

Zastosowanie funkcyjnego paradygmatu do tworzenia graficznego środowiska programistycznego

Praca magisterska pod opieką dra hab. Dariusza Biernackiego

Łukasz Czapliński

Wydział Matematyki i Informatyki UWr

November 4, 2016

Plan

- 1 Omówienie problemu
- 2 Cel
- 3 Realizacja
- 4 Wyniki
- 5 Wnioski i wyzwania na przyszłość

Ekrany dotykowe jako główna metoda interakcji z komputerem



Rozwiązanie: programowanie przez diagramy

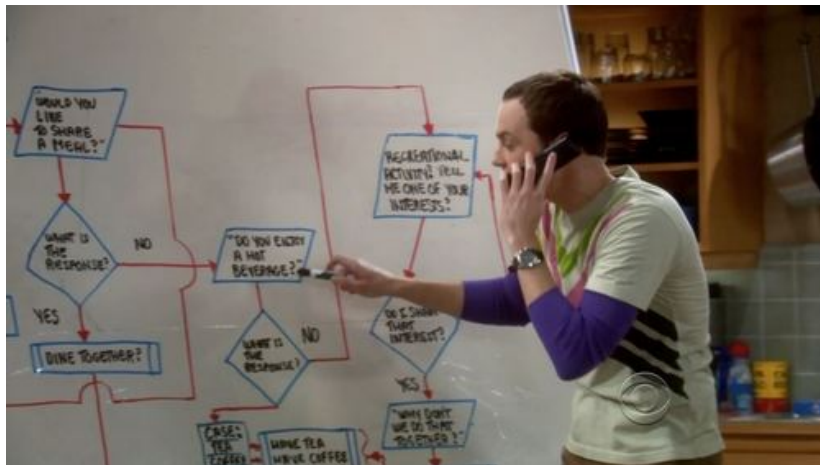
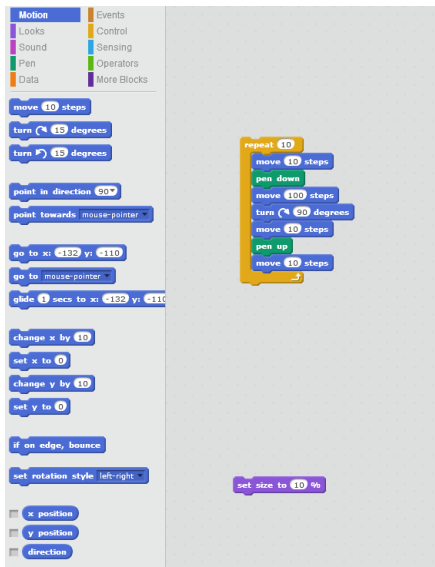


Figure: Copyright Warner Bros. Television

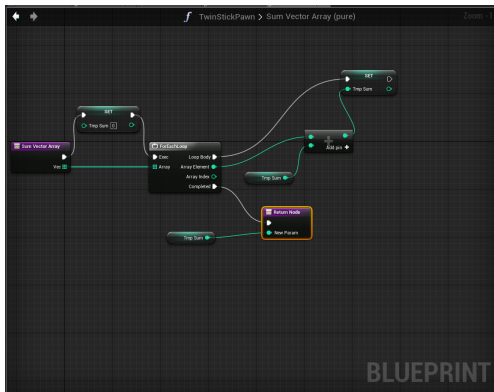
Graficzne języki programowania - Scratch

- Lifelong Kindergarten Group, MIT, 2005
- edukacyjny, dla dzieci w wieku 8-16 lat
- pozwala na tworzenie interaktywnych scen
- przypomina układanie puzzli



Graficzne języki programowania - Blueprints

- Unreal Engine, Epic, 2014
- pisanie poziomów gier
- dla developerów, nie programistów
- łączenie węzłów - druty symbolizują przepływ danych oraz wykonanie programu



Zalety

- Prostsze w nauce
- Doskonałe dla ekranów dotykowych
- Dopasowane do swoich nisz

Wady

- Brak graficznego języka programowania ogólnego zastosowania
- Egzotyczne sposoby zapisu

Przeszkody w zdobyciu popularności - Edytory

M	Filename	Length	Date	Time	File
-rw-rw-rw-		1725	5-Nov-2016	12:25:26	Project.json
-rw-rw-rw-		1263	5-Nov-2016	12:25:26	0.png
-rw-rw-rw-		6635	5-Nov-2016	12:25:26	1.svg
-rw-rw-rw-		6226	5-Nov-2016	12:25:26	2.svg
-rw-rw-rw-		1264	5-Nov-2016	12:25:26	3.png
-rw-rw-rw-		37420	5-Nov-2016	12:25:26	0.wav
-rw-rw-rw-		560	5-Nov-2016	12:25:26	1.wav
-rw-rw-rw-		55993			7 files


```
{
  "objName": "Stage",
  "sounds": [{
    "soundName": "pop",
    "soundID": 1,
    "md5": "83a9787d4cb6f3b7632b4ddfeb74367.wav",
    "sampleCount": 258,
    "rate": 11025,
    "format": ""
  }],
  "costumes": [{
    "costumeName": "backdrop1",
    "baseLayerID": 3,
    "baseLayerMD5": "739b5e2a2435f6e1ec2993791b423146.png",
    "bitmapResolution": 1,
    "rotationCenterX": 240,
    "rotationCenterY": 180
  }],
  "currentCostumeIndex": 0,
  "penLayerMD5": "5c81a336fab8be57adc039a8a2b33ca9.png",
  "penLayerID": 0,
  "tempoBPM": 60,
  "videoAlpha": 0.5,
  "children": [{
    "objName": "Sprite1",
    "scripts": [[45,
      80,
      [{"forward": 10}, {"turnRight": 15}, {"forward": 10}, {"turnLeft": 15}]],
    "sounds": [{
      "soundName": "meow",
      "soundID": 0,
      "md5": "83c36d086dc92327b9e7849a565c6bff.wav",
      "sampleCount": 18688,
      "rate": 22050,
      "format": ""
    }],
    "costumes": [{
      "costumeName": "costume1",
      "baseLayerID": 1,
      "baseLayerMD5": "09dc888b0b7df19f70d81588ae73420e.svg",
      "bitmapResolution": 1,
      "rotationCenterX": 47,
      "rotationCenterY": 55
    },
    {
      "costumeName": "costume2",
      "baseLayerID": 2,
      "baseLayerMD5": "3696356a03a8d938318676a593572843.svg",
      "bitmapResolution": 1,
      "rotationCenterX": 47,
      "rotationCenterY": 55
    }
  ]],
  "currentCostumeIndex": 0,
}
```





Przeszkody w zdobyciu popularności - Systemy kontroli wersji


Pliki projektu są wykrywane jako binarne.

```
diff --git a/project_p1.sb2 b/project_p1.sb2
new file mode 100644
index 0000000..39cc371
Binary files /dev/null and b/project_p1.sb2 differ
```






Przeszkody w zdobyciu popularności - Code Review

 **Open** scoiatael wants to merge 1 commit into scratch-cr-test from scratch-cr-test+b1





 Conversation **0**  Commits **1**  Files changed **2**

Changes from all commits ▾ 2 files ▾ **+3 -0** 

Unified Split **Review changes ▾**

3  README.md   View  

...	...	@@ -1,3 +1,6 @@
1	1	# Iteration 1 of Scratch project
2	2	* Added foo
3	3	* Moved 15 steps ahead at start
	4	+ # Iteration 2
	5	+* Added bar
	6	+* Rotated to mouse

BIN  project_p1.sb2  View  

Binary file not shown.

- Graficzne środowisko programistyczne
- Łatwa integracja z istniejącym systemem - znany język
- Łatwa integracja z istniejącym systemem - czytelny format zapisu
- Przygotowane do programowania na ekranach dotykowych
- Integracja z narzędziami dla programistów

```
(register-handler
  :repl-connected
  middlewares
  (fn [db ev]
    (-> db
      (assoc-in [:nrepl-connection] true)
      (r/update-suggestions ev))))

(defn- start-update-suggestions [{:keys [ns]}]
  (nrepl/functions! ns #(dispatch [:add-suggestion ns %])))
```

Realizacja: Technologia - Electron + Clojurescript

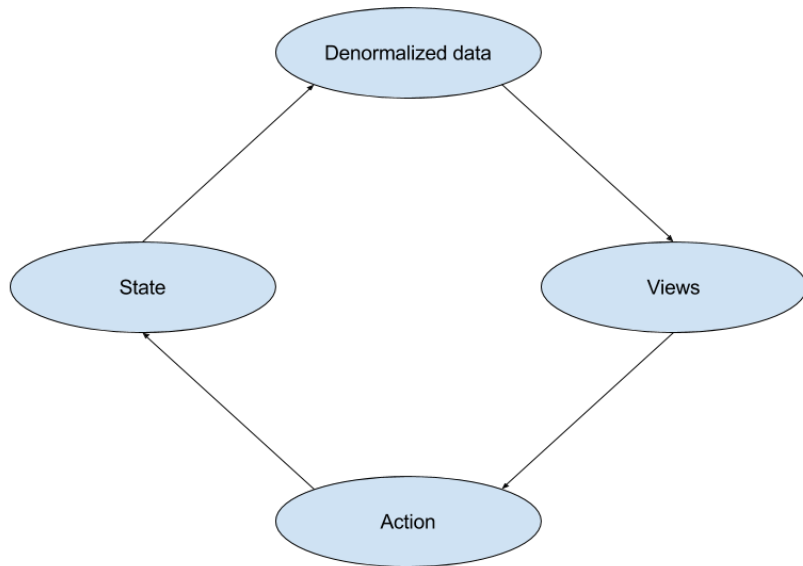


ELECTRON

Build cross platform desktop apps with web technologies

Formerly known as Atom Shell. Made with ♥ by GitHub.





Jarvis - reprezentacja

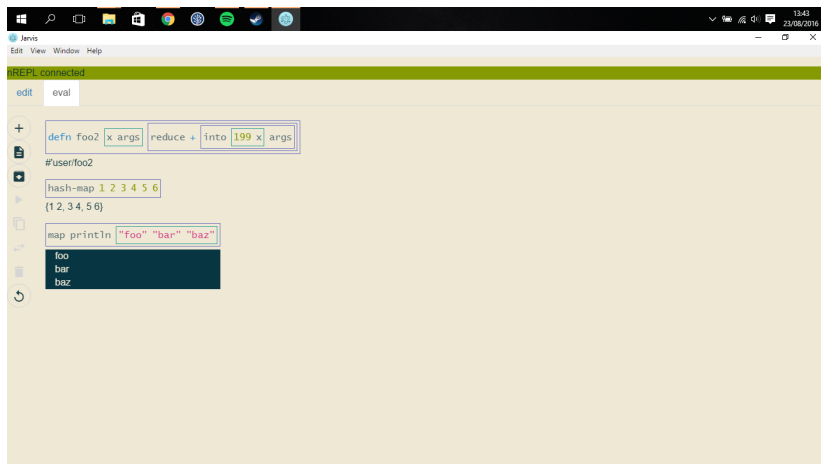
The screenshot displays the Jarvis REPL interface. At the top, a Windows taskbar shows various application icons and the system clock (13:42, 23/08/2016). The Jarvis window has a title bar and a menu bar (Edit, View, Window, Help). Below the menu bar, a green status bar indicates "REPL connected". The main editing area has tabs for "edit" and "eval". On the left side of the editing area, there is a vertical toolbar with icons for adding, deleting, undo, redo, and other editing actions. The code is entered in a light beige area and is visually structured with nested boxes:

- A top-level box contains the definition: `defn foo1 x + 100 x`.
- A second-level box contains the definition: `defn foo2 x args`.
- Inside the second-level box, a third-level box contains the expression: `reduce + into 199 x args`.
- A fourth-level box contains the definition: `defn foo3 a b`.
- Inside the fourth-level box, a fifth-level box contains the expression: `into hash-map 1 2 3 4 5 6 hash-map a b`.
- Below the fourth-level box, there is a standalone expression: `hash-map 1 2 3 4 5 6`.
- At the bottom, a box contains the expression: `map println "foo" "bar" "baz"`.

Jarvis - modyfikacja



Jarvis - testowanie



The screenshot shows a Windows taskbar at the top with various application icons. Below it is the Jarvis application window, which has a menu bar (Edit, View, Window, Help) and a title bar (Jarvis). The main area of the window is a REPL (Read-Eval-Print Loop) interface. It features a green header bar that says "REPL connected". Below this, there are two tabs: "edit" and "eval". The "eval" tab is active. The main area contains a Clojure code snippet:

```
defn foo2 [x args] (reduce + (into [199 x] args))
```

 followed by a comment `#user/foo2`. Below the code, there are two lines of code:

```
hash-map 1 2 3 4 5 6
```

 and

```
{1 2, 3 4, 5 6}
```

. At the bottom, there is a line of code:

```
map println ["foo" "bar" "baz"]
```

. The output of this code is displayed in a dark blue box:

```
foo  
bar  
baz
```

. The Jarvis application window is running on a Windows system, as indicated by the taskbar and the system clock in the top right corner showing 13:43 on 23/08/2016.

- Brak wsparcia dla wszystkich struktur Clojure
- Brak wsparcia dla makr
- Brak możliwości cofnięcia akcji

- Inne reprezentacja kodu
- Bardziej płynna, intuicyjna interakcja z użytkownikiem
- Ulepszony system modyfikacji kodu (drag & drop)
- Szersza integracja z systemem wykonywania kodu