VPL Taxonomy

SCG Seminar Project

Mario Kaufmann

Project

- VPL: visual programming language

- Create taxonomy for VPLs
  - how should a VPL be characterized?

- Existing surveys incomplete or out of date

Visual Programming

“Visual programming is programming in which more than one dimension is used to convey semantics”

Margaret M. Burnett, 1999

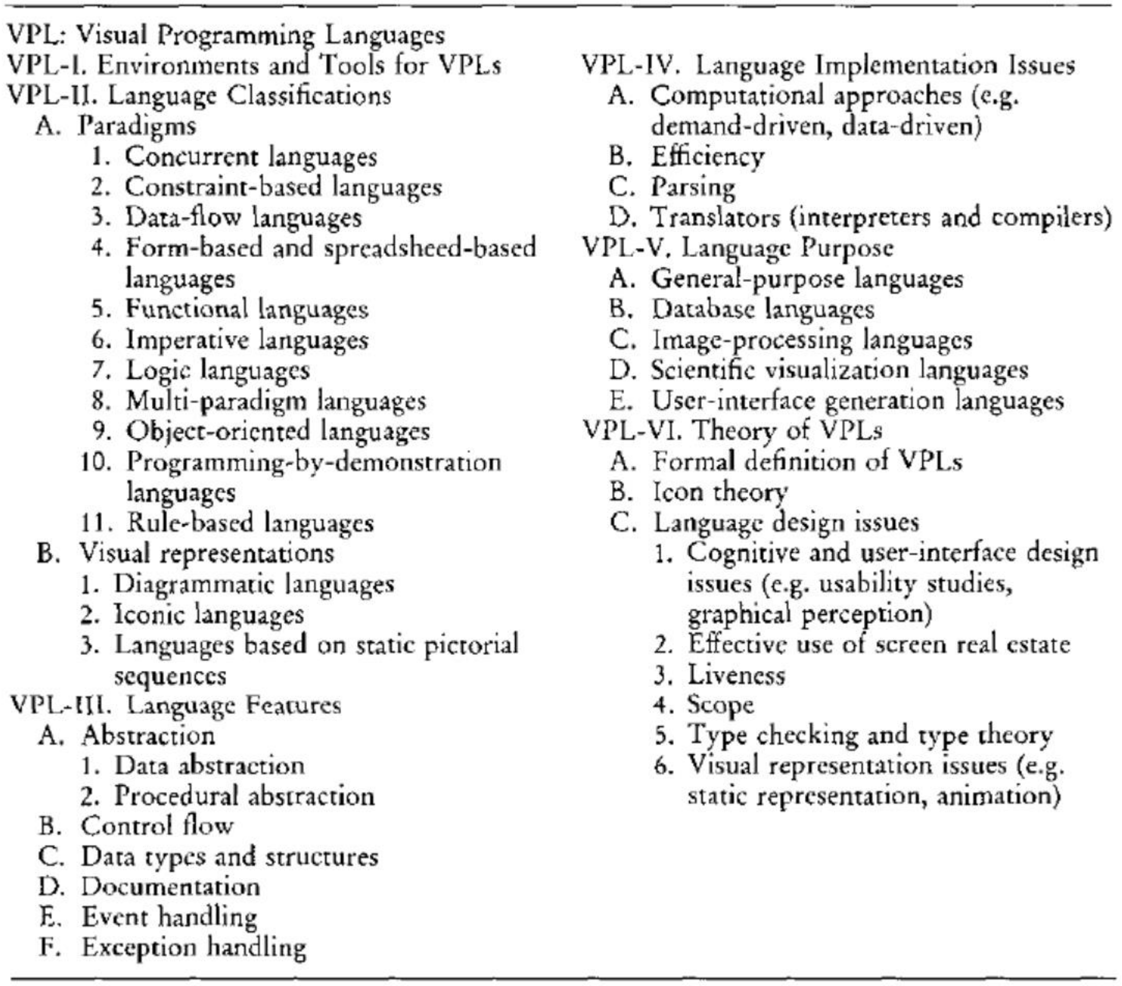
- Additional dimensions:
  - multidimensional objects
  - spatial relationships
  - time dimensions

Visual Programming Language

- Definition not clear-cut

- Approach:
  - collect VPLs
  - extract features
  - create a classification system

Taxonomy



VPL: Visual Programming Languages

VPL-I. Environments and Tools for VPLs

VPL-II. Language Classifications

- A. Paradigms
  1. Concurrent languages
  2. Constraint-based languages
  3. Data-flow languages
  4. Form-based and spreadsheed-based languages
  5. Functional languages
  6. Imperative languages
  7. Logic languages
  8. Multi-paradigm languages
  9. Object-oriented languages
  10. Programming-by-demonstration languages
  11. Rule-based languages
- B. Visual representations
  1. Diagrammatic languages
  2. Iconic languages
  3. Languages based on static pictorial sequences

VPL-III. Language Features

- A. Abstraction
  1. Data abstraction
  2. Procedural abstraction
- B. Control flow
- C. Data types and structures
- D. Documentation
- E. Event handling
- F. Exception handling

VPL-IV. Language Implementation Issues

- A. Computational approaches (e.g. demand-driven, data-driven)
- B. Efficiency
- C. Parsing
- D. Translators (interpreters and compilers)

VPL-V. Language Purpose

- A. General-purpose languages
- B. Database languages
- C. Image-processing languages
- D. Scientific visualization languages
- E. User-interface generation languages

VPL-VI. Theory of VPLs

- A. Formal definition of VPLs
- B. Icon theory
- C. Language design issues
  1. Cognitive and user-interface design issues (e.g. usability studies, graphical perception)
  2. Effective use of screen real estate
  3. Liveness
  4. Scope
  5. Type checking and type theory
  6. Visual representation issues (e.g. static representation, animation)

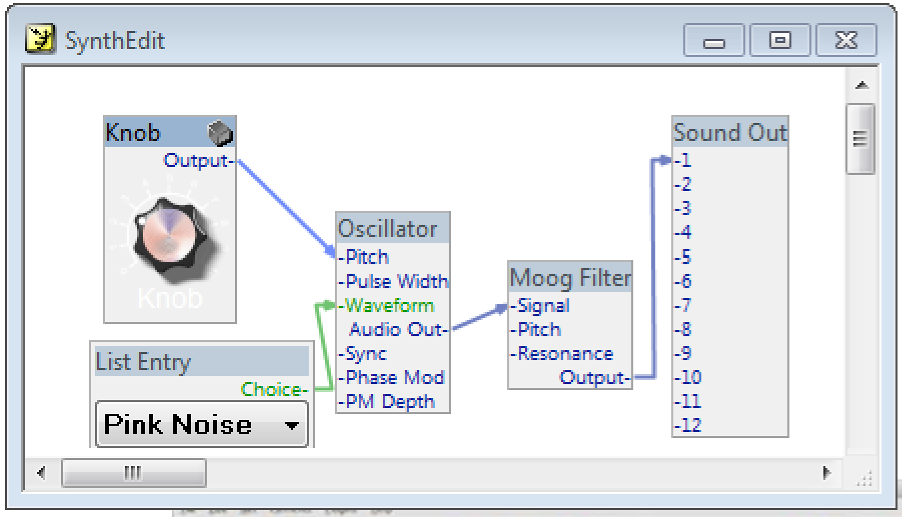
Classification system by Burnett and Baker

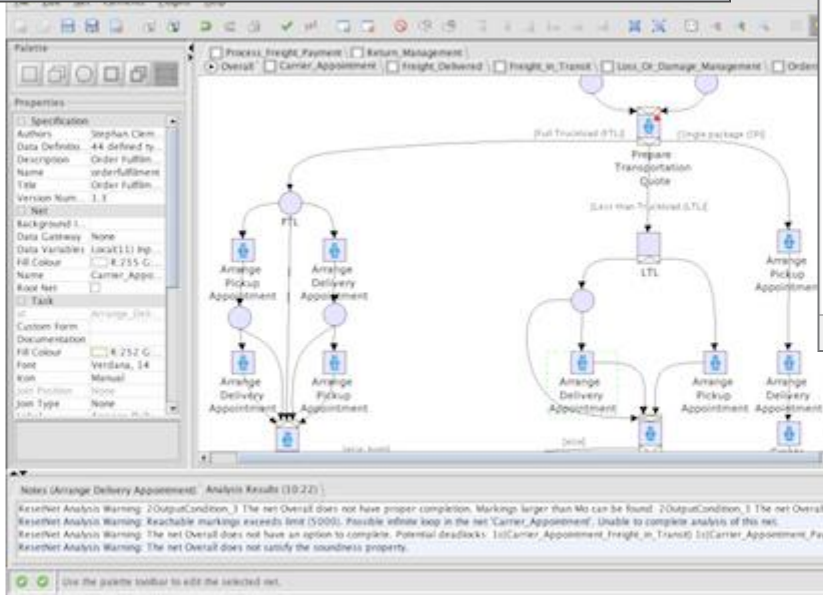
- paradigms and visual representation combined
- purpose

Taxonomy

- 2 extra dimensions based on VPLs found:
  - programming knowledge
  - amount of text code

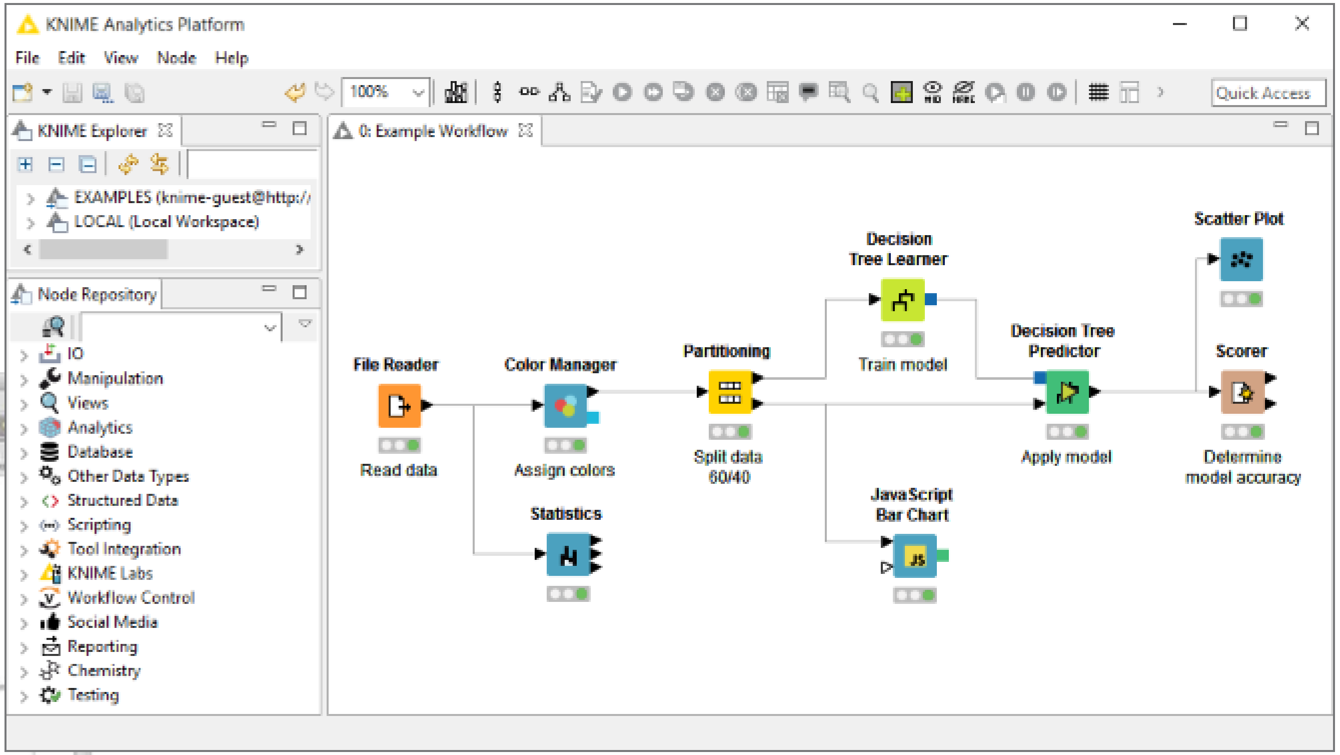
Paradigm: graph-based (dataflow)



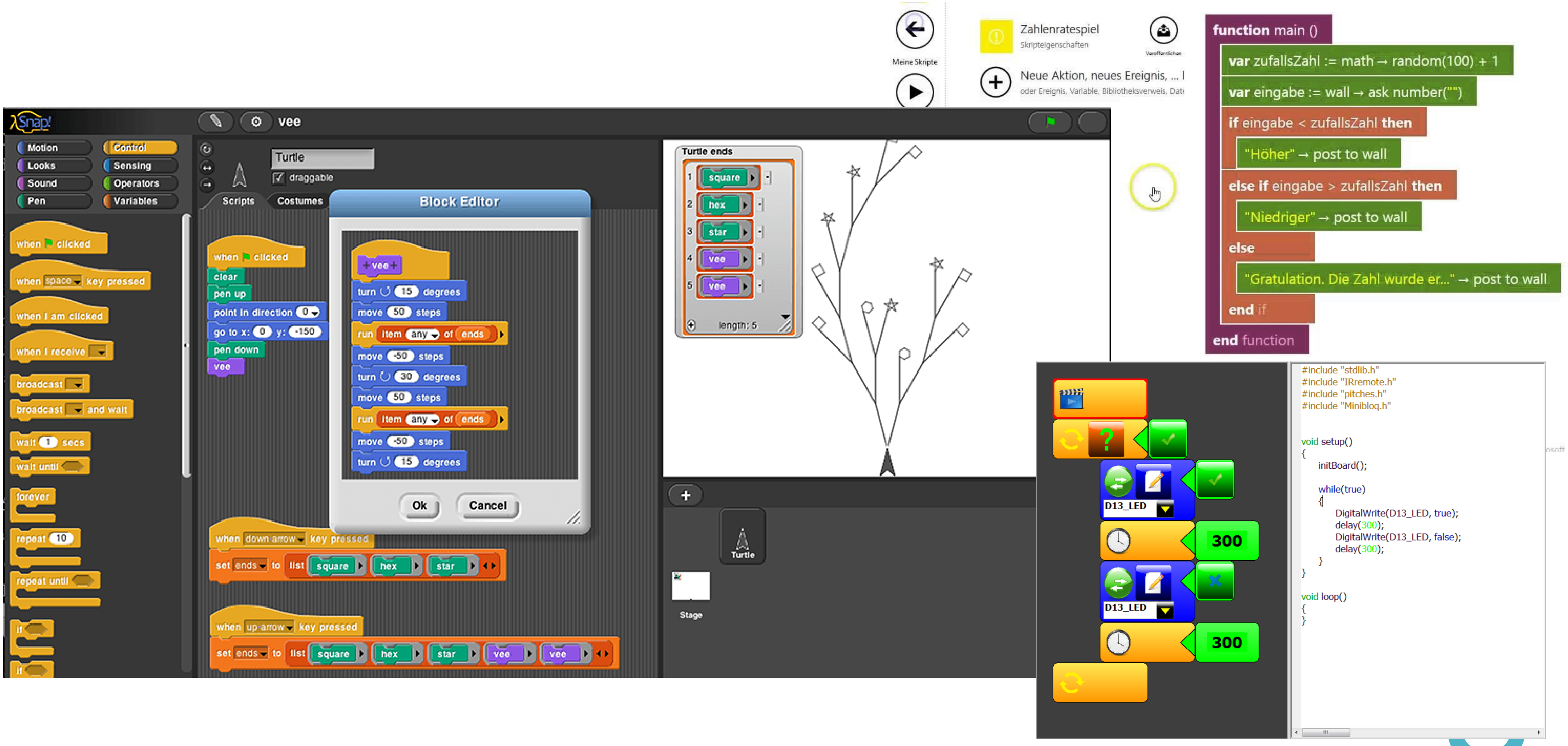

Prepare Transportation Quote
Arrange Pickup Appointment
Arrange Delivery Appointment
FTL
LTL
Notes (Arrange Delivery Appointment)
Analysis Results (10:22)

```
ResetNet Analysis Warning: 2OutputCondition_3 The net Overall does not have proper completion. Markings larger than Mo can be found: 2OutputCondition_3 The net Overall does not ha...
ResetNet Analysis Warning: Reachable markings exceeds limit (5000). Possible infinite loop in the net 'Carrier_Appointment'. Unable to complete analysis of this net.
ResetNet Analysis Warning: The net Overall does not have an option to complete. Potential deadlocks: 1i(Carrier_Appointment_Freight_in_Transit) 1i(Carrier_Appointment_Payment)
ResetNet Analysis Warning: The net Overall does not satisfy the soundness property.
```

Use the palette toolbar to edit the selected net.



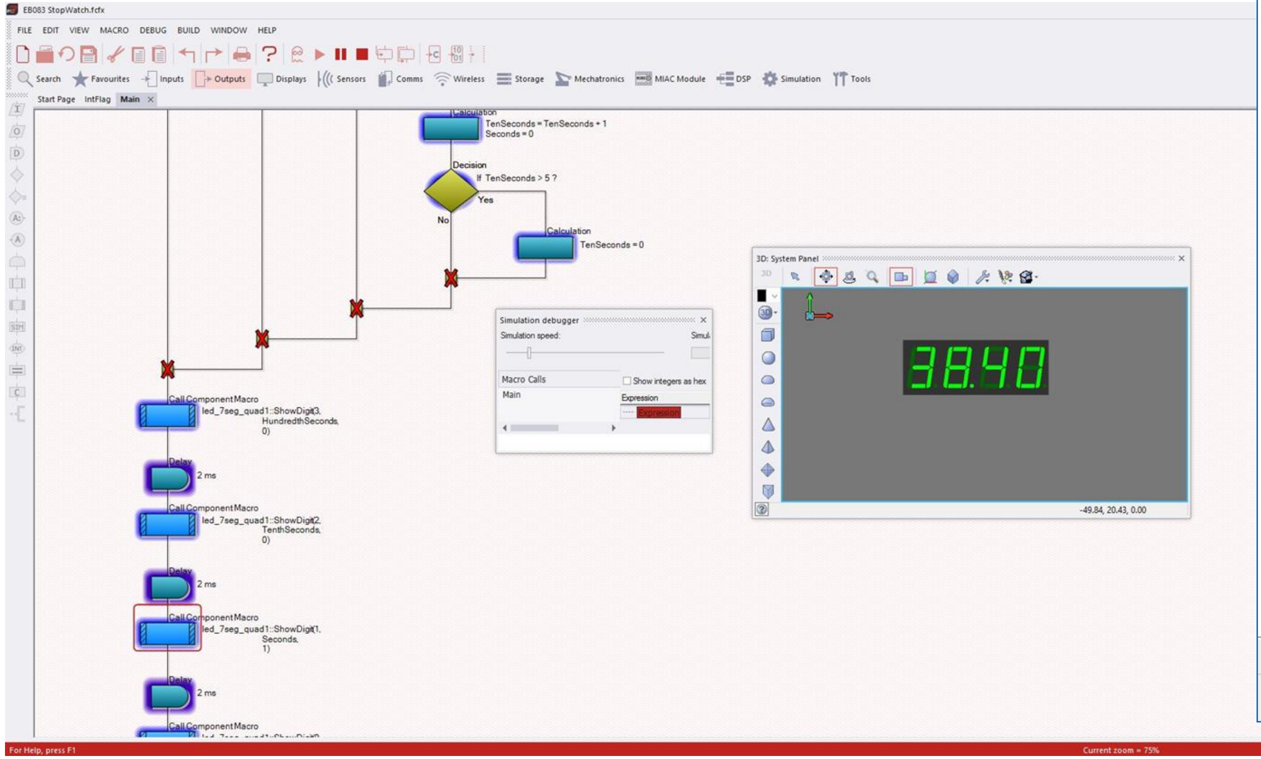
Paradigm: tile-based

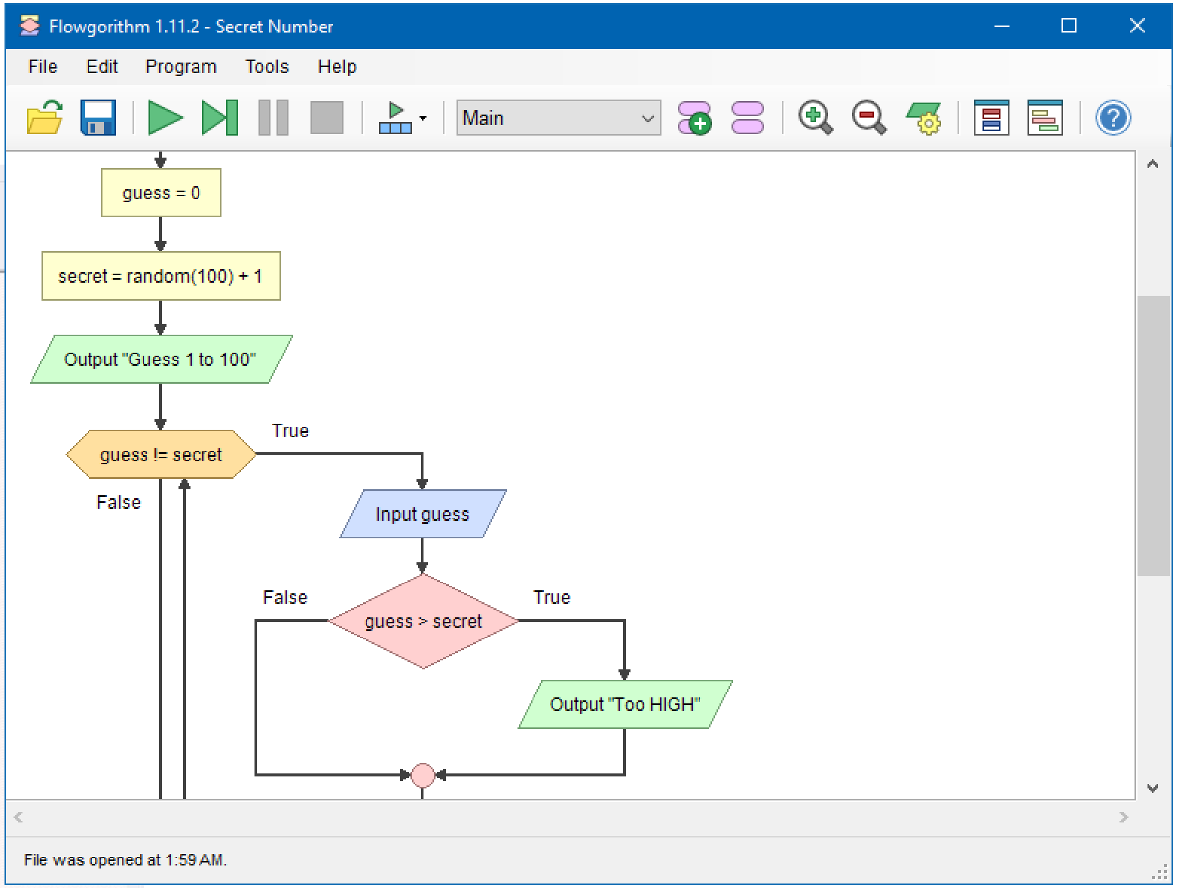


```
function main ()
    var zufallsZahl := math → random(100) + 1
    var eingabe := wall → ask number("")
    if eingabe < zufallsZahl then
        "Höher" → post to wall
    else if eingabe > zufallsZahl then
        "Niedriger" → post to wall
    else
        "Gratulation. Die Zahl wurde er..." → post to wall
    end if
end function
```

```
#include "stdlib.h"
#include "IRremote.h"
#include "pitches.h"
#include "Minibloq.h"


void setup()
{
    initBoard();

    while(true)
    {
        DigitalWrite(D13_LED, true);
        delay(300);
        DigitalWrite(D13_LED, false);
        delay(300);
    }
}

void loop()
{
}
```

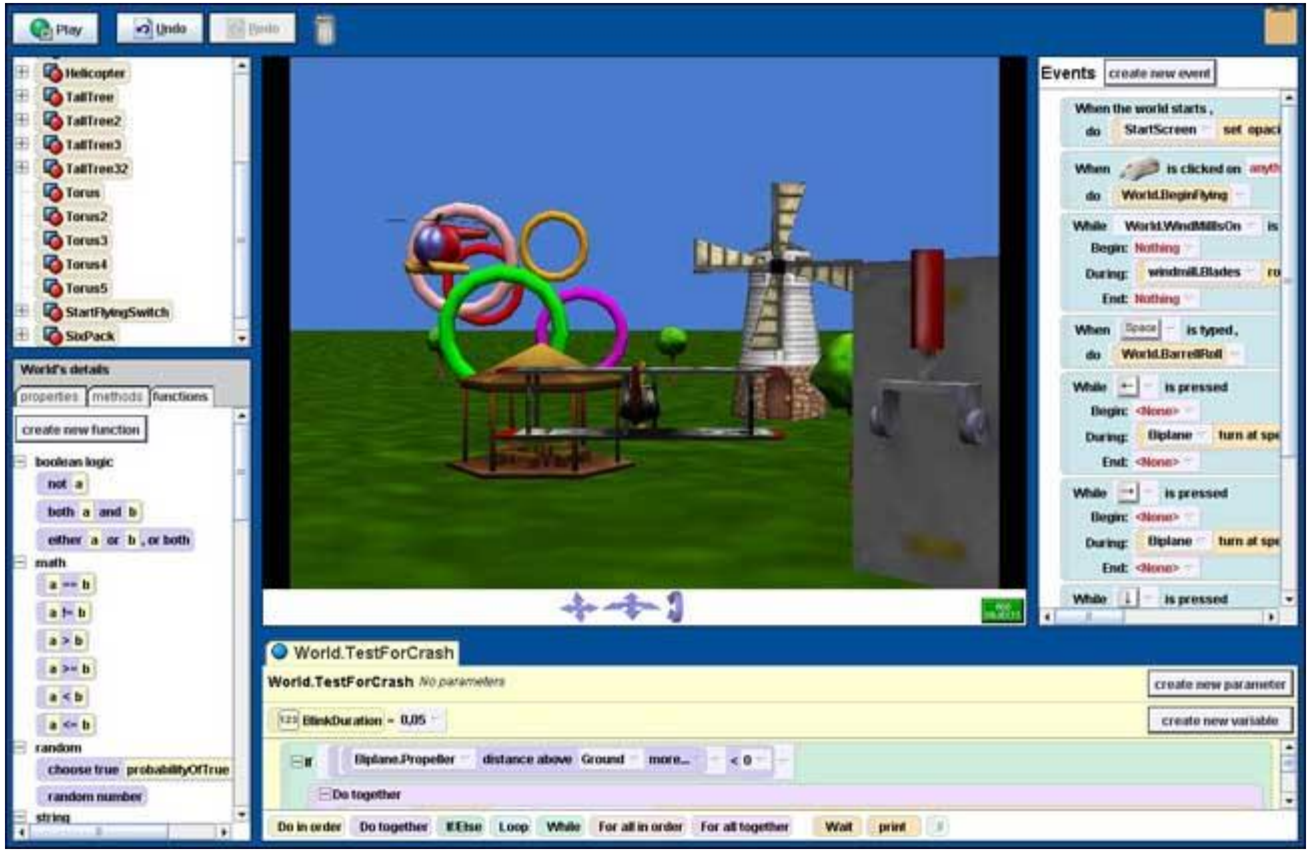
Paradigm: flowchart-based





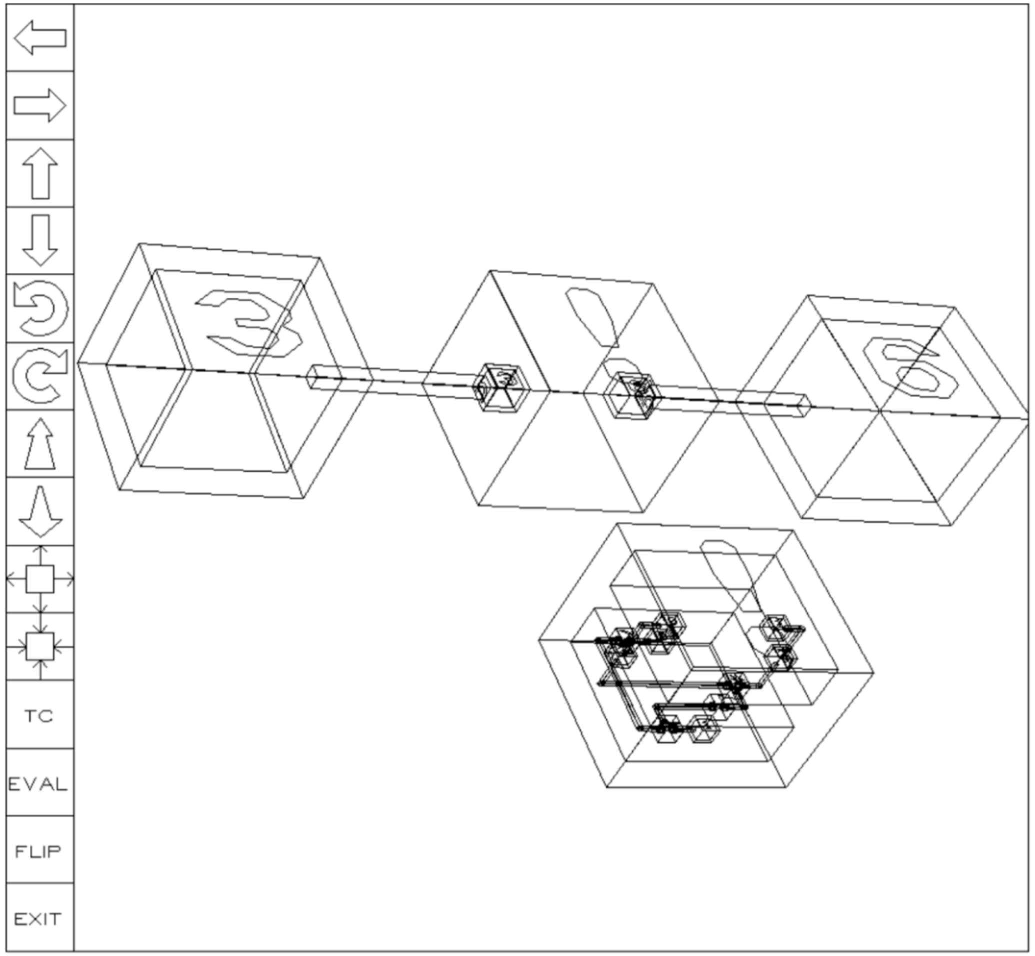
File was opened at 1:59 AM.

Paradigm: 3D programming

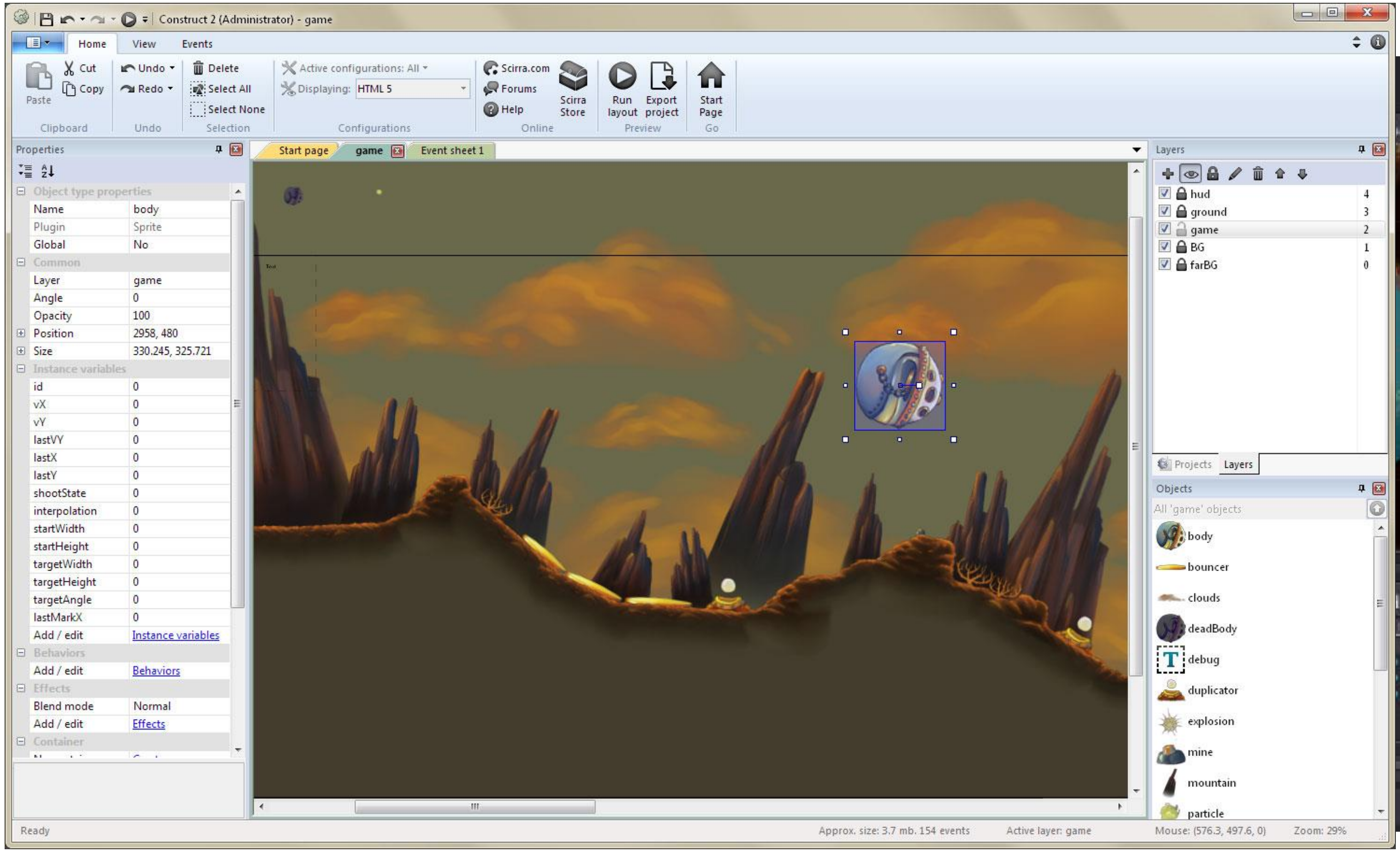


```
When the world starts ,
  do  StartScreen  set opaci

When  is clicked on anyth
  do  World.BeginFlying

While  World.WindMillsOn  is
  Begin: Nothing
  During:  windmill.Blades  ro
  End: Nothing

When  Space  is typed ,
  do  World.BarrelRoll

While  ←  is pressed
  Begin: <None>
  During:  Biplane  turn at spe
  End: <None>

While  →  is pressed
  Begin: <None>
  During:  Biplane  turn at spe
  End: <None>

While  ↓  is pressed
```

```
World.TestForCrash
World.TestForCrash No parameters
123 BlinkDuration = 0,05
If  Biplane.Propeller  distance above  Ground  more...  < 0
  Do together
```

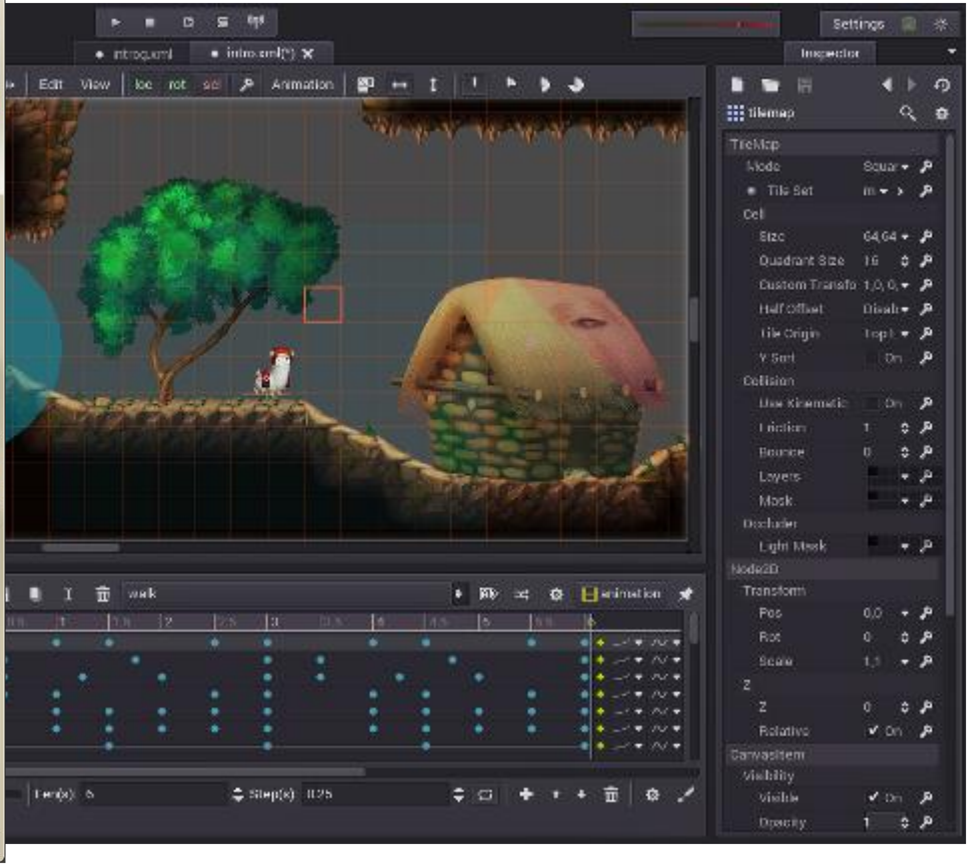


Paradigm: WYSIWYG editing



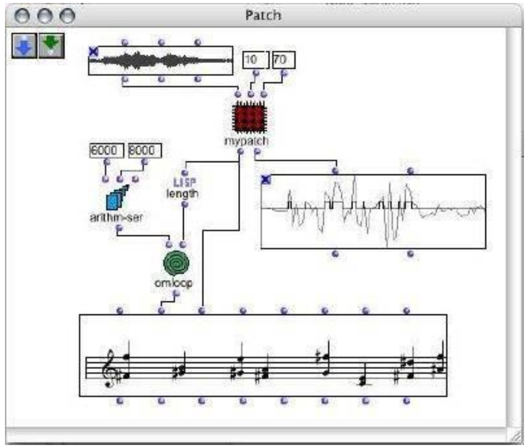
Object type properties	
Name	body
Plugin	Sprite
Global	No
Common	
Layer	game
Angle	0
Opacity	100
Position	2958, 480
Size	330.245, 325.721
Instance variables	
id	0
vX	0
vY	0
lastVY	0
lastX	0
lastY	0
shootState	0
interpolation	0
startWidth	0
startHeight	0
targetWidth	0
targetHeight	0
targetAngle	0
lastMarkX	0
Add / edit	Instance variables
Behaviors	
Add / edit	Behaviors
Effects	
Blend mode	Normal
Add / edit	Effects
Container	

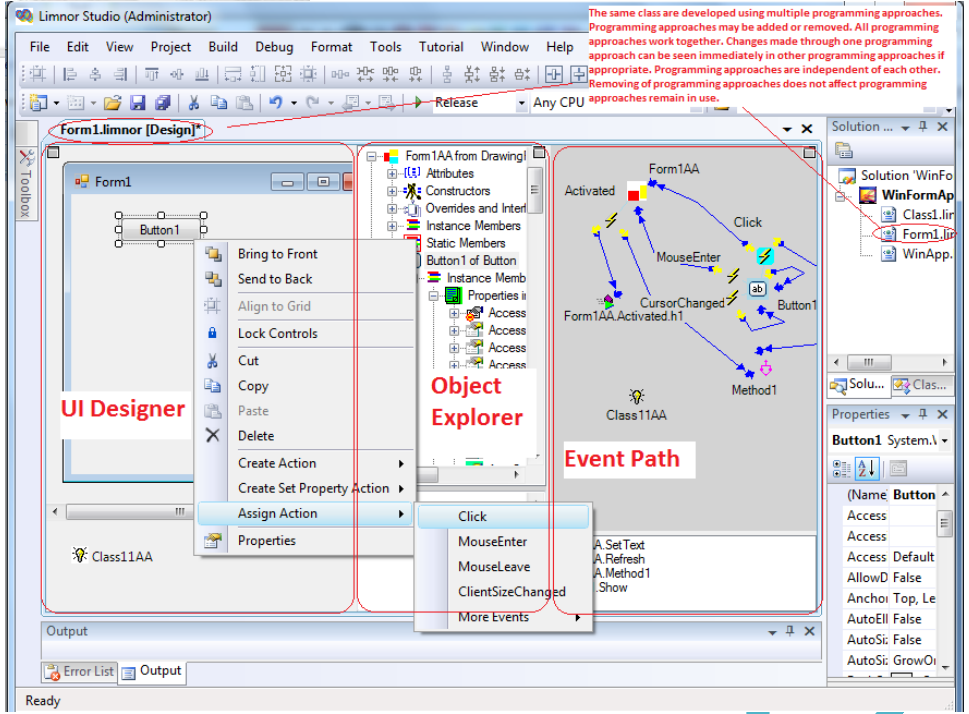
Ready
Approx. size: 3.7 mb. 154 events
Active layer: game
Mouse: (576.3, 497.6, 0)
Zoom: 29%



Purpose

- general-purpose
- multimedia processing
- user interface generation
- visualization
- simulation



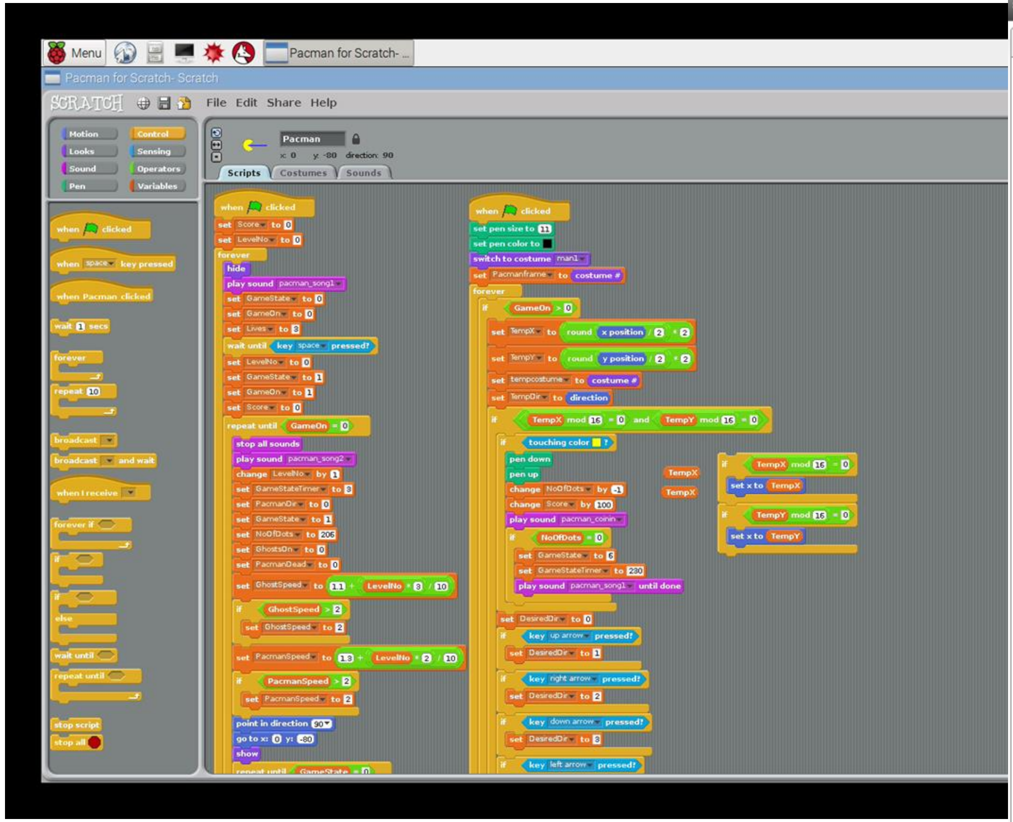


The same class are developed using multiple programming approaches. Programming approaches may be added or removed. All programming approaches work together. Changes made through one programming approach can be seen immediately in other programming approaches if appropriate. Programming approaches are independent of each other. Removing of programming approaches does not affect programming approaches remain in use.

Programming knowledge

- experienced programmers
- beginner programmers
- basic scripting
- non-programmers

Programming knowledge: beginner

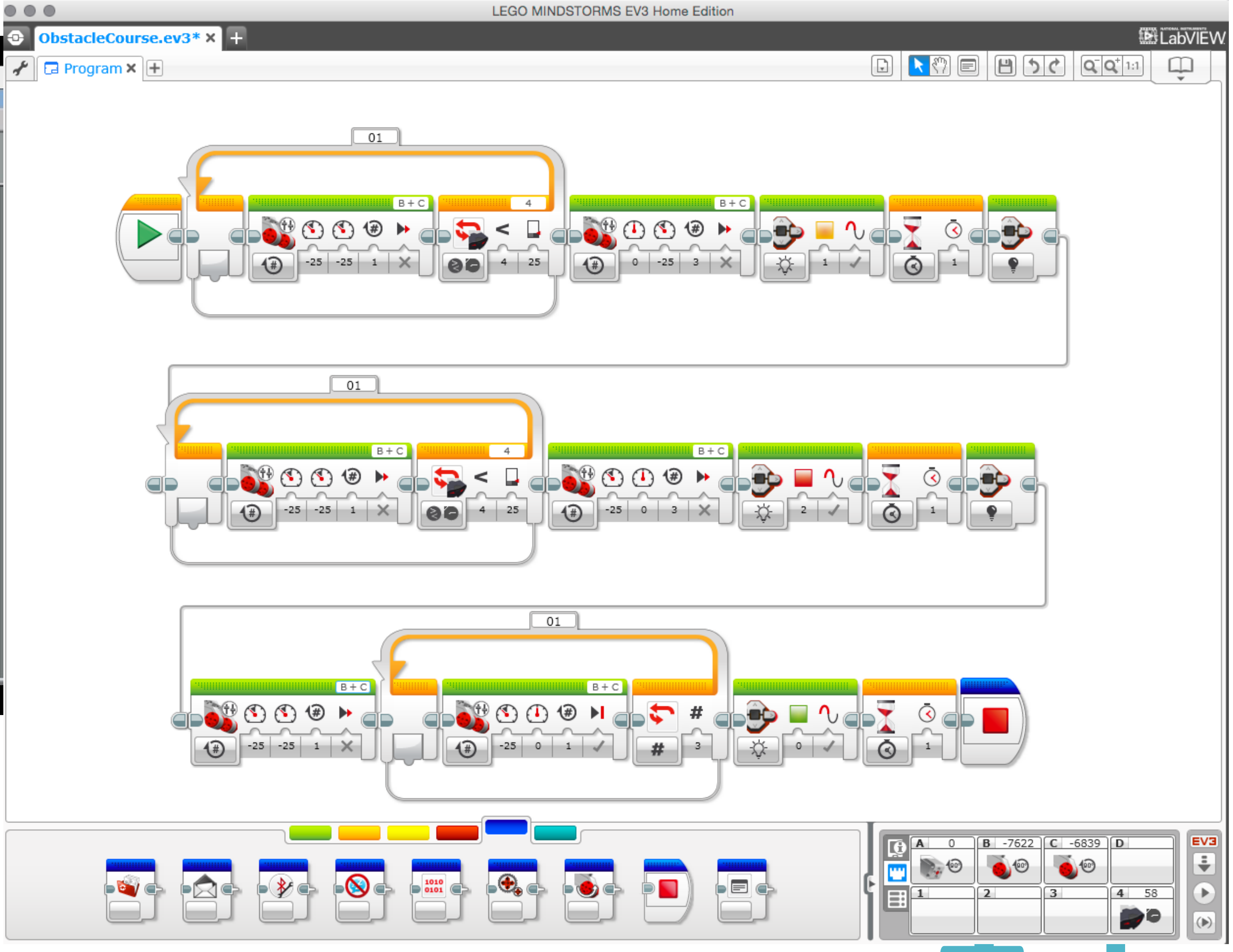


```
when [flag] clicked
set Score to 0
set LevelNo to 0
forever
  hide
  play sound pacman_song1
  set GameState to 0
  set GameOn to 0
  set Lives to 3
  wait until <key space pressed?>
  set LevelNo to 0
  set GameState to 1
  set GameOn to 1
  set Score to 0
  repeat until <GameOn = 0>
    stop all sounds
    play sound pacman_song2
    change LevelNo by 1
    set GameStateTimer to 3
    set PacmanDir to 0
    set GameState to 1
    set NoOfDots to 206
    set GhostsOn to 0
    set PacmanDead to 0
    set GhostSpeed to (1.1 + ((LevelNo * 3) / 10))
    if <GhostSpeed > 2>
      set GhostSpeed to 2
    set PacmanSpeed to (1.3 + ((LevelNo * 2) / 10))
    if <PacmanSpeed > 2>
      set PacmanSpeed to 2
    point in direction 90
    go to x: 0 y: -80
    show
    repeat until <GameState = 0>
```

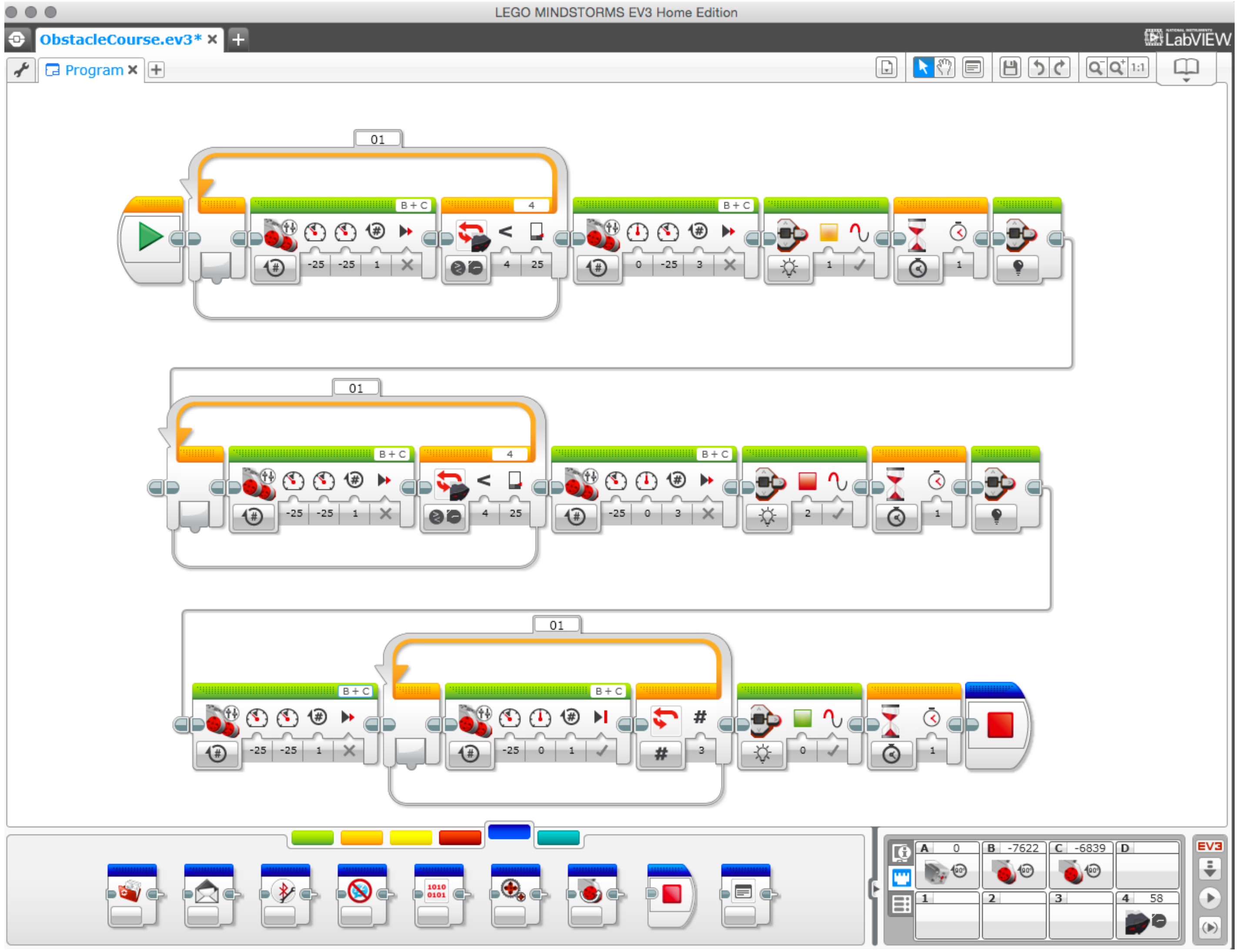
```
when [flag] clicked
set pen size to 11
set pen color to [■]
switch to costume man1
set Pacmanframe to costume #
forever
  if <GameOn > 0>
    set TempX to ((round (x position / 2)) * 2)
    set TempY to ((round (y position / 2)) * 2)
    set tempcostume to costume #
    set TempDir to direction
    if <<(TempX mod 16) = 0> and <(TempY mod 16) = 0>>
      if <touching color [■]?>
        pen down
        pen up
        change NoOfDots by -1
        change Score by 100
        play sound pacman_coinin
        if <NoOfDots = 0>
          set GameState to 6
          set GameStateTimer to 230
          play sound pacman_song1 until done
    set DesiredDir to 0
    if <key up arrow pressed?>
      set DesiredDir to 1
    if <key right arrow pressed?>
      set DesiredDir to 2
    if <key down arrow pressed?>
      set DesiredDir to 3
    if <key left arrow pressed?>
```

```
TempX
TempX
```

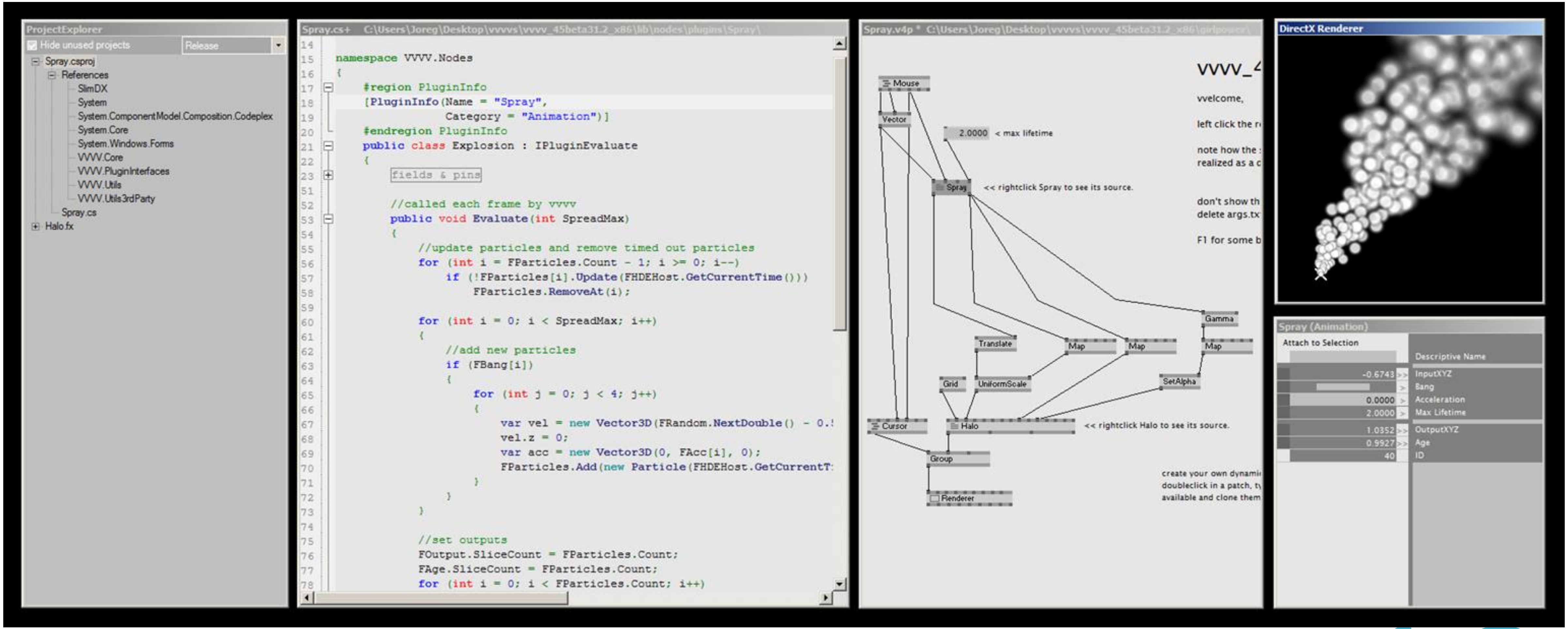
```
if <(TempX mod 16) = 0>
  set x to TempX
if <(TempY mod 16) = 0>
  set x to TempY
```



Amount of text code: just visual



Amount of text code: hybrid



ProjectExplorer

☑ Hide unused projects   Release

- Spray.csproj
  - References
    - SlimDX
    - System
    - System.ComponentModel.Composition.Codeplex
    - System.Core
    - System.Windows.Forms
    - VVVV.Core
    - VVVV.PluginInterfaces
    - VVVV.Utils
    - VVVV.Utils3rdParty
  - Spray.cs
- Halo.fx

Spray.cs+   C:\Users\Joreg\Desktop\vvvvs\vvvv_45beta31.2_x86\lib\nodes\plugins\Spray\

```
namespace VVVV.Nodes
{
    #region PluginInfo
    [PluginInfo(Name = "Spray",
                Category = "Animation")]
    #endregion PluginInfo
    public class Explosion : IPluginEvaluate
    {
        fields & pins

        //called each frame by vvvv
        public void Evaluate(int SpreadMax)
        {
            //update particles and remove timed out particles
            for (int i = FParticles.Count - 1; i >= 0; i--)
                if (!FParticles[i].Update(FHDEHost.GetCurrentTime()))
                    FParticles.RemoveAt(i);

            for (int i = 0; i < SpreadMax; i++)
            {
                //add new particles
                if (FBang[i])
                {
                    for (int j = 0; j < 4; j++)
                    {
                        var vel = new Vector3D(FRandom.NextDouble() - 0.!
                        vel.z = 0;
                        var acc = new Vector3D(0, FAcc[i], 0);
                        FParticles.Add(new Particle(FHDEHost.GetCurrentT:
                    }
                }
            }

            //set outputs
            FOutput.SliceCount = FParticles.Count;
            FAge.SliceCount = FParticles.Count;
            for (int i = 0; i < FParticles.Count; i++)
```

Spray.v4p * C:\Users\Joreg\Desktop\vvvvs\vvvv_45beta31.2_x86\girlpower\

< max lifetime

<< rightclick Spray to see its source.

<< rightclick Halo to see its source.

vvvv_4

vvelcome,

left click the r

note how the
realized as a c

don't show th
delete args.tx

F1 for some b

create your own dynami
doubleclick in a patch, t
available and clone them

DirectX Renderer

Spray (Animation)

Attach to Selection

		Descriptive Name
-0.6743	>>	InputXYZ
	>	Bang
0.0000	>	Acceleration
2.0000	>	Max Lifetime
1.0352	>>	OutputXYZ
0.9927	>>	Age
40		ID

Conclusion

- 4 dimensions:
  - paradigm & visual representation
  - purpose
  - programming knowledge
  - amount of text code