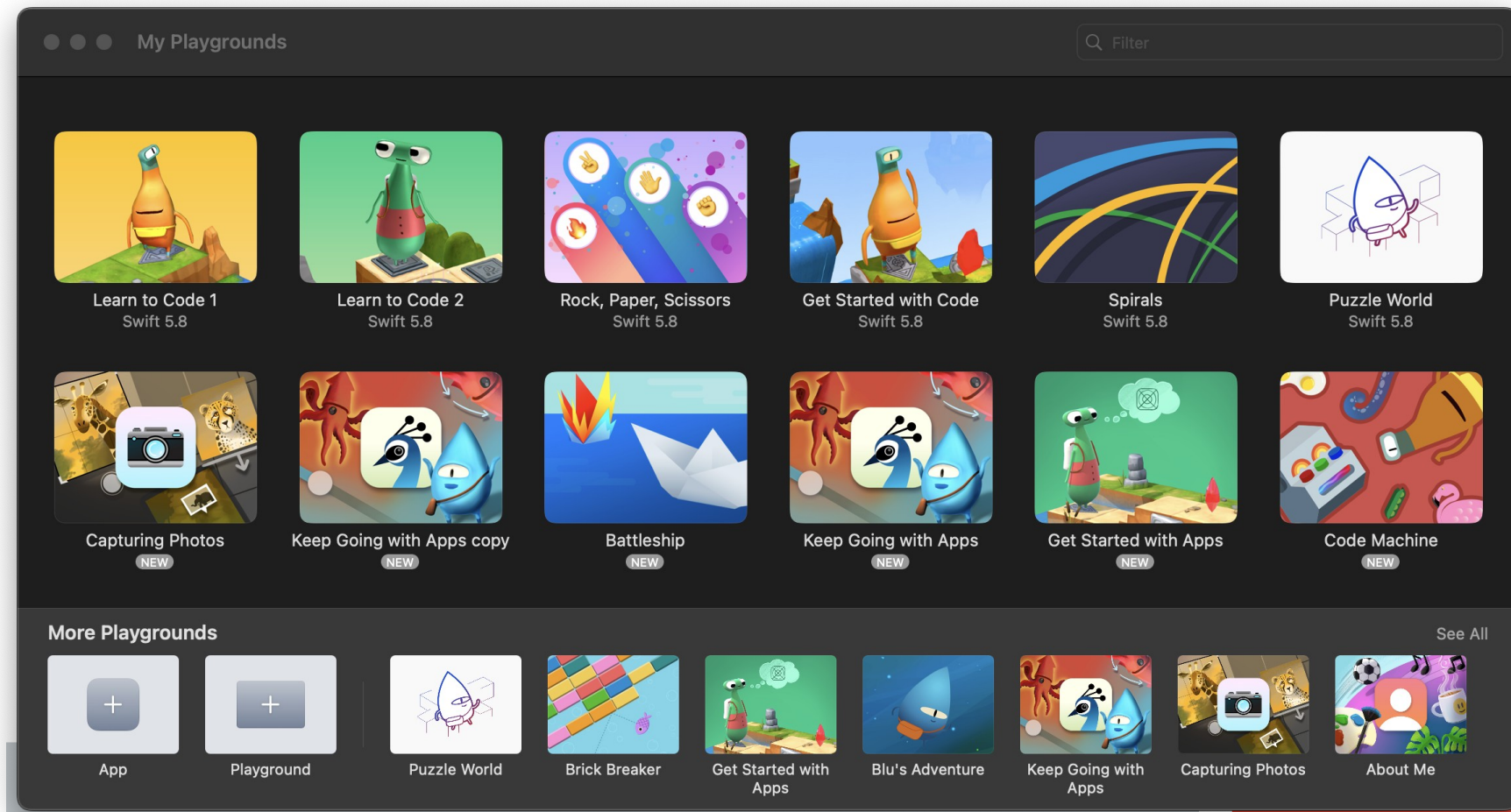


Other interesting systems



Andrea Sterbini – sterbini@di.uniroma1.it

Swift Playgrounds (for MacOS)



Swift programming language

Born as a replacement of Objective-C (originally for NeXT OS, then in Mac OSX)

Runs well together with C/C++ and Objective-C compiled libraries

Statically typed, with type inference

With a simpler syntax (e.g. no ‘;’, simpler range specs: 1 ... N)

With closures (aka anonymous functions)

Protocol-based class extensions (called “mixins” in other languages)

Functions can be passed as values

Generic functions/methods, operator overloading

Array bounds always checked

Variables always initialized

No pointer arithmetic or other unsafe operations

Playgrounds: small programming tasks

E.g. Loops, conditionals, boolean ops, functions, variables, types, methods

Start with a short set of slides explaining new concepts

Describe the task (normally involving an animated character)

With automatic checks for the program syntax

With programming templates to write control structures and function calls

Small programming tasks moving a creature in a labyrinth

Move around
Collect gems
Activate buttons
Activate portals
Use conditions
Use loops
...

Methods in C

Get Started with Code
Looping All the Sides

Goal: Use a `for` loop to repeat a sequence of commands.


In this puzzle, you must collect four gems that are located in the same relative locations around a square. You'll create a `loop` that repeats the code below for each of the sides to solve the entire puzzle.

- 1 Select `for` in the shortcut bar to add a `for` loop into your code.
- 2 Press the bottom curly brace to select the loop.
- 3 Press and hold the curly brace, then drag it downward to pull the existing code into the loop.

```
for i in 1 ... 4 {  
  moveForward()  
  collectGem()  
  for i in 1 ... 3 {  
    moveForward()  
  } ..  
  turnRight()  
}
```

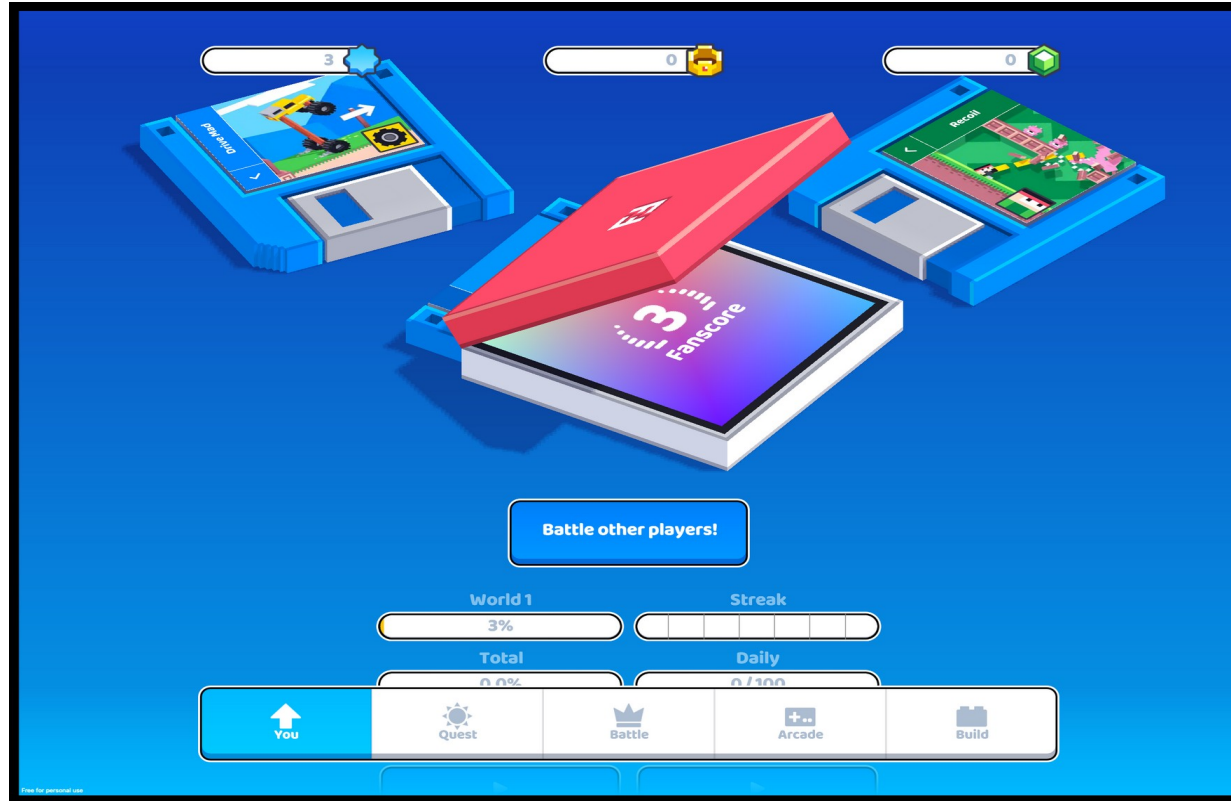
foricollectGem()moveForward()turnRight()

1/4



Stop

FANCADE



Fancade: a 3D data-flow game building language

Available in Android, Windows, MacOS, or in the Browser

Data-flow computation WITH VARIABLES

Typed links (Number, Boolean, Position, Rotation, Object, Constraint, ...)

NO text! (except for code comments and menus)

Executed 60 times for second (implicit forever loop)

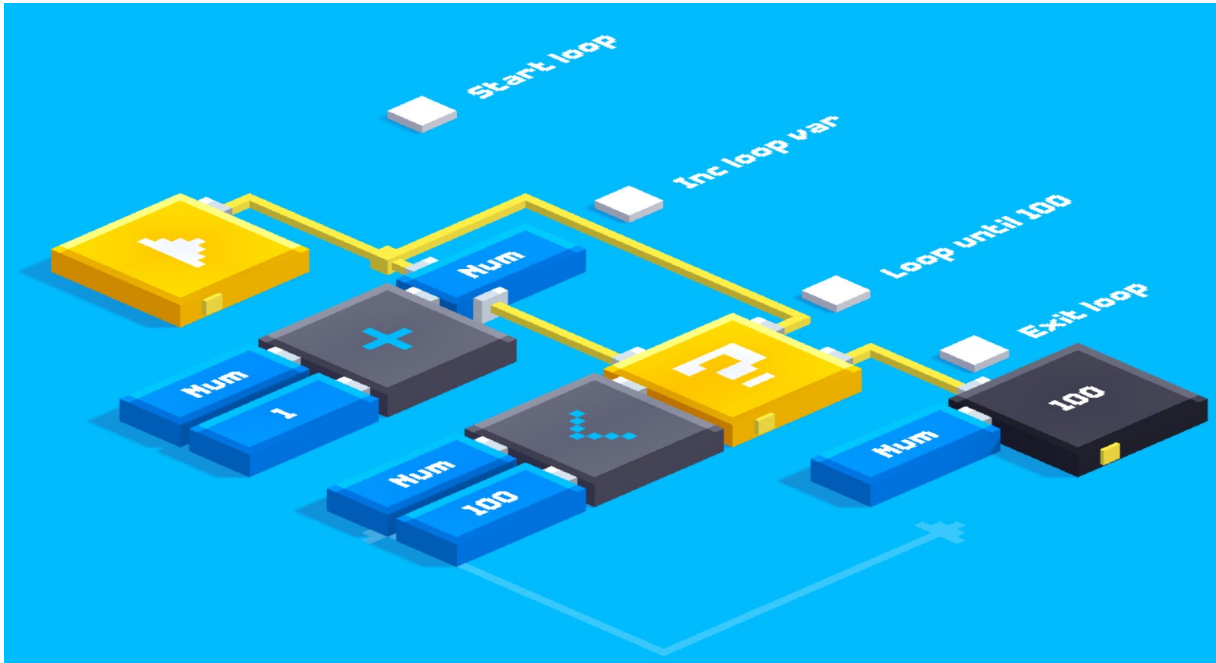
Execution order: left-right and top-down (with respect to the isometric positions)

Explicit Variables (used also to reduce wires and increase fan-out)

Code can read/write game blocks properties (position/speed/bounce ...)

Can add or change movement/rotation constraints for blocks/agents

Explicit Control = before/after links (jellow wires)



Loops

do:

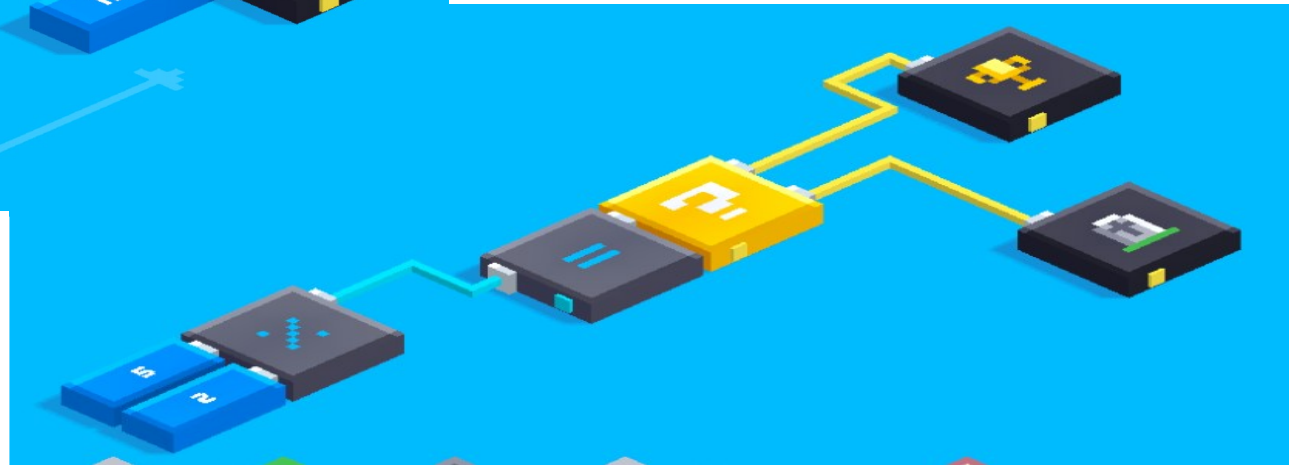
Num += 1

while num<100

If-then-else

if $5\%2==0$:

then: win else: lose



Functions: custom blocks

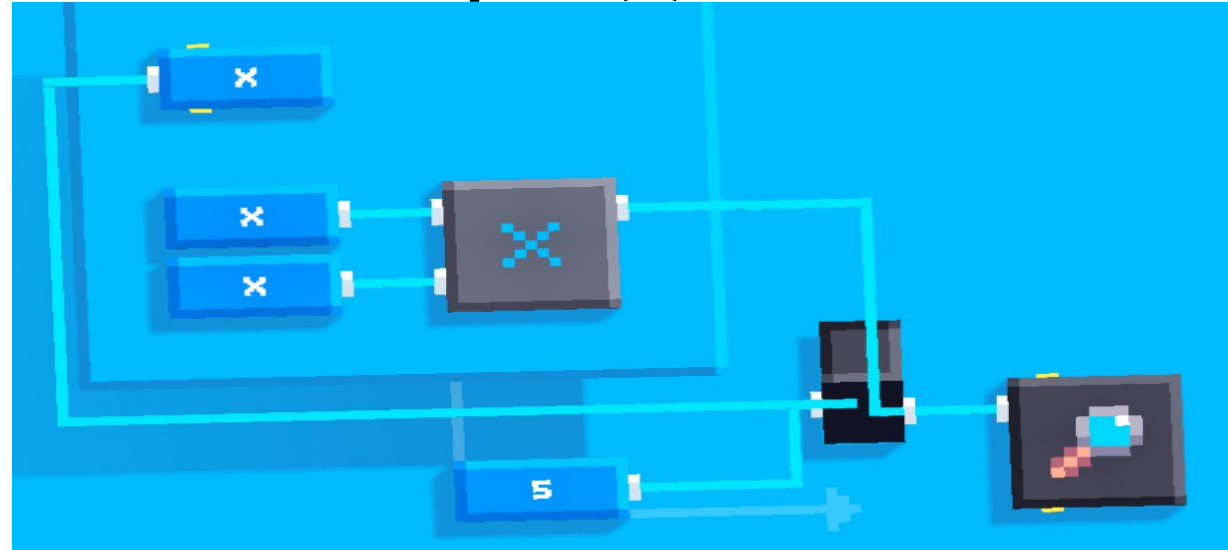
Functions? YES

Just put “code” inside a block, with external connectors

Modularization:

- custom blocks can be placed inside other blocks
- BUT: NO recursion is allowed (direct or indirect)

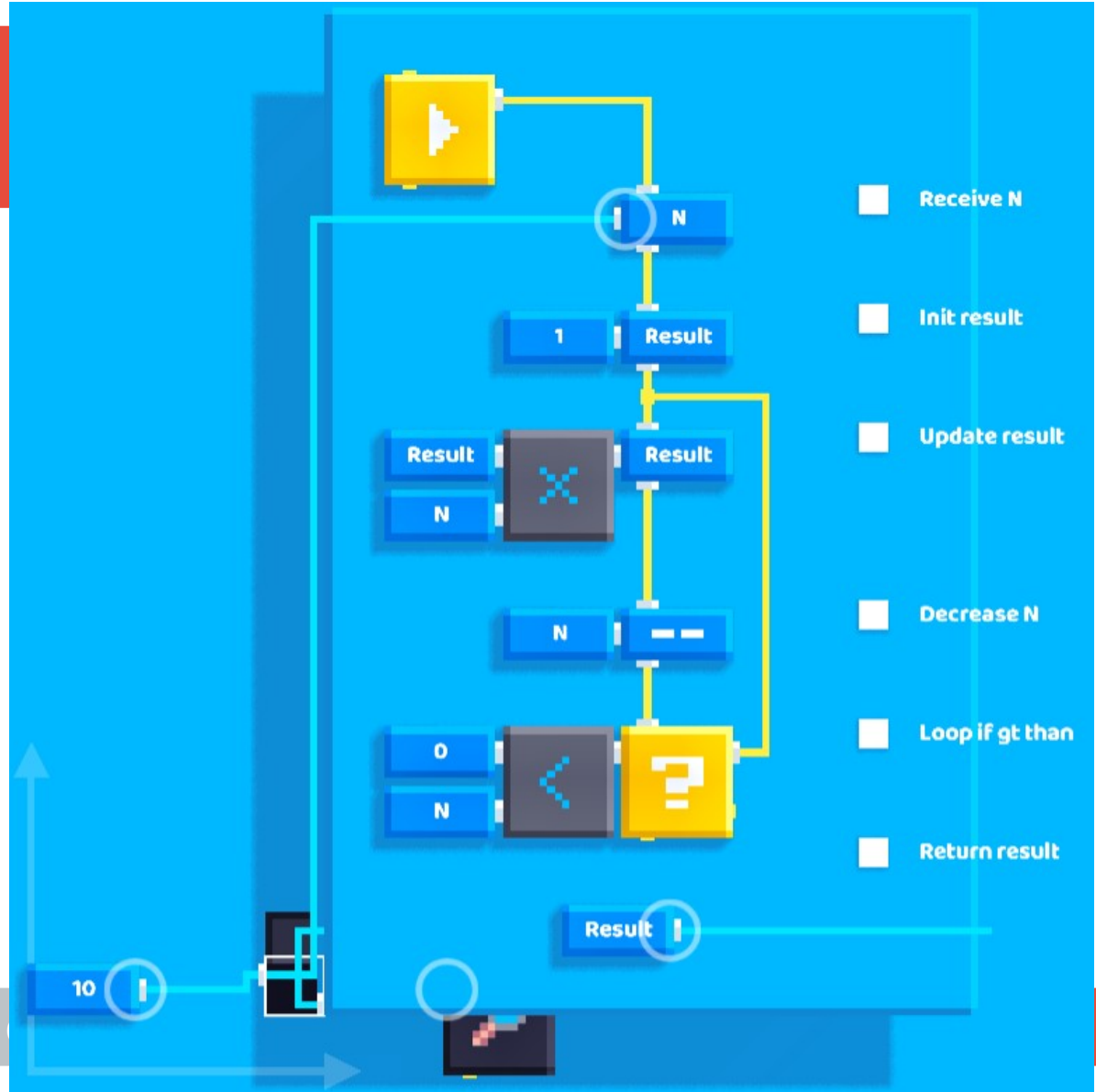
```
def square(x): return x*x
```



No recursion (even implicit)

Iterative factorial(N)

Methods in Computer Science



Many game-oriented

VALUES (different types)

VARIABLES (getter/setters)

MATH (operators)

CONTROL (if-then, loops ..)

PHYSICS (gravity, vectors, ..)

GAME (win/lose/score)

SCENERY

SCRIPTS

INSPECTORS

CREATURES

Methods in Compute



Values



Variables



Control



Physics



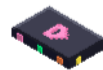
Game



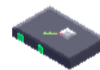
Objects



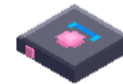
Get Position



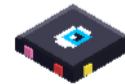
Set Position



Raycast



Get Size



Set Visible



Create Object



Destroy Object

DEMO

PlayOsmo

TANGIBLE interaction for preschool kids (with iPad or Fire)

Your tablet scans the table in front of it (on a stand, with a mirror on camera)

You play with tiles (instructions or tangram pieces or letters or drawings)

It recognizes your “code tiles” or words or drawing depending on the game/app



Loops and simple repetitions

Blocks represent action and direction

- modifier: number of repetitions

Loops

The instructions' meaning depends on the game

- Music playing game
- Monster in a labyrinth game



But also many game apps

Drawing (virtual reality helped drawing)

Spell/word games (character recognition)

Tangram (shape recognition)

Pizza shop (shapes + money/math)

Wizards (math or words)

...



Ren'Py

Dialogue-based adventures (Visual Novels)

Python-inspired simplified syntax (with indentation)

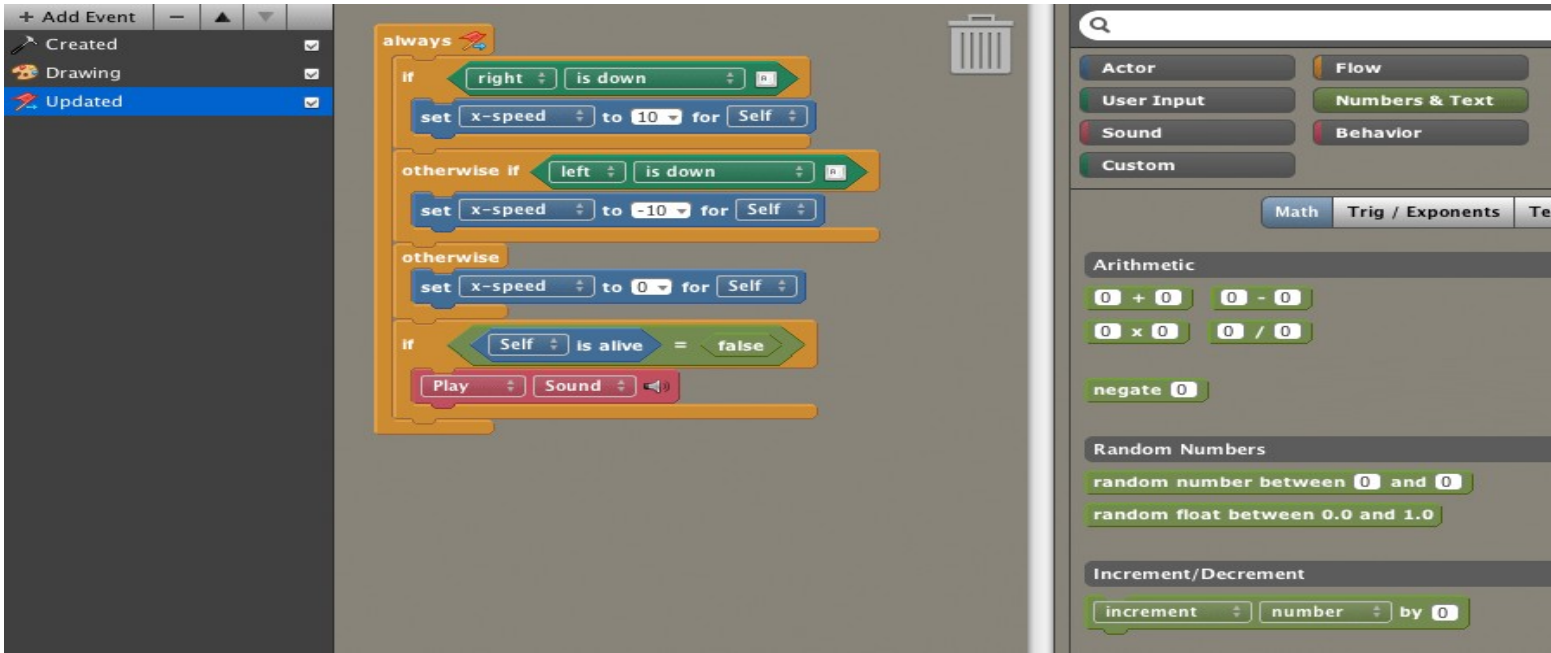
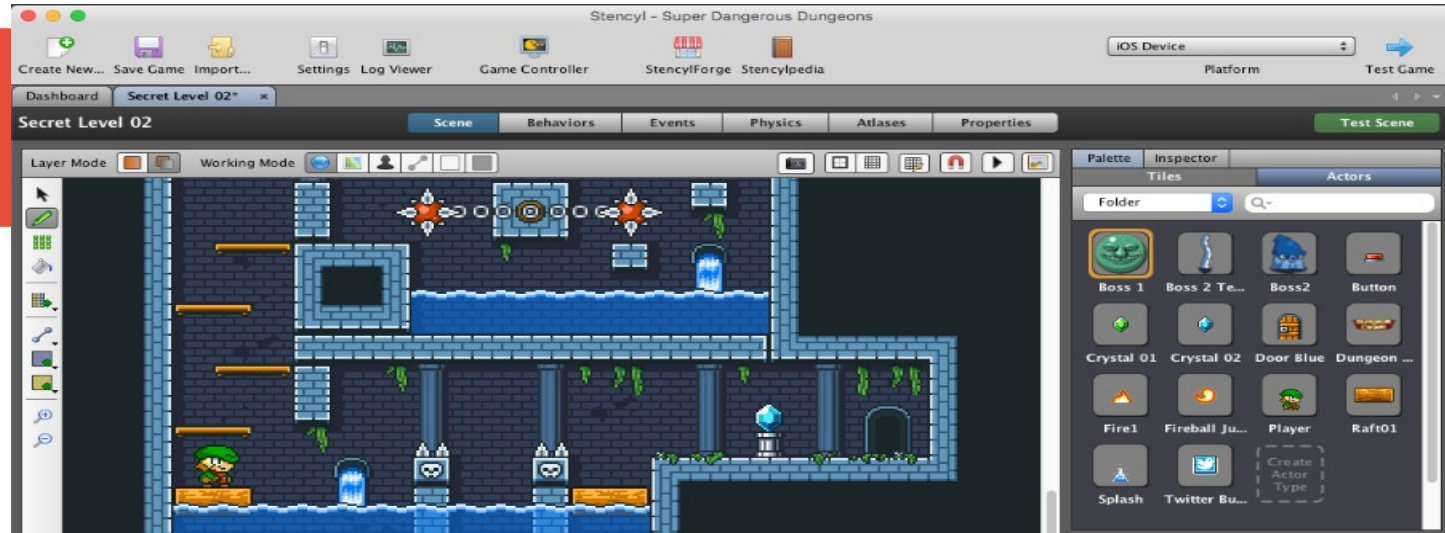
Runs in Python



Methods in

Stencyl

Game designer (platforms)



22-23 Other systems

Alice 3 : 3D world programming

Alice



Alice 3 interface showing a 3D scene and a procedure editor.

Scene View (Left):

- A:** File Edit Project Run Window Help menu bar.
- B:** 3D scene view showing a forest with mushrooms and a table.
- C:** Object list showing **cheshireCat**.
- D:** Procedures and Functions panel for **cheshireCat**, showing methods like **say**, **think**, **move**, **moveToward**, **moveAwayFrom**, **moveTo**, **place**, **turn**, **roll**, and **turnToFace**.
- E:** Run button.
- F:** Setup Scene button.

Procedure Editor (Right):

- H:** Tab bar showing **Scene**, **initializeEventListeners**, and **myFirstMethod**.
- I:** Save icon.
- J:** Procedure editor area showing the **myFirstMethod** procedure.
- K:** Bottom toolbar with options like **do in order**, **count**, **while**, **for each in**, **if**, **do together**, **each in together**, **variable...**, **assign**, and **//comment**.

myFirstMethod Procedure:

```
declare procedure myFirstMethod
do in order
  <alice> say "Where am I?" , duration = 2.0 add detail
  <cheshireCat> say "Where do you want to go?" , duration = 2.0 add detail
  <alice> say "I do not know..." , duration = 2.0 add detail
  <cheshireCat> say "Then it does not matter where you are..." , duration = 2.0 add detail
  <cheshireCat> setOpacity = 0.0 , duration = 4.0 add detail
```

Others

Kodu: Design 3D games on Xbox (and PC)

Kodular: game editor

Construct: HTML5 game editor

MakeBlock: robots/microcontrollers + Scratch

Tynker: Scratch-inspired

Roblox: 3D game programming

... and many other game-editors