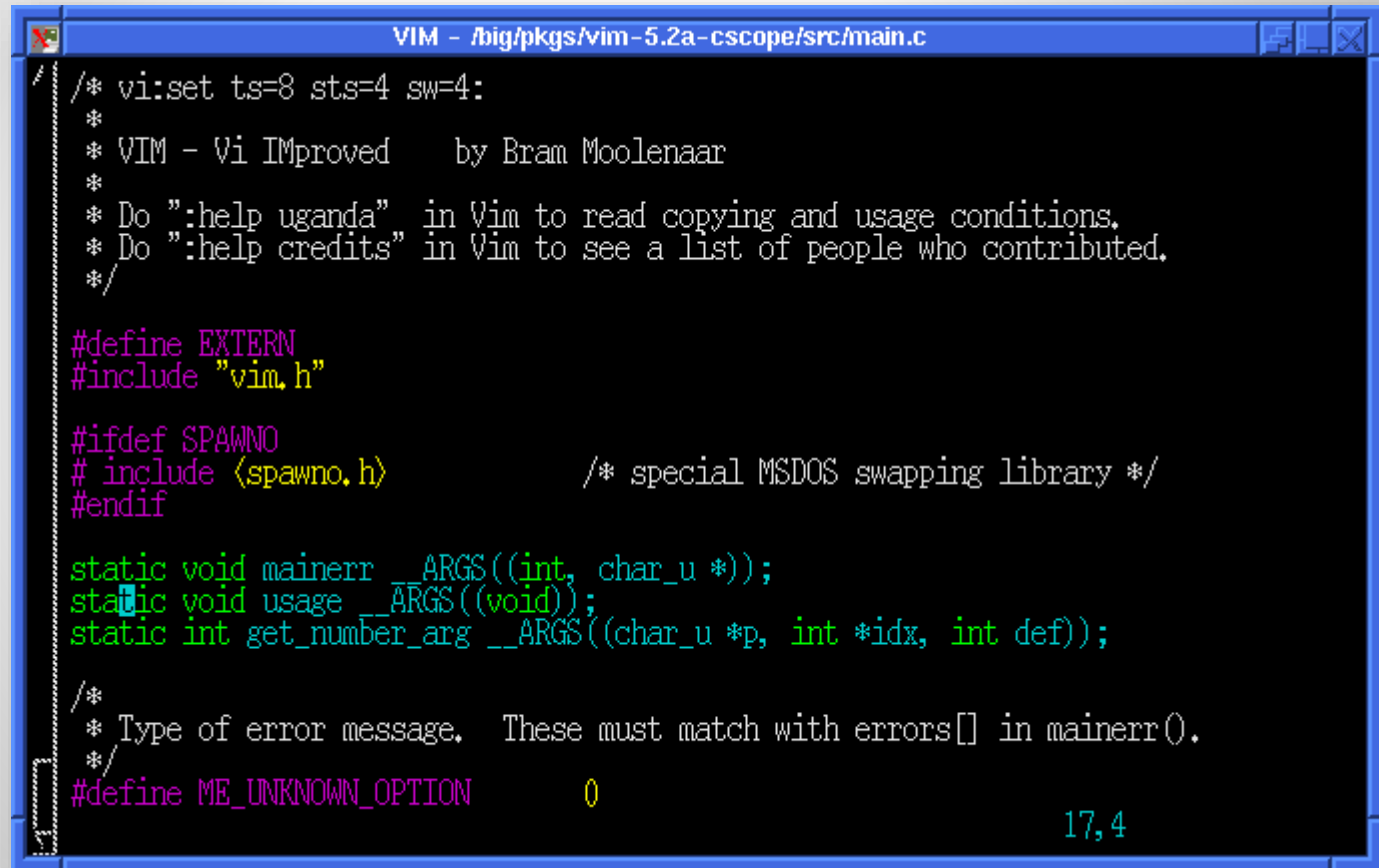


Some people, when confronted with a problem, think "I know, I'll use regular expressions." **Now they have two problems.**





# Coding in 80s & 90s



```
VIM - /big/pkgs/vim-5.2a-cscope/src/main.c
/* vi:set ts=8 sts=4 sw=4:
 *
 * VIM - Vi IMproved    by Bram Moolenaar
 *
 * Do ":help uganda"  in Vim to read copying and usage conditions.
 * Do ":help credits" in Vim to see a list of people who contributed.
 */

#define EXTERN
#include "vim.h"

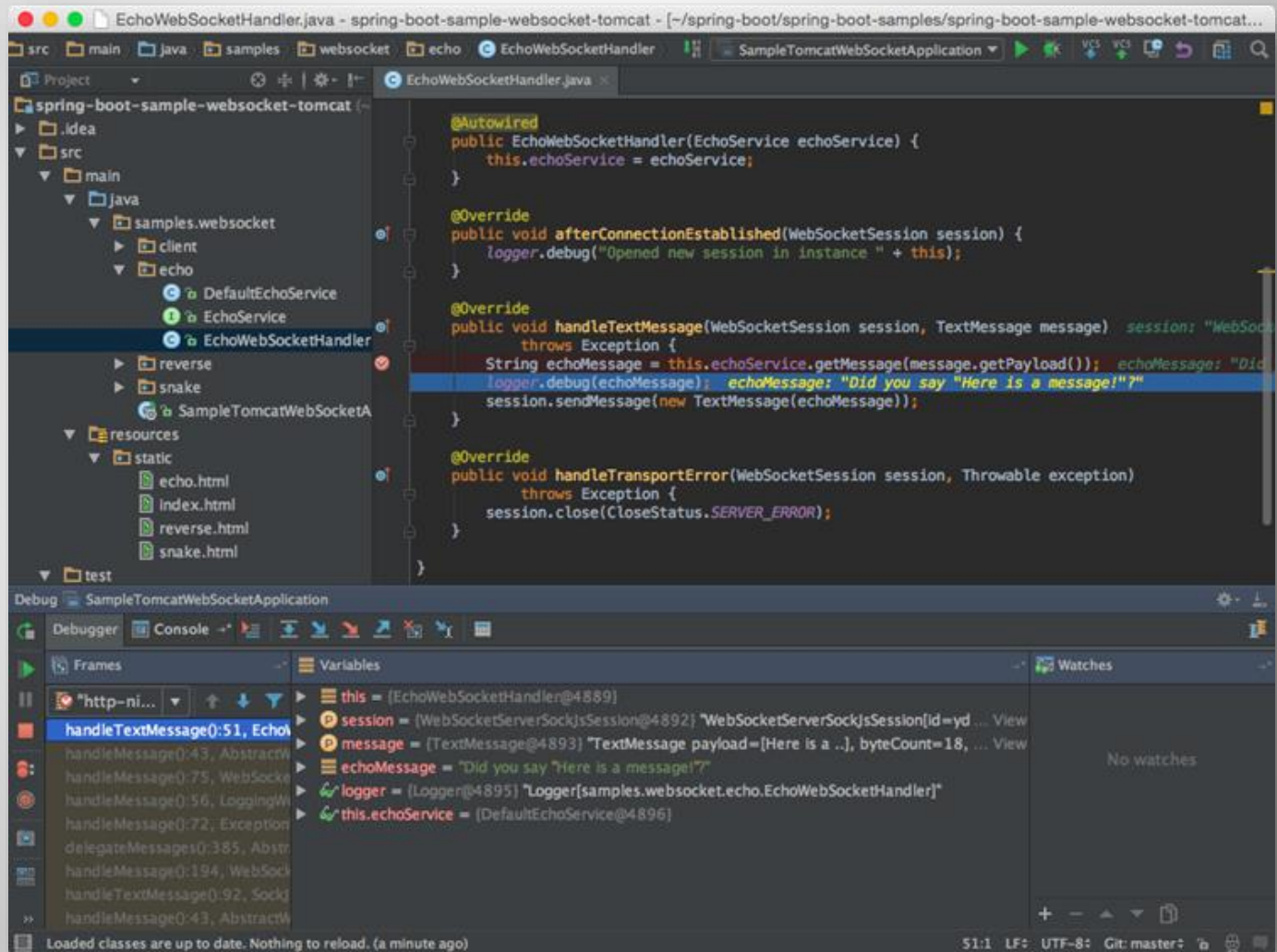
#ifdef SPAWNO
# include <spawn.h>          /* special MSDOS swapping library */
#endif

static void mainerr __ARGS((int, char_u *));
static void usage __ARGS((void));
static int get_number_arg __ARGS((char_u *p, int *idx, int def));

/*
 * Type of error message.  These must match with errors[] in mainerr().
 */
#define ME_UNKNOWN_OPTION      0

17,4
```

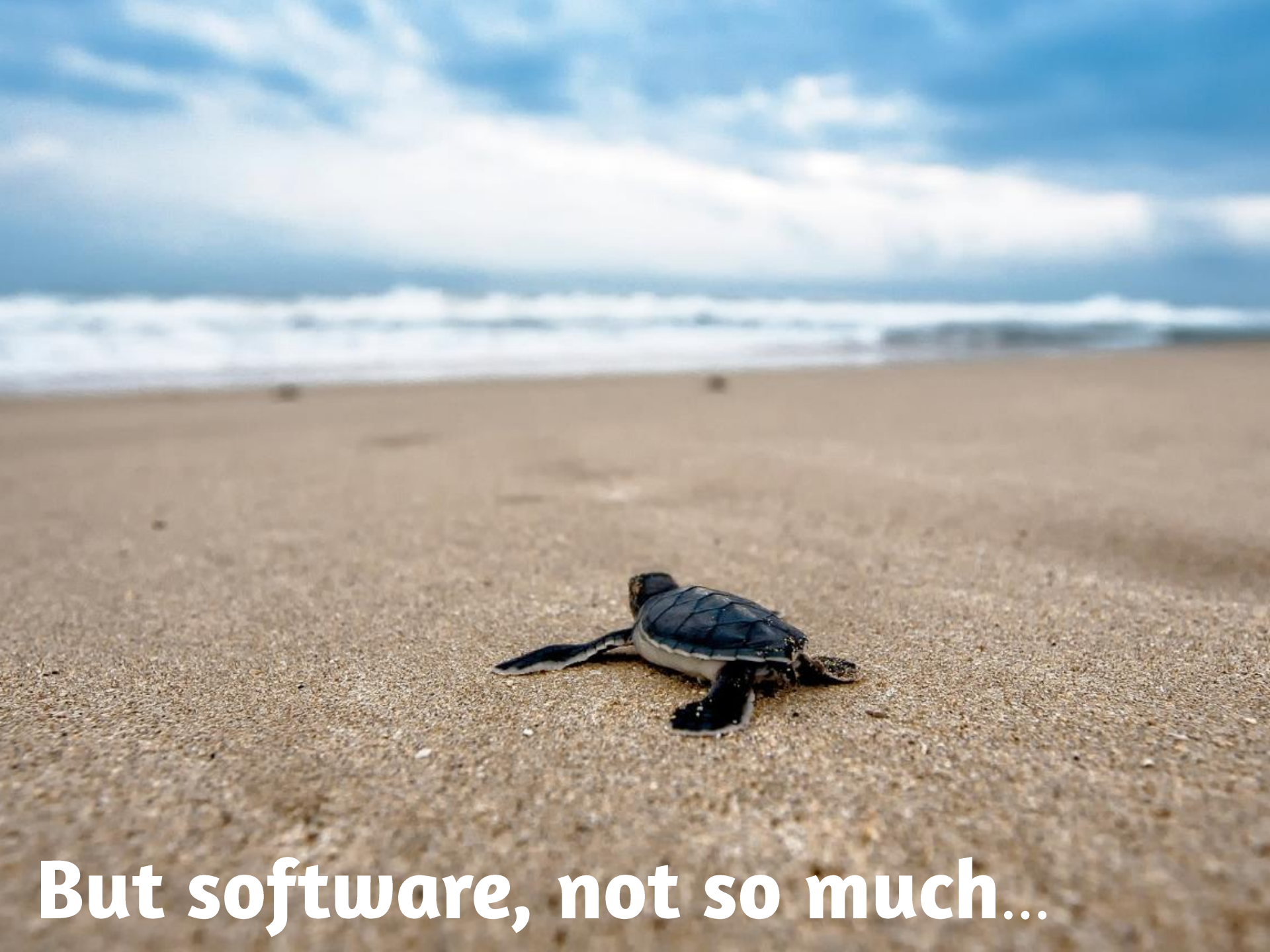
# CODING TODAY





# Hardware skyrocketed...





**But software, not so much...**



---

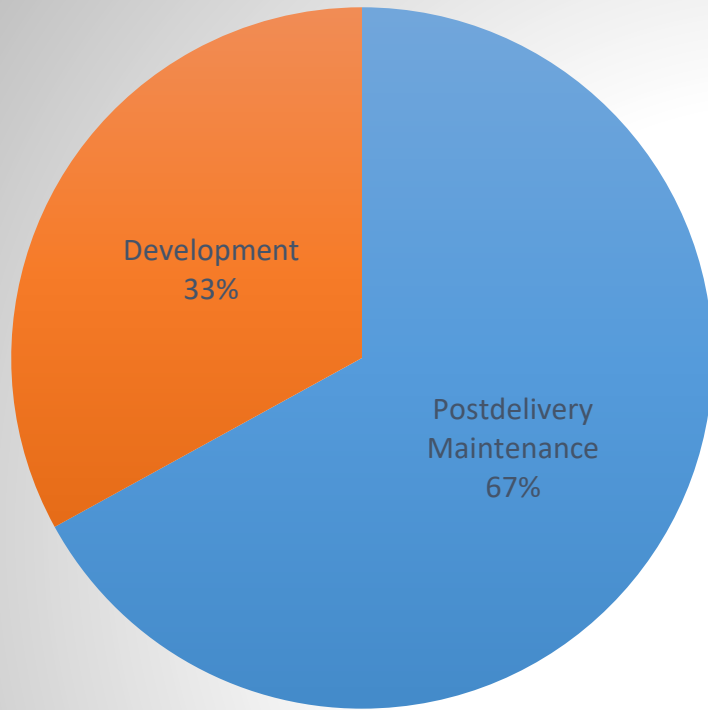
# stop coding

*(phrasal verb)*

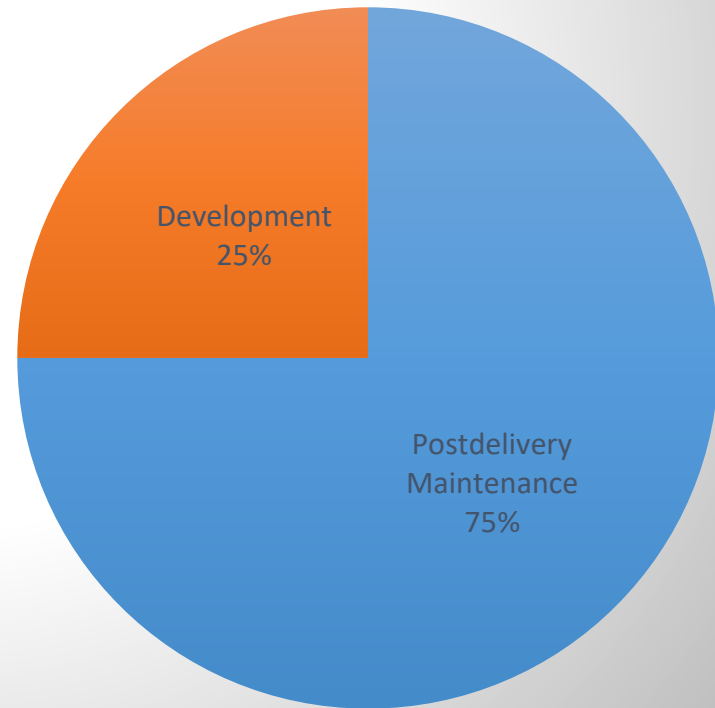
a mystical action that would stop all kinds of programming errors at once and for good.

---

## Between 1976-1981



## Between 1992-1998



# Costs\* ...





**Software aging...**

**Signs are saying us something...**





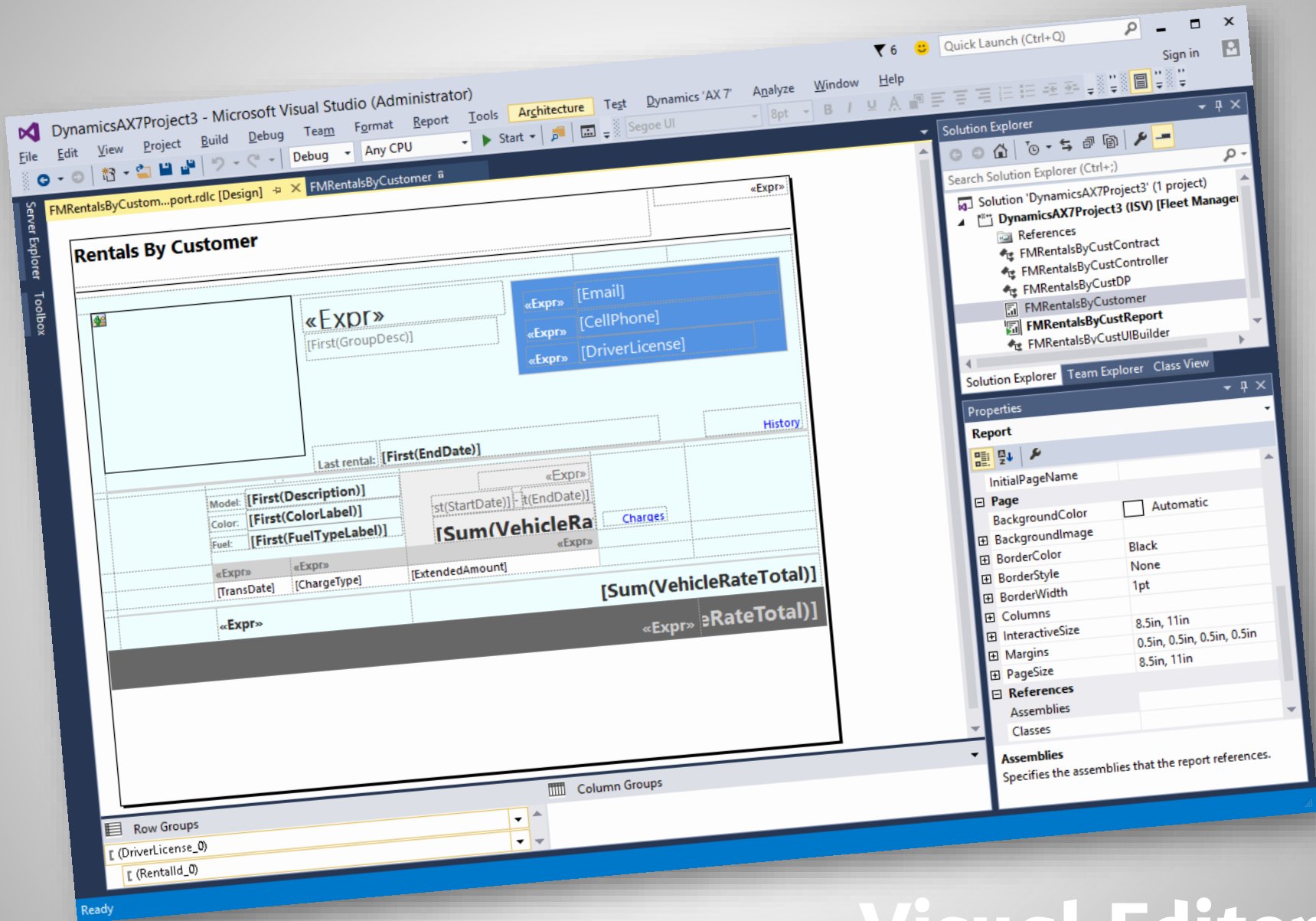


**We have options.**

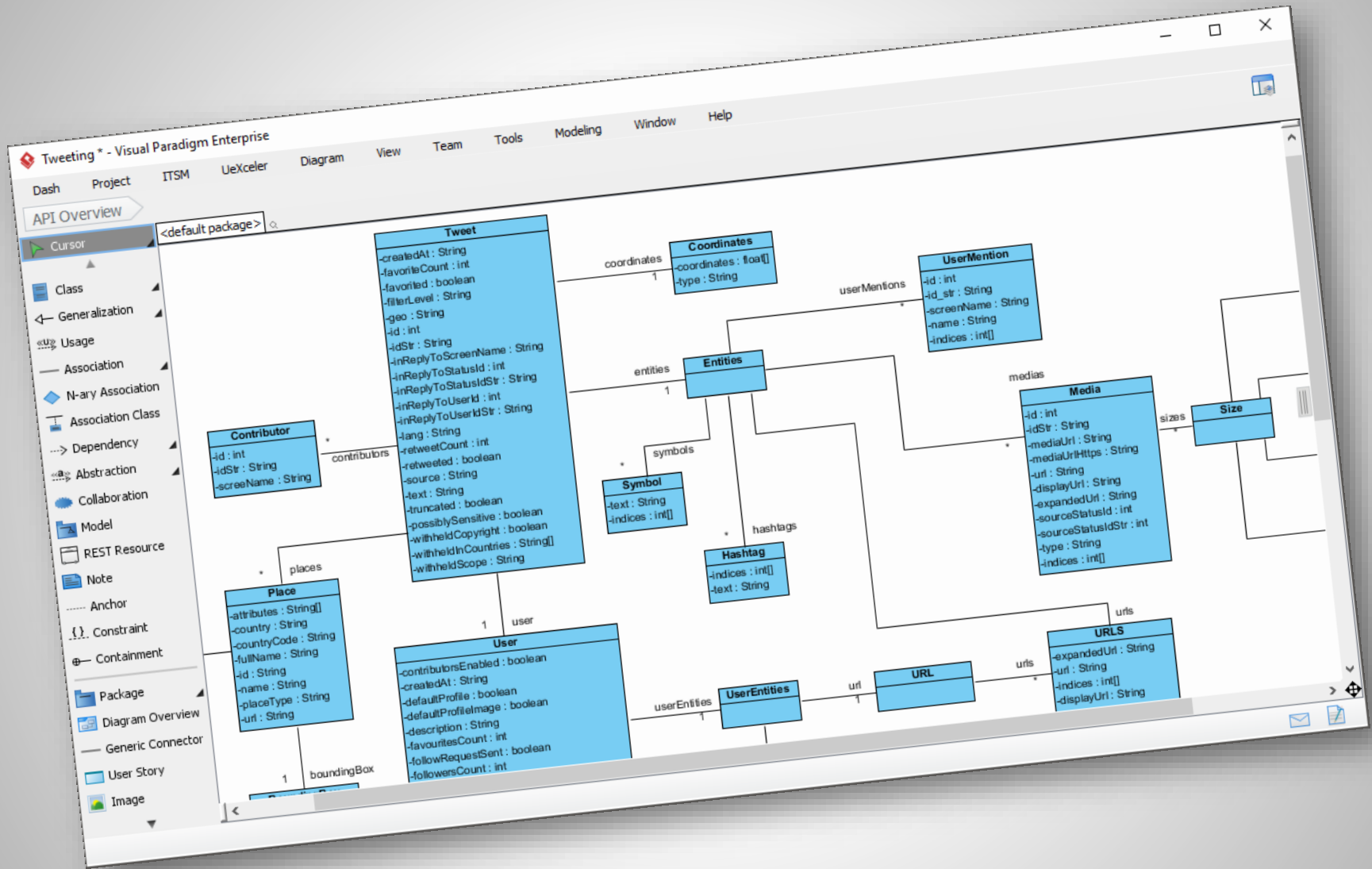
The image shows a programming environment with a block-based editor on the left and a text-based editor on the right. The block-based editor has a sidebar with categories: Logic, Loops, Math, Text, Lists, Color, Variables, and Functions. The code in the block editor consists of a 'do while' loop: 'set Count to 1', 'repeat while' (with 'Count <= 3'), 'do' containing 'print "Hello World!"' and 'set Count to' (with 'Count + 1'). The text editor shows the equivalent JavaScript code: 'Language: JavaScript', 'var Count;', 'Count = 1;', 'while (Count <= 3) {', ' window.alert('Hello World!');', ' Count = Count + 1;', '}'.

# Block-based languages





Visual Editors



# Unified Modeling Language

**What is  
beyond  
code?**







**Accomplishing its intended  
purpose successfully!**



**Can be useless according to the domain!**



**The right tool  
for this domain...**







**Time to  
think**

```
SELECT      EmployeeID, FirstName, LastName, HireDate
FROM        Employees
WHERE       HireDate BETWEEN '1-june-2012' AND '15-d
```

```
SELECT EmployeeID, FirstName, LastName, HireDate,
WHERE City IN ('Seattle', 'Tacoma', 'Redmond')
```

```
SELECT      EmployeeID, FirstName, LastName, HireDate
FROM        Employees
WHERE       HireDate NOT BETWEEN '1-june-2012' AND '
```



```
<!DOCTYPE HTML PUBLIC "-//W3C//
```

```
<html>
```

```
<head>
```

```
<title>Untitled Page</title>
```

```
<meta http-equiv="content
```

```
<meta http-equiv="content
```

```
</head>
```

```
<body onLoad="onLoadFun
```

```
<div class="outerDiv"
```

```
<div class="content"
```

```
<script type="text/
```

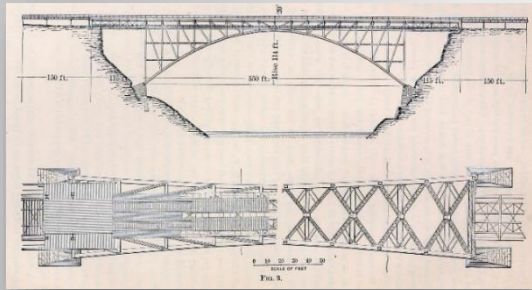
```
on getSequ
```

```
= {}
```

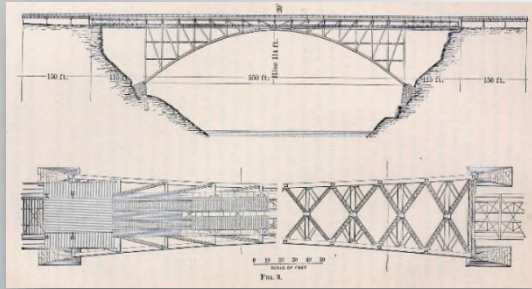




# To build a bridge...



**We first build models.**



**Why do we model?**







**Wouldn't  
it be nice?**



**And end up with  
the original product?**





*Software has the property that allows us to directly evolve models into full-fledged implementations without changing the engineering medium, tools, or methods!*



*Bran Selic*



# How does this help?

**Give me a modeling environment with resistors, capacitors, transistors, wires...**

The screenshot shows a circuit simulation software window titled "Circuits1.circuit: a circuit". On the left is a "Toolbox" with categories "Circuit Diagrams" and "General". The "Circuit Diagrams" category includes: Pointer, Capacitor, Comment Conn..., Comment, Junction, Resistor, Transistor, and Wire. The "General" category is currently selected.

The main workspace displays a circuit diagram with the following components and labels:

- Resistors: RB2, RC2, RC1, RB1, Rout
- Capacitors: C2, C1
- Transistors: T2, T1
- Output terminal: out
- Power rails: -ve (top), +ve (bottom)

Annotations in red-bordered boxes:

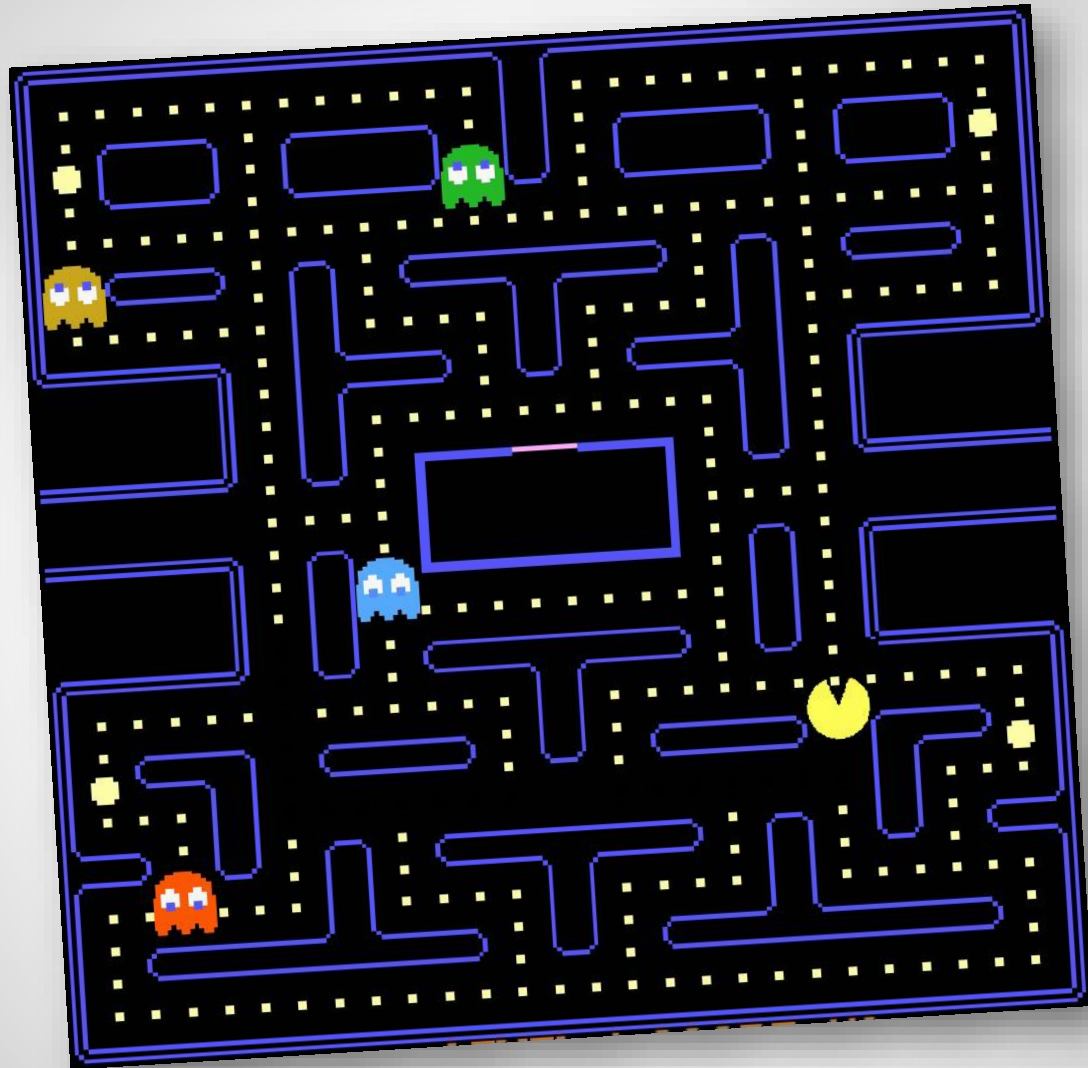
- Bottom-left: "You can use a non-polar capacitor if it's available in the size you want, but if you do use a polar one, it should be this way around." (with a dotted line pointing to capacitor C2)
- Bottom-center: "Double-click components to rotate them. Drag, CTRL+Drop components to copy. To tidy alignment, select a row or column of components that are approximately aligned, then click 'Align' in the menu bar." (with a blue highlight on transistor T1)
- Bottom-right: "Period =  $0.7 \cdot C \cdot RB$   
RC about  $RB/10$   
Rout  $\gg$  RC"

On the right side, there is a "Validate" button and a component count table:

Capacitor	2
Resistor	5
Transistor	2

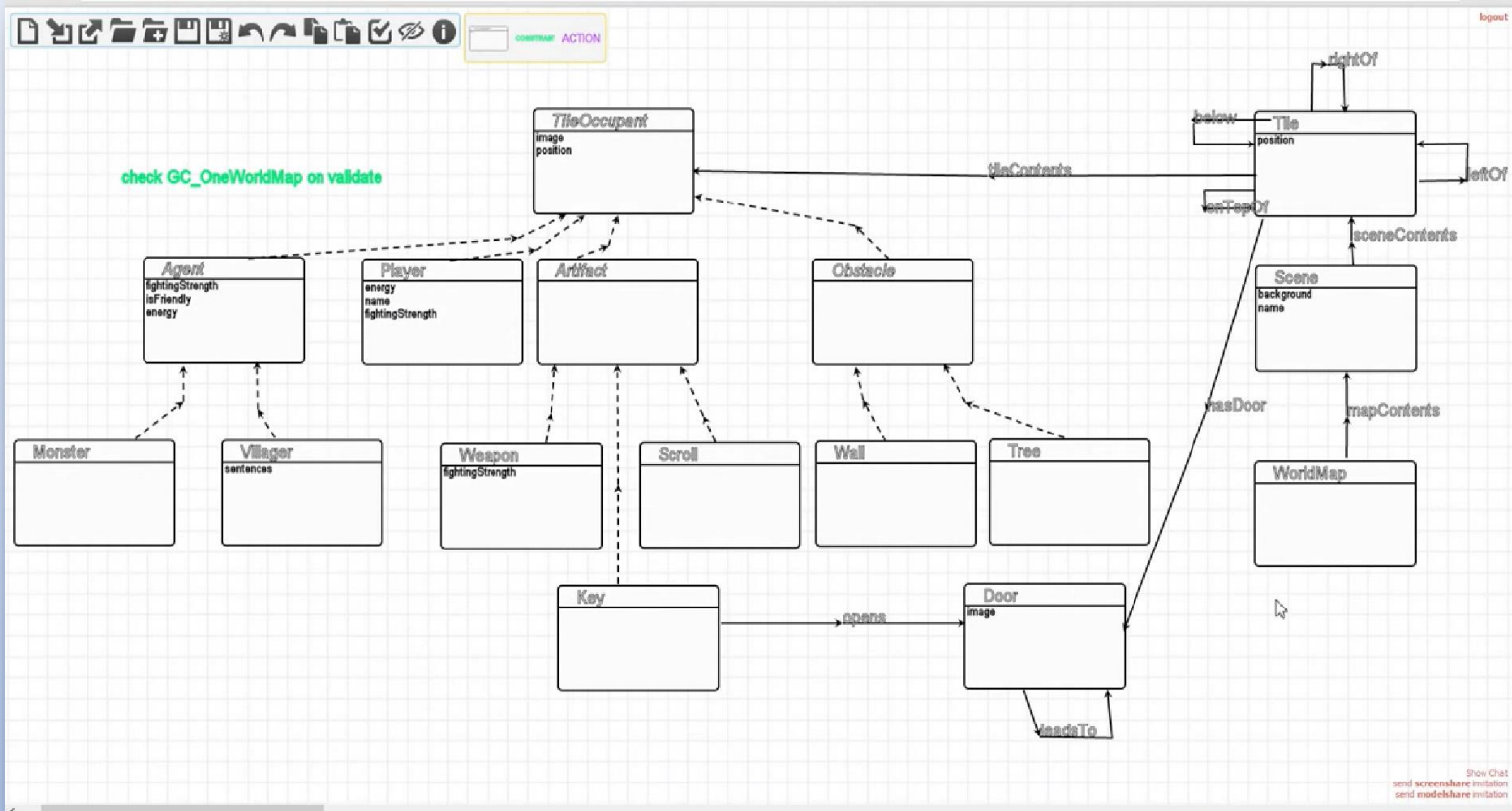
Give me a PACMAN environment with a pacman, dots (foods), ghosts, walls...

Electrical circuit environment

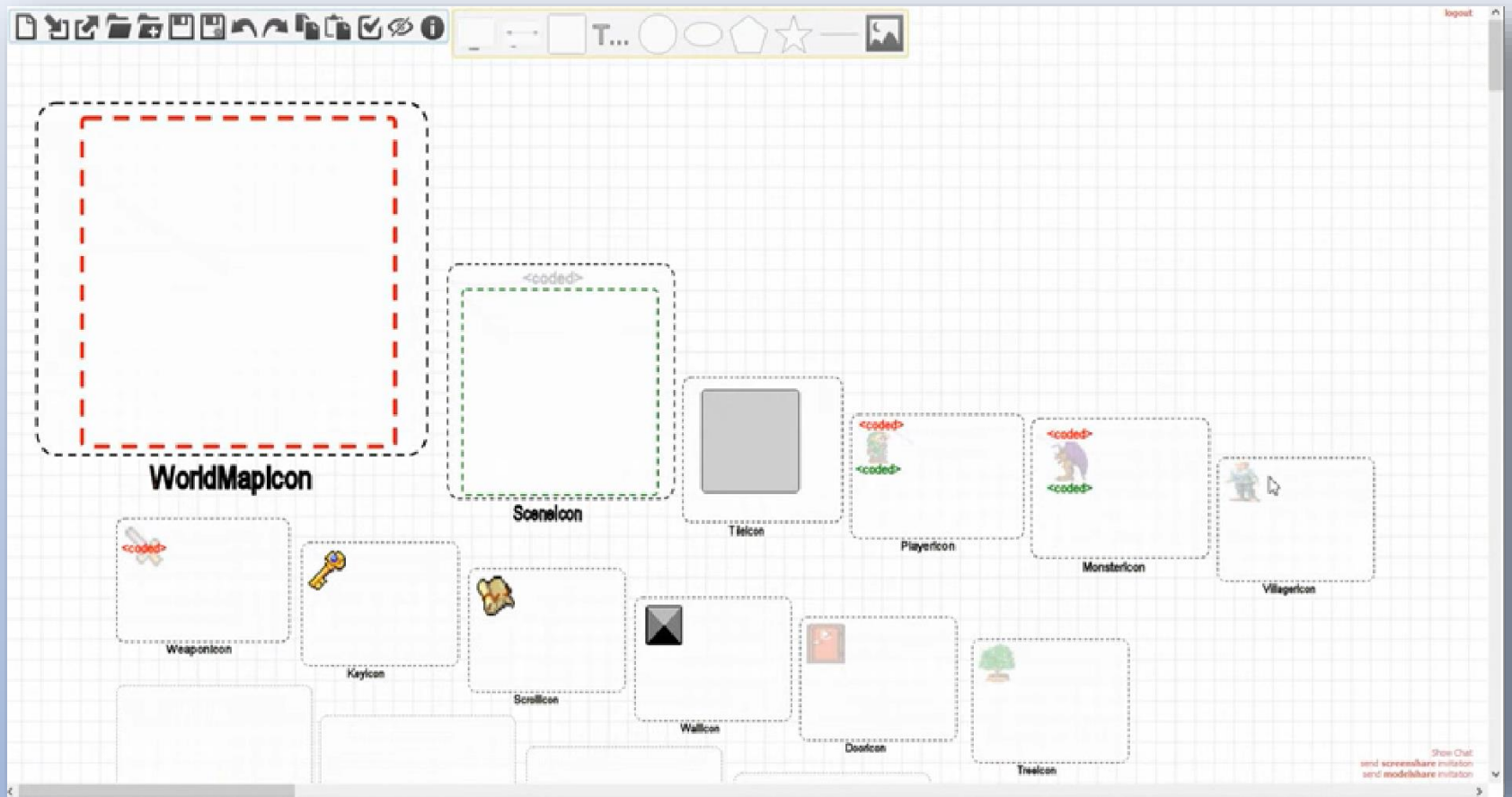


**PACMAN environment**

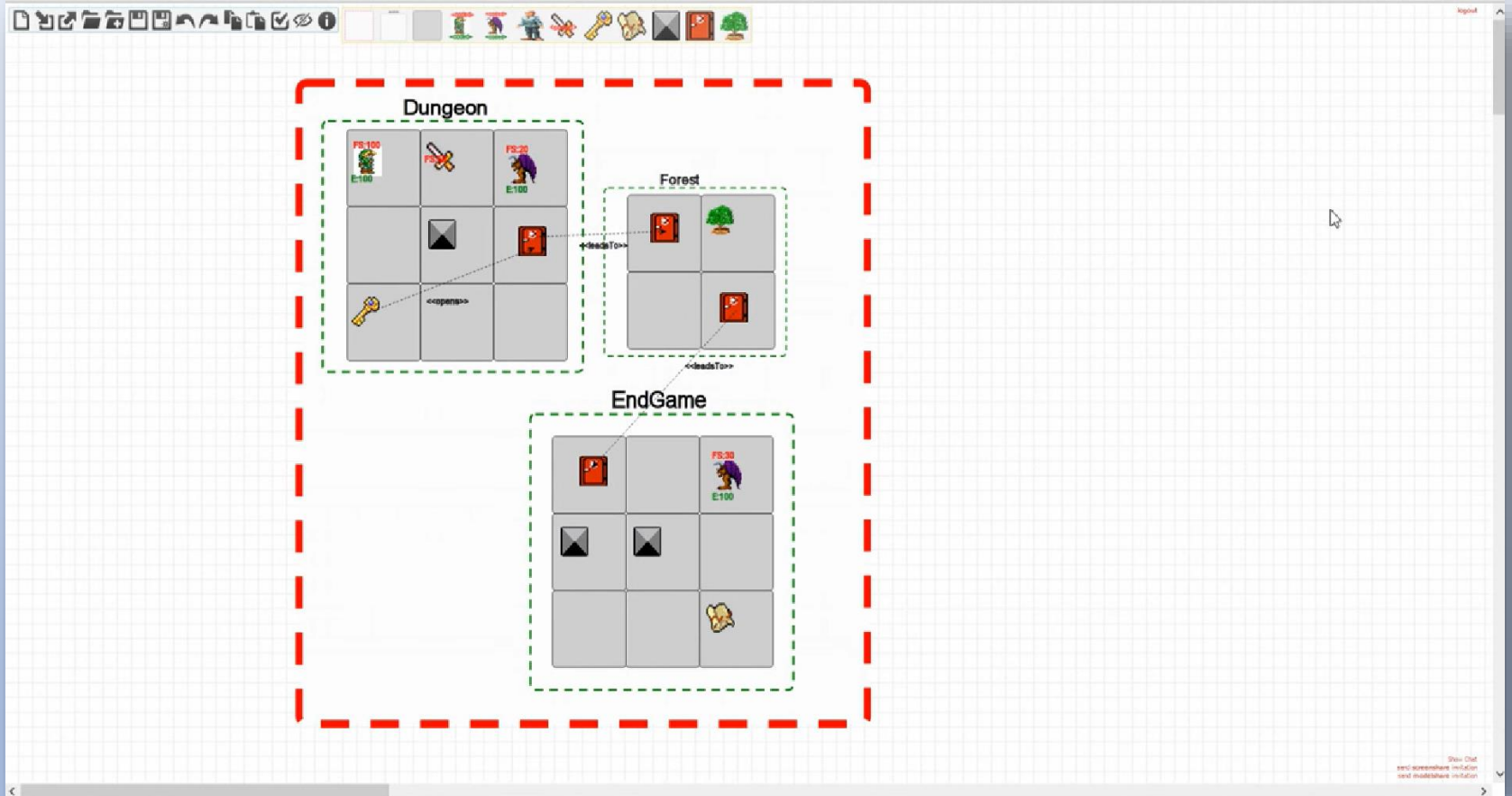




Define elements

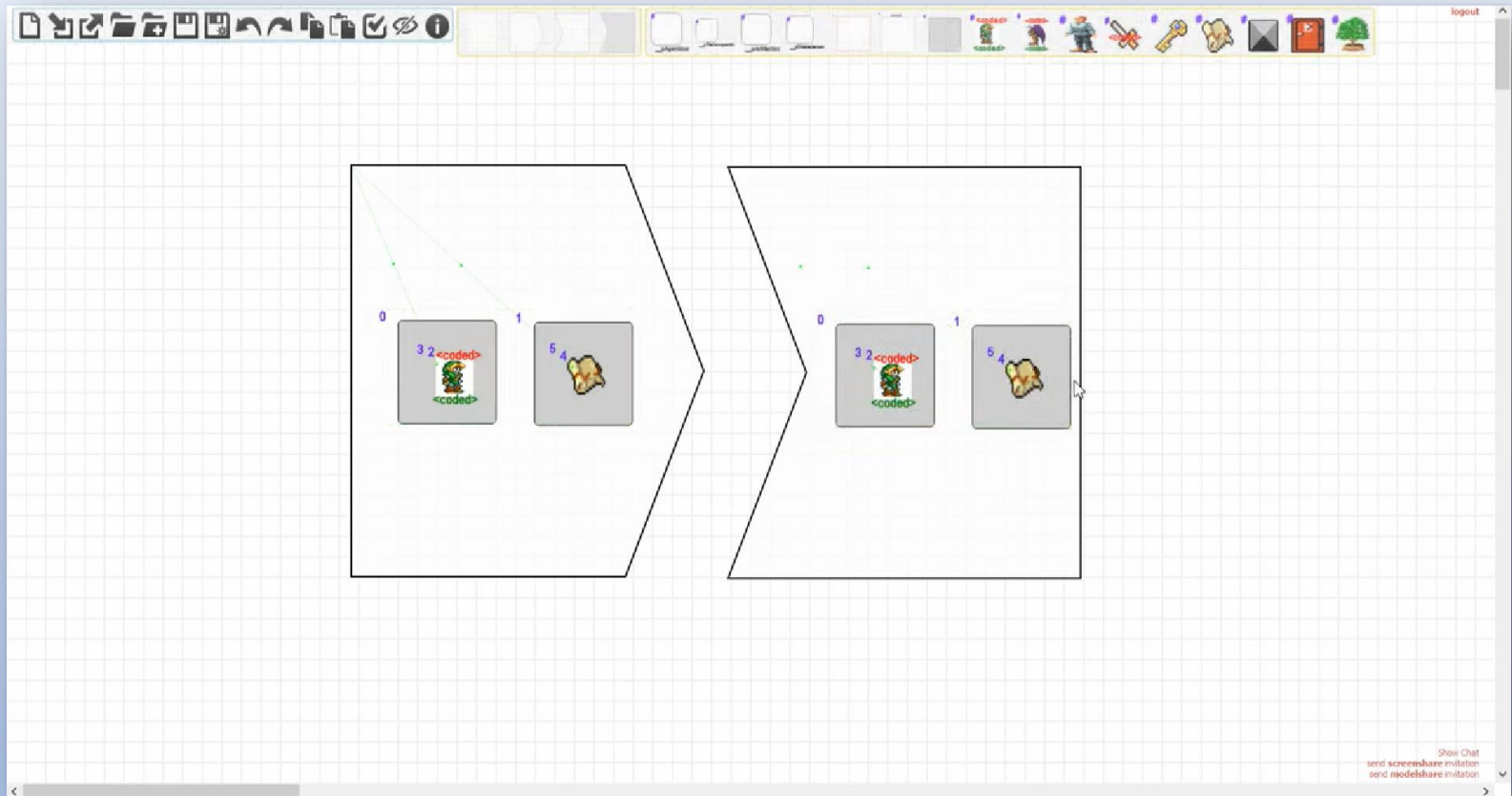


Design how they look like

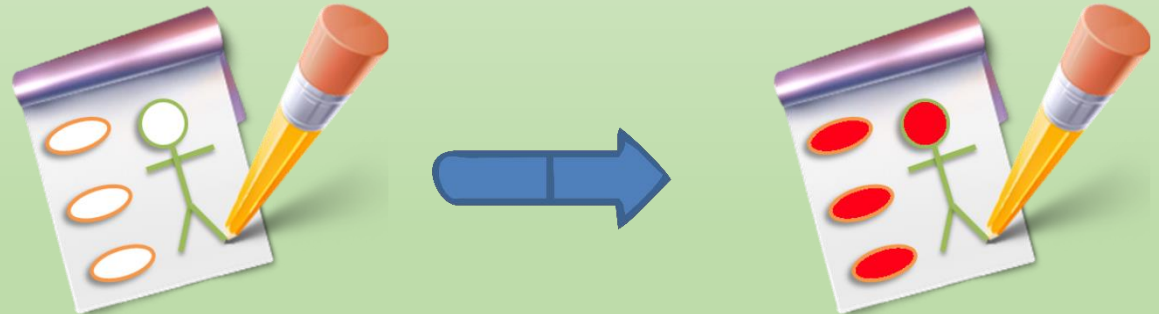
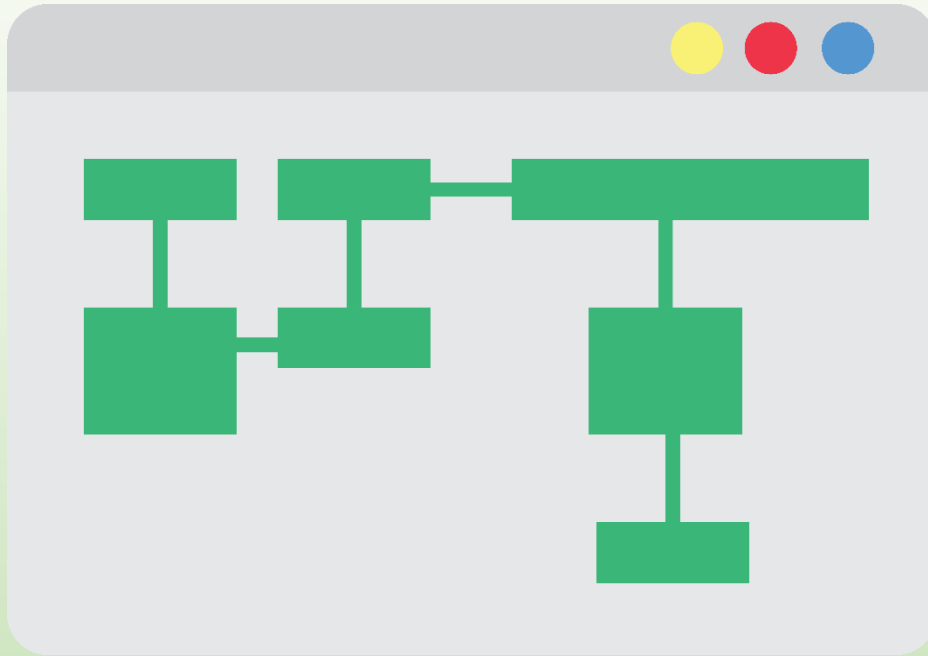


Your game environment is ready...





Behavior using model transformation...



**A generic modeling environment...**



**Again,  
how does this help?**







**They know their domain better than any developer!**



Consequences  
Just





# CONCLUSIONS

---

A

B

C

Accidental

VS

**ESSENTIAL**





A photograph of a classroom. In the background, a large black chalkboard is mounted on a wall with wood paneling. The text is overlaid on the chalkboard. In the foreground, several rows of wooden desks with attached benches are visible. One desk in the middle ground has a stack of books on it. The lighting is warm, and the overall scene is that of a traditional lecture hall.

**CS 439/539: Model-Driven Design  
(pending approval for Spring 2019)**