

Other notable systems



Fancade: a 3D data-flow game building language

Data-flow computation

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

NO text!

Executed 60 times for second (implicit forever loop)

Execution order: left-right and top-down

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

Code can read/write blocks properties

Explicit Control = before/after links (jellow wires)

Loops

do:

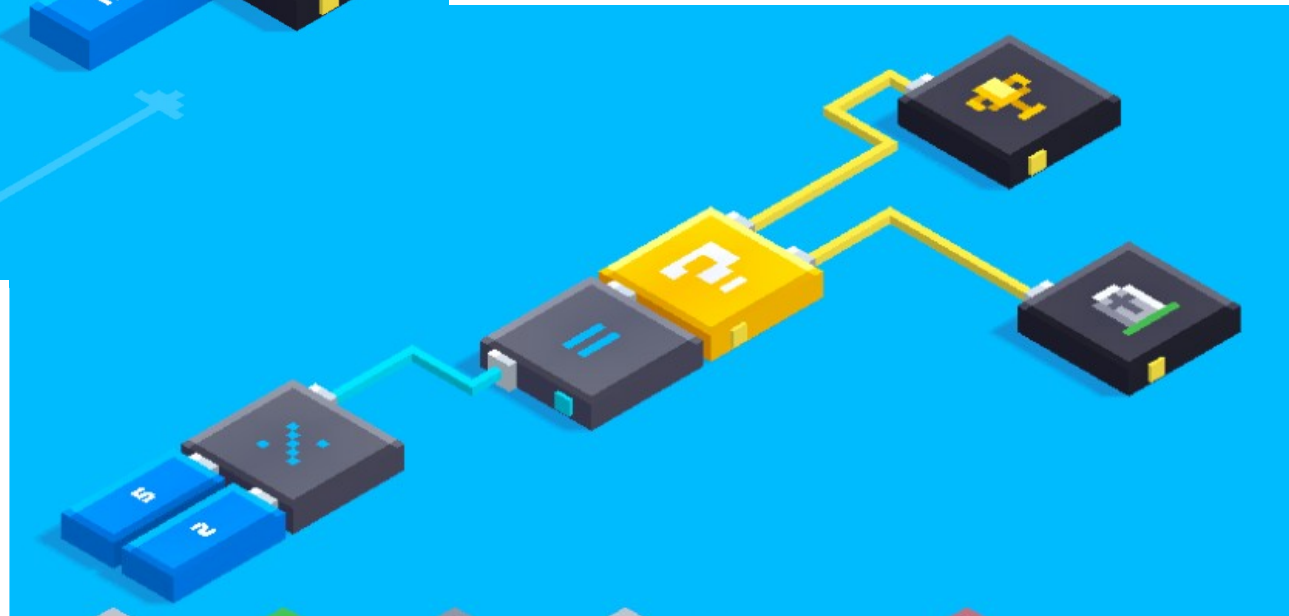
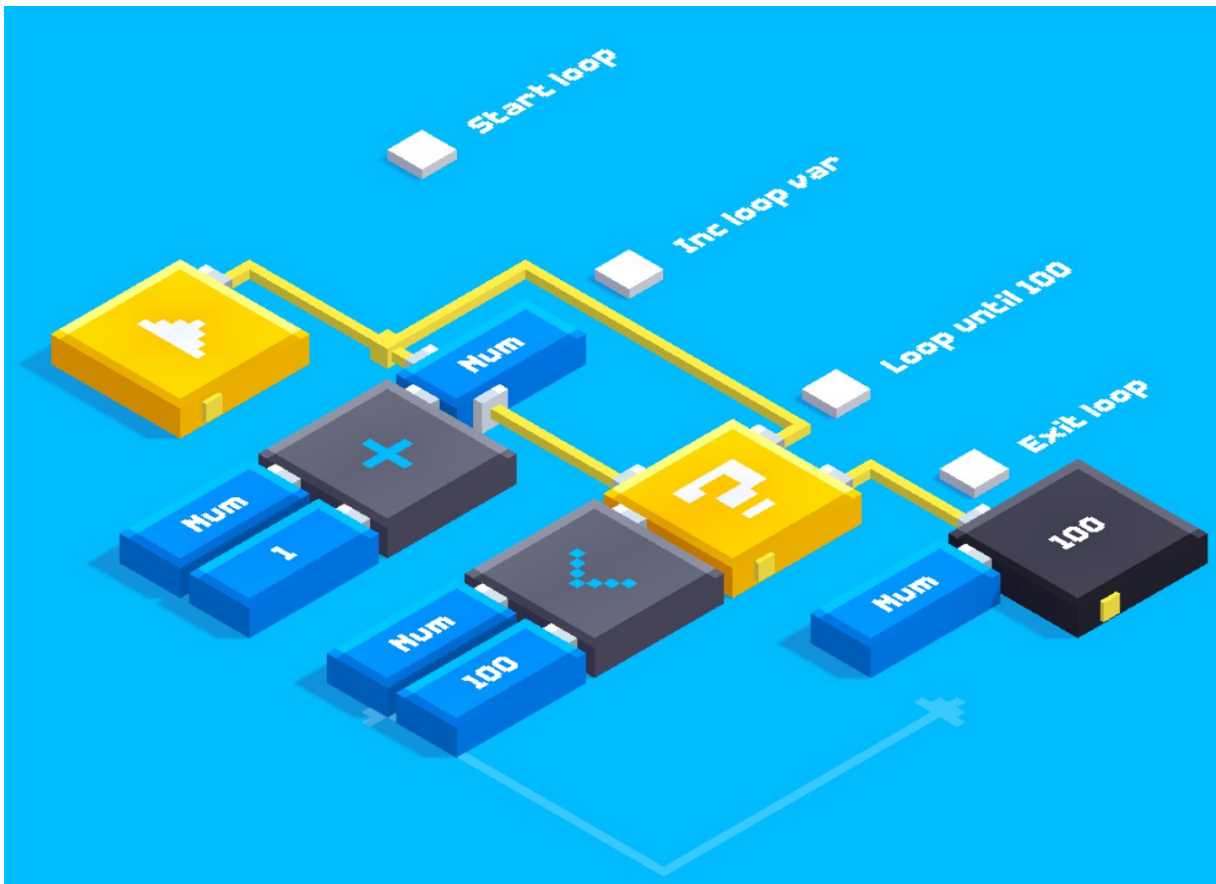
Num += 1

while num < 100

If-then-else

if $5 \% 2 == 0$:

then: win else: lose

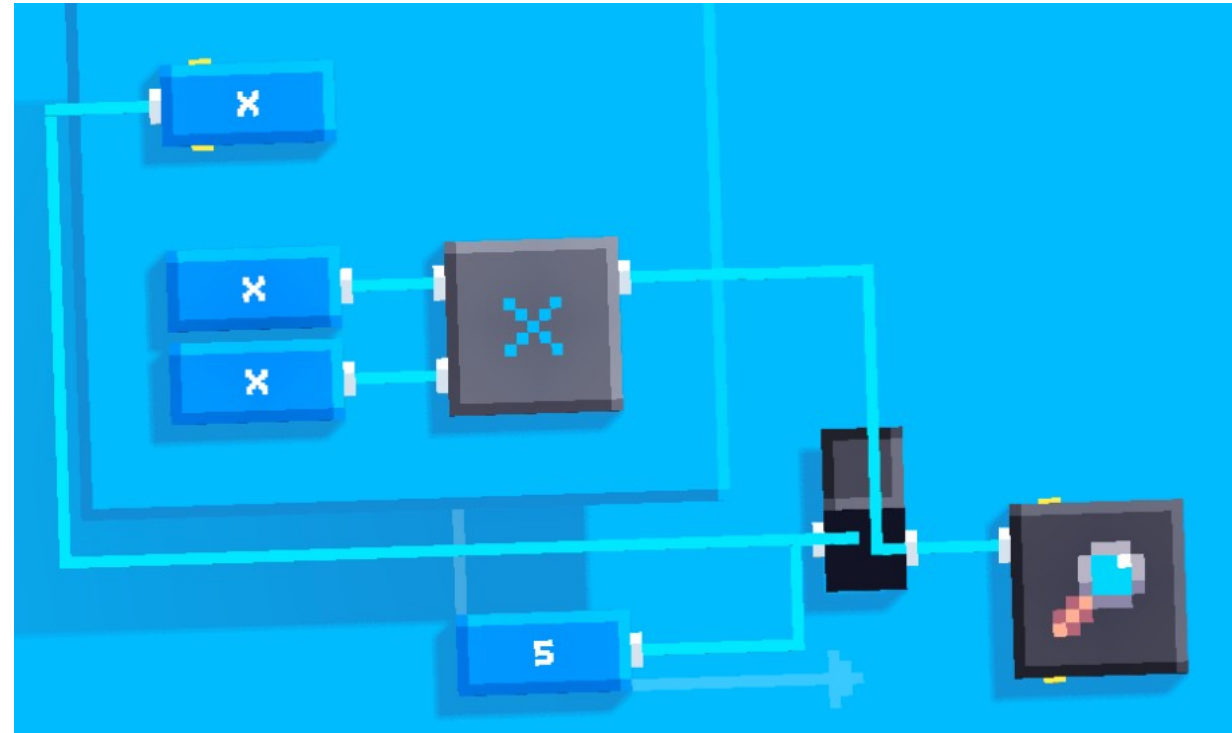


Functions: custom blocks

Functions? YES

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

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



Modularization:

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

Many game-oriented block categories

VALUES (different types)

GAME (win/lose/score)

VARIABLES (getter/setters)

CREATURES

MATH (operators)

PHYSICS (gravity, vectors,
springs, ...)

CONTROL (if-then, loops ...)

SCENERY

SCRIPTS

INSPECTORS

DEMO

PlayOsmo: tangible interaction for kids

Your iPad scans the table in front of it (with an attached mirror)

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



or



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

Drawing helper (virtual reality drawing)

Spell/word games (character recognition)

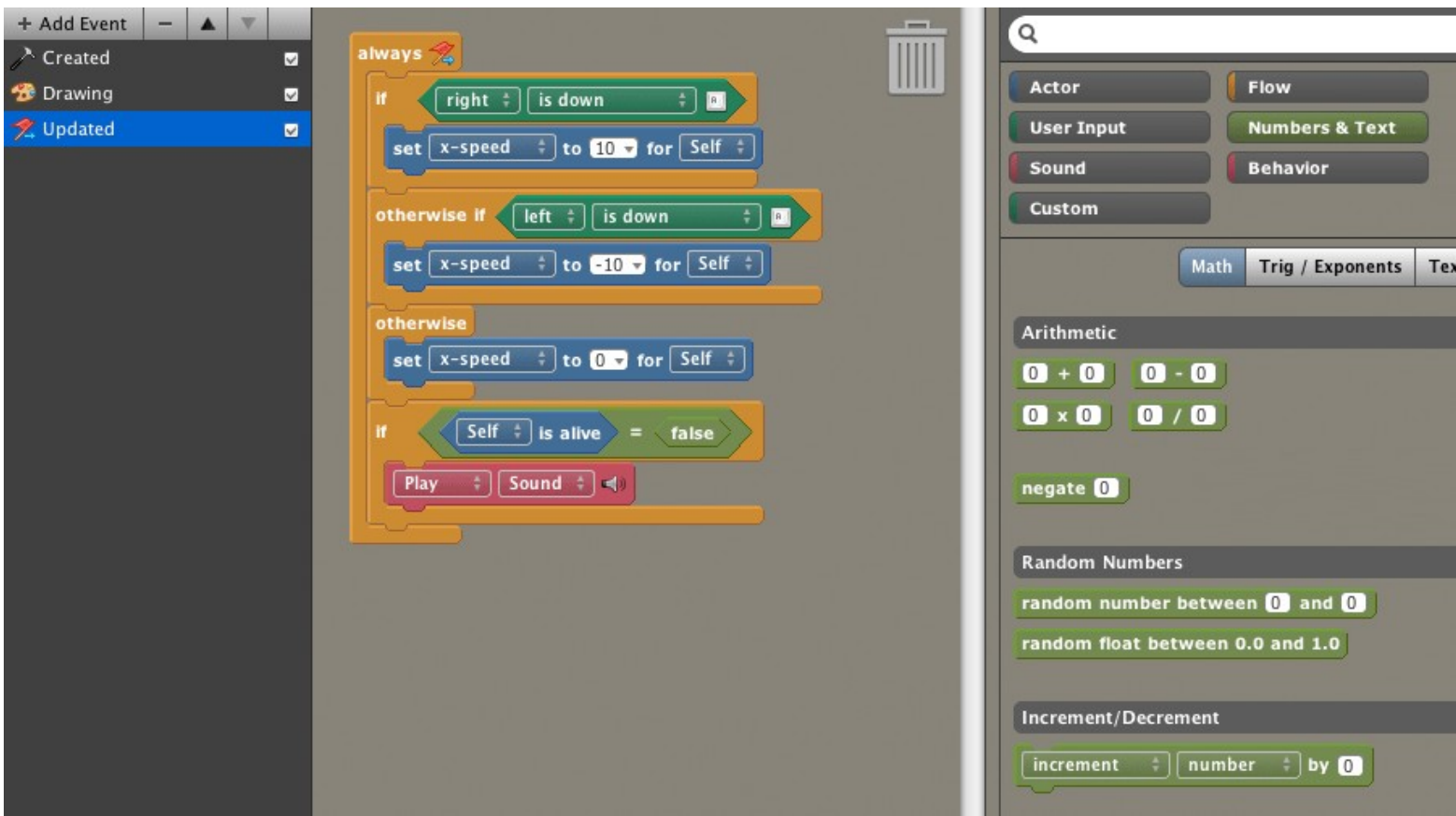
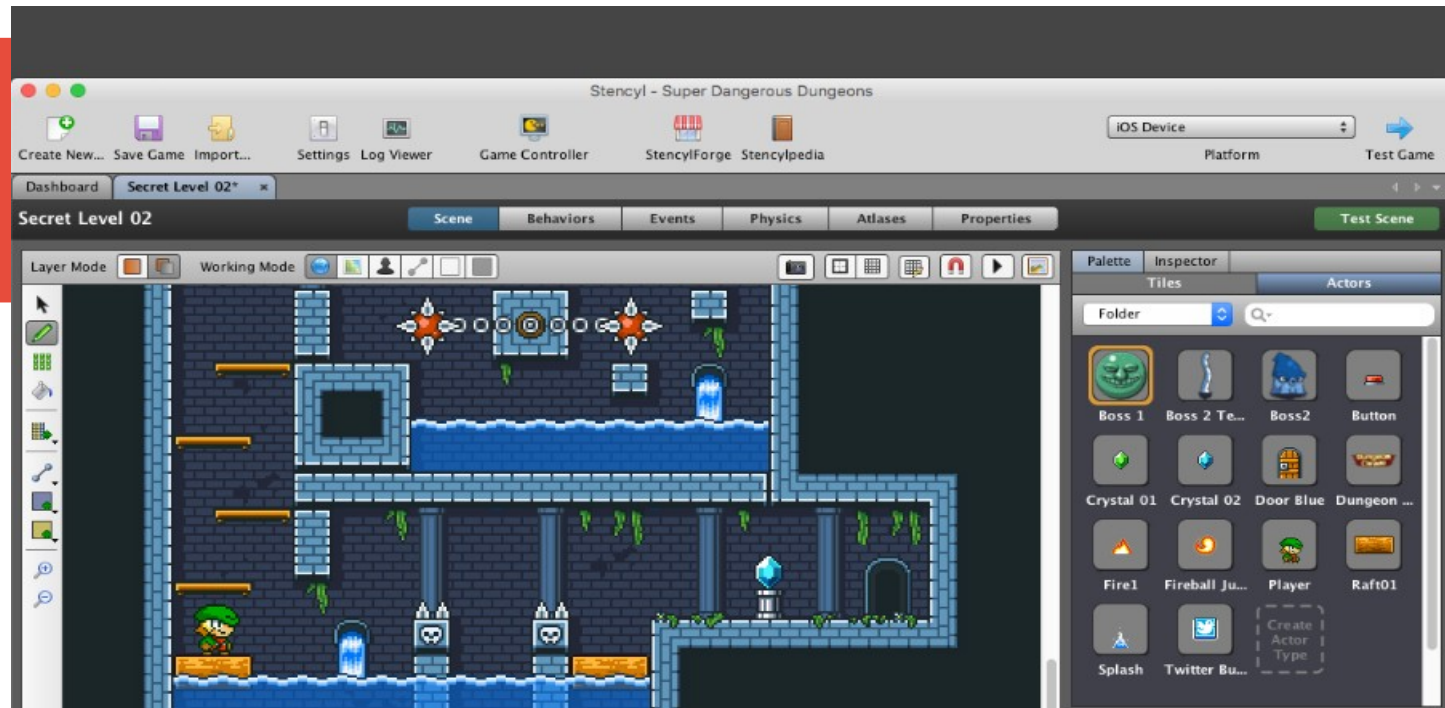
Tangram (shape recognition)

Games (shape recognition)

DEMO

Stencyl

Game designer (platforms)



2021-22 Other systems

Alice 3 : 3D world programming

Alice



Figure 1: Alice 3 interface showing the 3D scene and the programming environment.

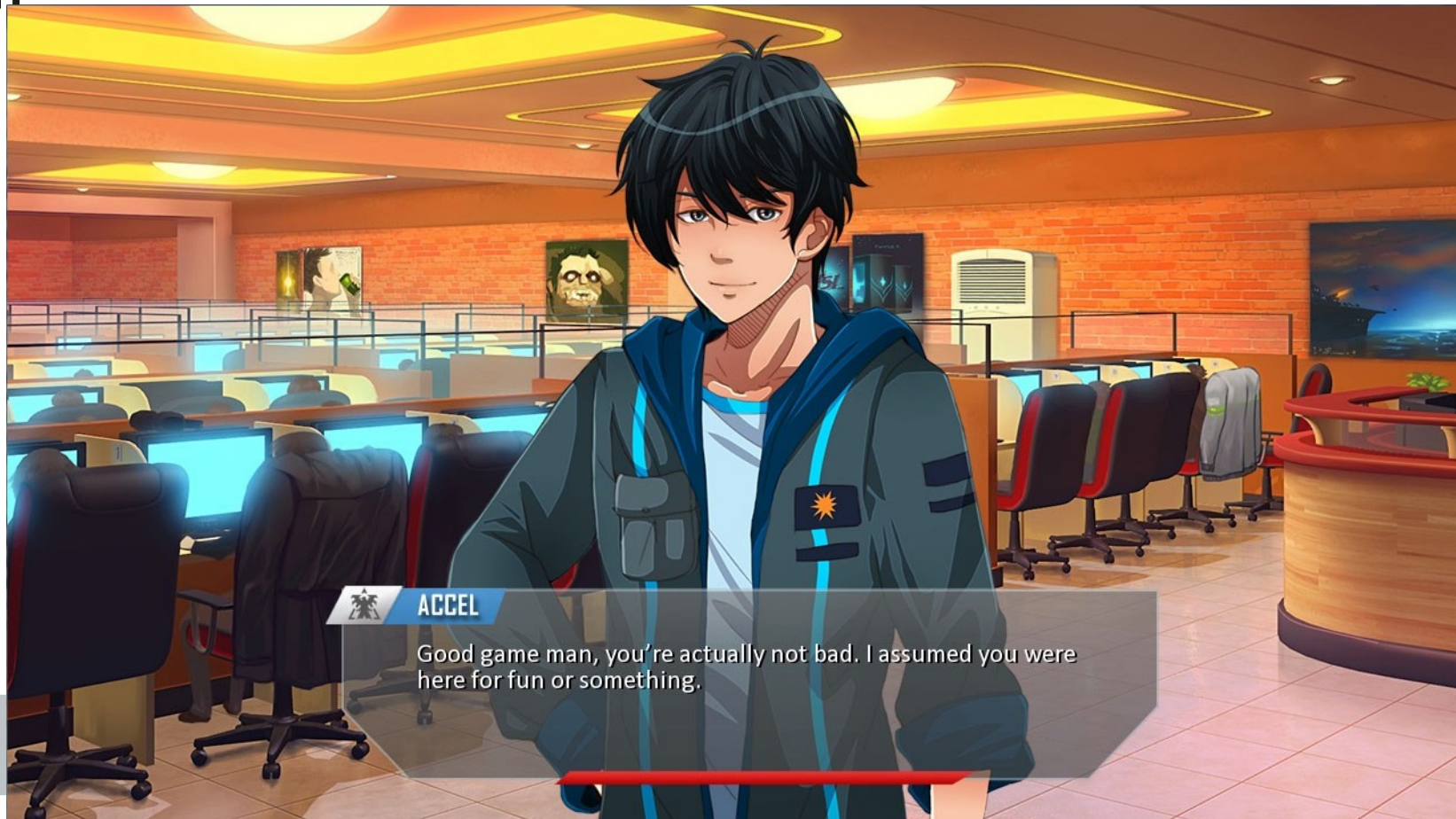
The interface is divided into several panels:

- Scene View (A):** Displays the 3D world with Alice and the Cheshire Cat in a forest setting.
- Setup Scene (F):** A button to configure the scene.
- Object List (C):** Shows the selected object, **cheshireCat**.
- Procedures/Functions (D):** A list of available actions for the selected object, categorized by type (say, think, position, orientation).
- Scripting Area (H):** The main area for writing code. It shows a procedure named **myFirstMethod** with the following steps:
 - 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
- Run Button (G):** A green play button to execute the script.
- Scene Tab (H):** A tab to switch between different scenes.
- Scripting Area (J):** The main area for writing code.
- Scripting Area (K):** The main area for writing code.

Dialogue-based adventures (Visual Novels)

Python-inspired simplified syntax (with indentation)

Runs in Python



Others

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

Kodular: game editor

Construct: HTML5 game editor

Swift Playground: by Apple

MakeBlock: robots/microcontrollers + Scratch

Tynker: Scratch-inspired

... and many other game-editors