

Zastosowanie funkcyjnego paradygmatu do tworzenia graficznego środowiska programistycznego

Praca magisterska pod opieką dra hab. Dariusza Biernackiego

Łukasz Czapliński

Wydział Matematyki i Informatyki UWr

November 1, 2016

Outline

- 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

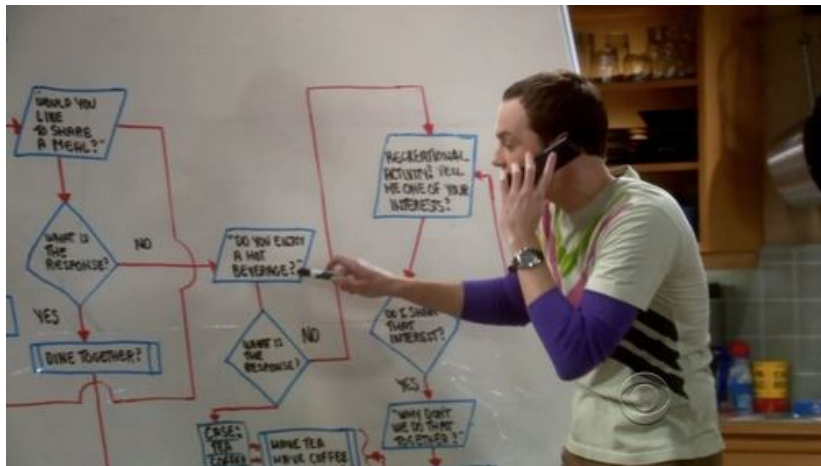
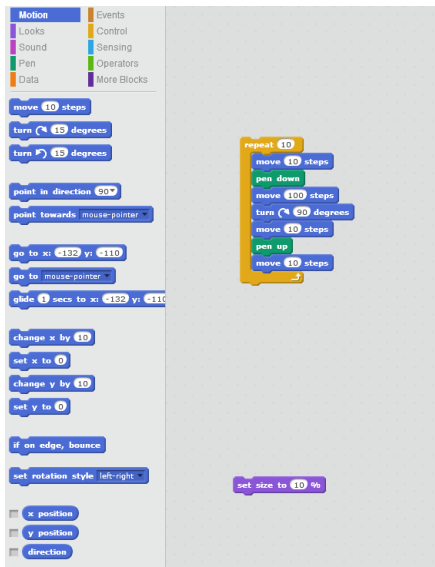


Figure: Copyright Warner Bros. Television

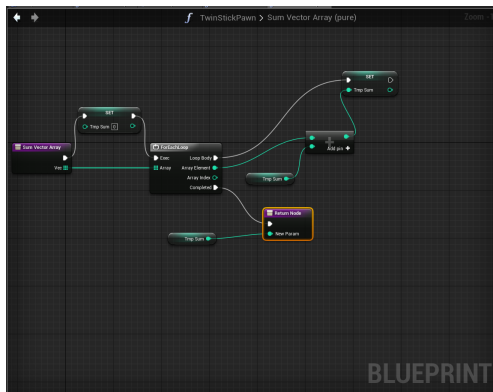
Graficzne języki programowania - Scratch

- 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



Zalety

- Prostsze w nauce
- Doskonałe dla ekranów dotykowych

Wady

- Binarny sposób zapisu

Przeszkody w zdobyciu popularności - Edytory

Przeszkody w zdobyciu popularności - Systemy kontroli wersji

Przeszkody w zdobyciu popularności - Code Review

- 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

Realizacja: Język - Clojure

Realizacja: Technologia - Electron + Clojurescript

Realizacja: Integracja z REPLem

- Brak wsparcia dla wszystkich struktur Clojure
- Brak wsparcia dla makr

Możliwe ulepszenia

- Inne reprezentacja kodu
- Lepsza interakcja z użytkownikiem