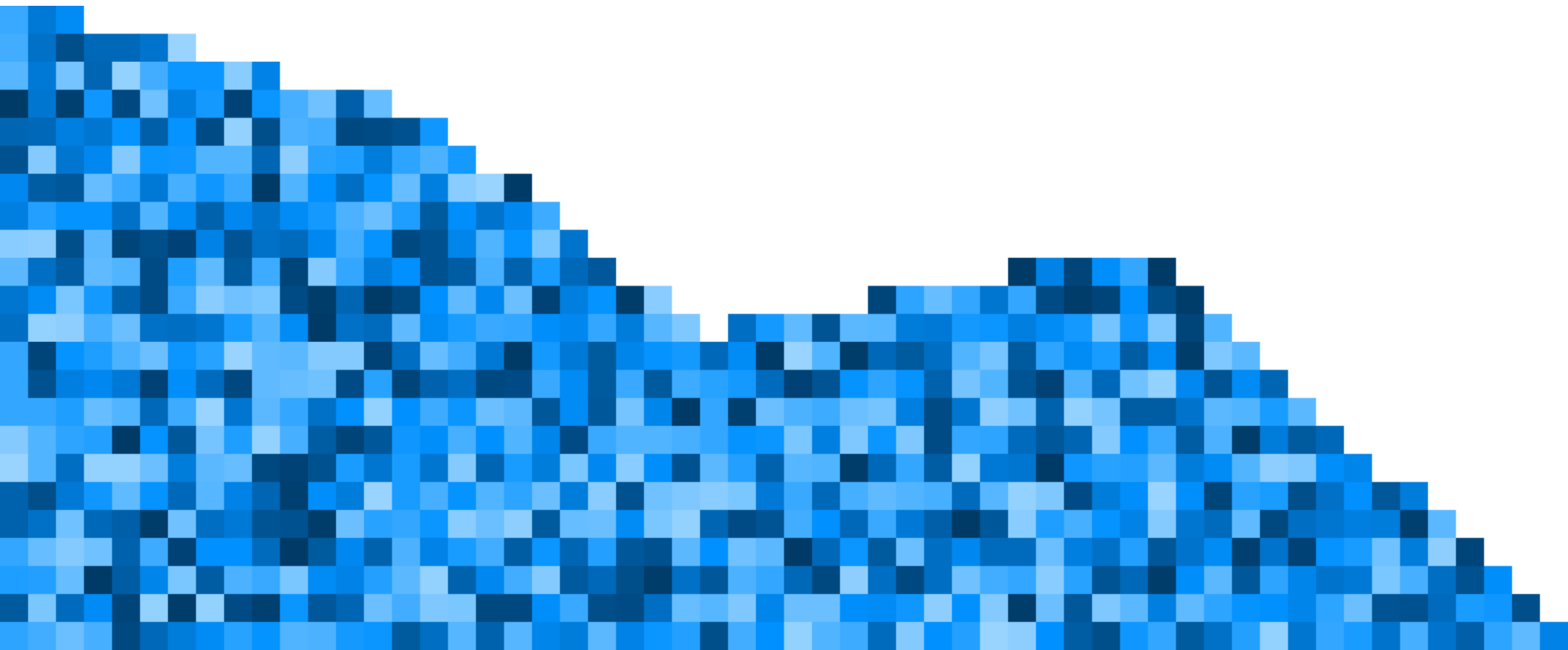


Vizuální programování

Adam Zmrzlý

Seminář LaSArIS, 28. 11. 2013



Obsah

- Motivace
- Co je vizuální programování
- Novost přístupu
- Cíle vizuálního programování
- Různé přístupy
- Jazyk Shades

Motivace

- Zvýrazňování syntaxe
 - pro různé jazyky
 - v různých IDE

V praxi neexistuje jednotná norma

- stejný jazyk v různých IDE
- stejné konstrukty jednotlivých jazyků v rámci jednoho IDE

Motivace

Java v Netbeans 6+

```
if (!(this.type).equals(other.type)) {
    return false;
}
if (!Arrays.equals(this.data, other.data)) {
    return false;
}
return true;
}
```

@Override

```
public String toString() {
    final String dataSize = "" + ((data == null) ? "no data" : data.l
    return "Token (Luminance: " + type.getLuminance() + ", Alpha: " +
}
```

Motivace

JavaScript v Netbeans 6+

```
/**
 * Enables notifications according to user's preferences. This method
 */
initNotifications : function() {
    let prefs = ise.prefs;

    if(prefs.getBoolPref("notifications")) {
        if(typeof ise.timeout == "number") {
            clearTimeout(ise.timeout);
        }
        let interval = prefs.getIntPref("interval");
        if(interval < 5) {
            interval = 5;
            prefs.setIntPref("interval", 5);
        }
        ise.timeout = setTimeout(ise.notify, interval * 60 * 1000);
    }
},
```

Motivace

JavaScript v Aptana Studio 3+

```
/**
 * Enables notifications according to user's preferences. This method
 */
initNotifications : function() {
    let prefs = ise.prefs;

    if(prefs.getBoolPref("notifications")) {
        if(typeof ise.timeout == "number") {
            clearTimeout(ise.timeout);
        }
        let interval = prefs.getIntPref("interval");
        if(interval < 5) {
            interval = 5;
            prefs.setIntPref("interval", 5);
        }
        ise.timeout = setTimeout(ise.notify, interval * 60 * 1000);
    }
},
```

Motivace

Zvýrazňování syntaxe by mělo pomáhat programátorům pracovat efektivněji

- subjektivní → možnost úpravy
- nenese žádnou dodatečnou informaci, neukládá se, často ani nelze zkopírovat
- zvýraznění syntaxe logicky seskupuje jednotlivé tokeny kódu (velmi hrubozrnně)
- zkusme se na kód se zvýrazněnou syntaxí podívat z jiného úhlu

Motivace

```
@Override
public String toString() {
    final String dataValue = data.toString();
    return "Token: Luminance: " + type.getLuminance()
}
```


Motivace

```
@Override
public String toString() {
    final String dataValue = data.toString();
    return "Token: Luminance: " + type.getLuminance()
}
```



Co je vizuální programování

- Nové paradigma?

Co je vizuální programování

- Nové paradigma?
 - ne, vizuální programování lze použít pro paradigma objektově orientované, funkcionální, logické, procedurální, ...

Co je vizuální programování

- Nové paradigma?
 - ne, vizuální programování lze použít pro paradigma objektově orientované, funkcionální, logické, procedurální, ...
- Nový programovací jazyk?

Co je vizuální programování

- Nové paradigma?
 - ne, vizuální programování lze použít pro paradigma objektově orientované, funkcionální, logické, procedurální, ...
- Nový programovací jazyk?
 - ne, může existovat prakticky neomezené množství různých vizuálních jazyků

Co je vizuální programování

- Nové paradigma?
 - ne, vizuální programování lze použít pro paradigma objektově orientované, funkcionální, logické, procedurální, ...
- Nový programovací jazyk?
 - ne, může existovat prakticky neomezené množství různých vizuálních jazyků
 - vizuální varianty stávajících jazyků

Co je vizuální programování

- Nové paradigma?
 - ne, vizuální programování lze použít pro paradigma objektově orientované, funkcionální, logické, procedurální, ...
- Nový programovací jazyk?
 - ne, může existovat prakticky neomezené množství různých vizuálních jazyků
 - vizuální varianty stávajících jazyků, nový člen .NET rodiny, Visual Java? :)

Co je vizuální programování

Vizuální programování je způsob tvorby, reprezentace a zpracování zdrojového kódu.

- Zdrojový kód v grafické podobě
 - důraz kladen na vizuální podobu vytvářených programů
 - zcela odlišný přístup k programování
→ nové možnosti, využití i výzvy

Novost přístupu

Představuje vizuální programování zcela nový přístup k reprezentaci dat a jejich následnému zpracování?



Novost přístupu

Představuje vizuální programování zcela nový přístup k reprezentaci dat a jejich následnému zpracování?

- čárové kódy
 - pro nás nepříliš zajímavé
 - slouží ke kódování a dekódování krátkého řetězce znaků (obvykle identifikátor)
 - určeno pro tisk a následné optické rozpoznávání → různá omezení (barevnost, délka kódované informace, ...)

Novost přístupu

- čárové kódy
 - první snahy o estetičtější přístup k vizuálnímu uchovávání dat

Novost přístupu

- čárové kódy
 - první snahy o estetičtější přístup k vizuálnímu uchovávání dat



Novost přístupu

- „2D“ čárové kódy
 - evoluce původních čárových kódů
 - data kódována do čtvercových obrázků
 - zpočátku sloužily stále jen k vizuálnímu kódování a dekódování dat (odkazy, vizitky, ...)



Novost přístupu

- „2D“ čárové kódy
 - využití redundance a opravných kódů přineslo původně nezamýšlené možnosti úpravy vzhledu jednotlivých kódů

Novost přístupu

- „2D“ čárové kódy
- využití redundance a opravných kódů přineslo původně nezamýšlené možnosti úpravy vzhledu jednotlivých kódů



Novost přístupu

- „2D“ čárové kódy
 - využití redundance a opravných kódů přineslo původně nezamýšlené možnosti úpravy vzhledu jednotlivých kódů



Novost přístupu

- „2D“ čárové kódy
 - využití redundance a opravných kódů přineslo původně nezamýšlené možnosti úpravy vzhledu jednotlivých kódů

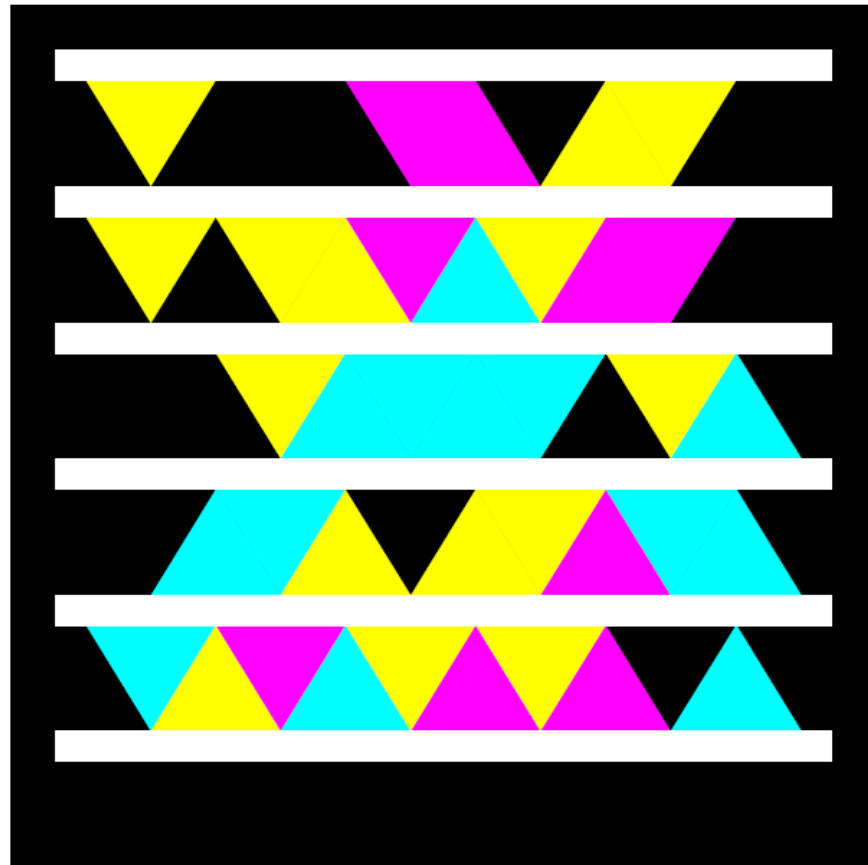


Novost přístupu

- „2D“ čárové kódy
 - High Capacity Color Barcode
 - vytvořen společností Microsoft
 - v samotné specifikaci využívá barev a místo čtverců trojúhelníky → vyšší kapacita oproti jiným kódům
 - existuje jediná implementace – Microsoft Tag, oznámeno ukončení podpory v roce 2015

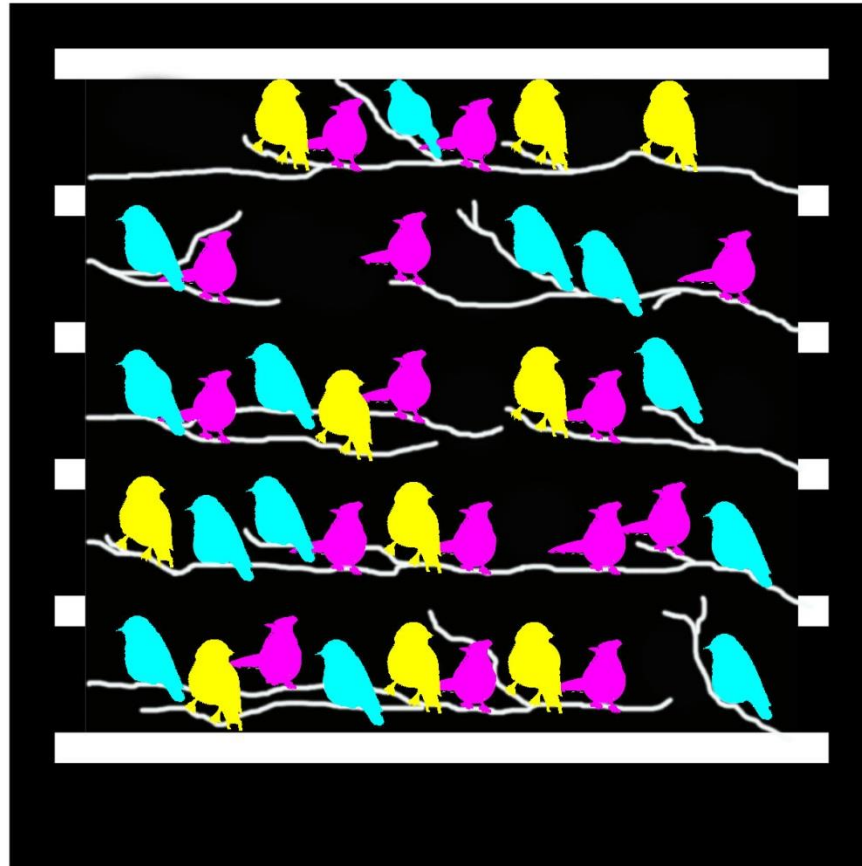
Novost přístupu

- „2D“ čárové kódy
 - High Capacity Color Barcode



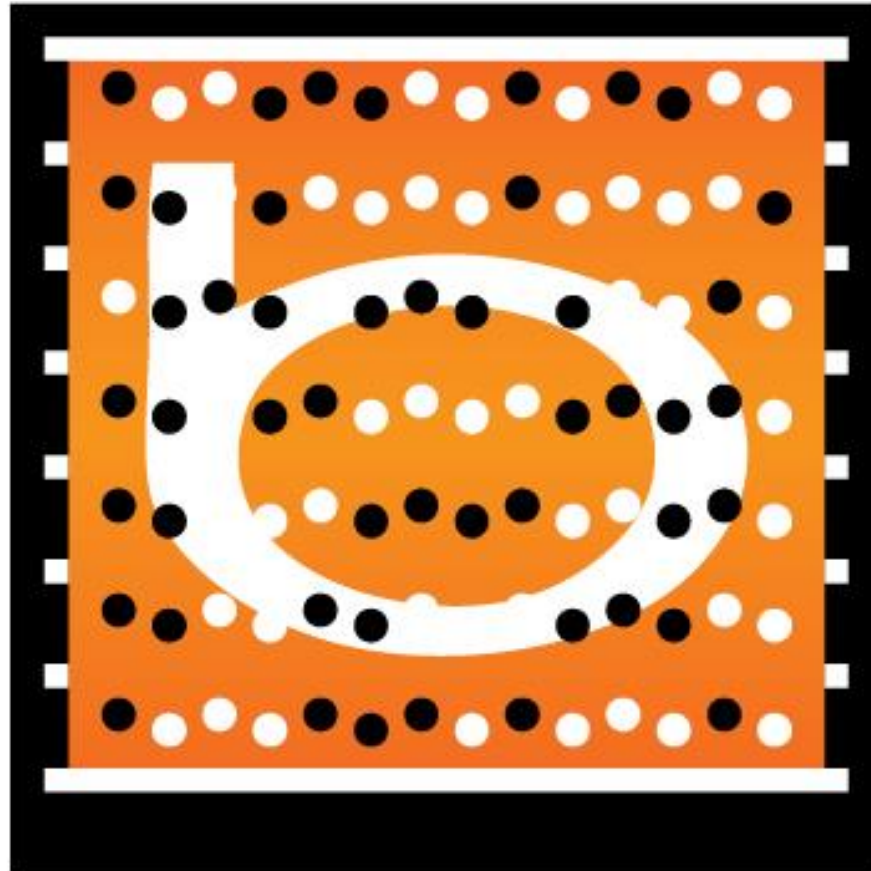
Novost přístupu

- „2D“ čárové kódy
 - High Capacity Color Barcode



Novost přístupu

- „2D“ čárové kódy
 - High Capacity Color Barcode



Novost přístupu

Stále se ale bavíme jen o uchovávání dat ve vizuální podobě pomocí čárových kódů, kde je ono slibované vizuální programování?

Novost přístupu

Stále se ale bavíme jen o uchovávání dat ve vizuální podobě pomocí čárových kódů, kde je ono slibované vizuální programování? Odpověď je blíž, než by se mohlo zdát. Spíše blíž, než bychom možná chtěli.

Novost přístupu

Stále se ale bavíme jen o uchovávání dat ve vizuální podobě pomocí čárových kódů, kde je ono slibované vizuální programování? Odpověď je blíž, než by se mohlo zdát. Spíše blíž, než bychom možná chtěli.



Obyčejný QR kód?

Novost přístupu

Stále se ale bavíme jen o uchovávání dat ve vizuální podobě pomocí čárových kódů, kde je ono slibované vizuální programování? Odpověď je blíž, než by se mohlo zdát. Spíše blíž, než bychom možná chtěli.



Obyčejný QR kód?

V podstatě ano, obsahuje ale kód v JavaScriptu. Některé čtečky jej i dnes automaticky interpretují.

Novost přístupu

Existují v současné době nějaké vizuální jazyky?



Novost přístupu

Existují v současné době nějaké vizuální jazyky?

V omezené míře ano, jejich počet je ale velmi malý (méně než 10). Autoři těchto jazyků je řadí mezi čistě ezoterické, pojem vizuální jazyk se nepoužívá.

Expresivní síla těchto jazyků je obvykle nízká.

- *Mobius*
 - prochází pixely, porovnává hodnoty RGB kanálů a na jejich základě provádí akce

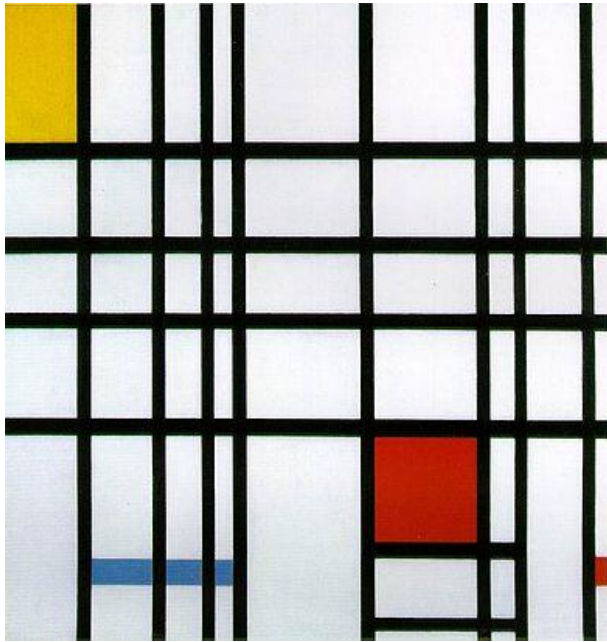
Novost přístupu

Nejpokročilejší z existujících jazyků je *Piet*

- interpretovaný, používá zásobníkovou architekturu, je Turing-complete
- pracuje s omezenou škálou 20 barev rozdělených do cyklů podle světlosti a odstínu
- používá dva ukazatele pro pohyb v obrázku
- při průchodu barevnou oblastí sčítá pixely
- při přechodu do jiné oblasti v závislosti na změně barvy a cyklu provede konkrétní operaci

Novost přístupu

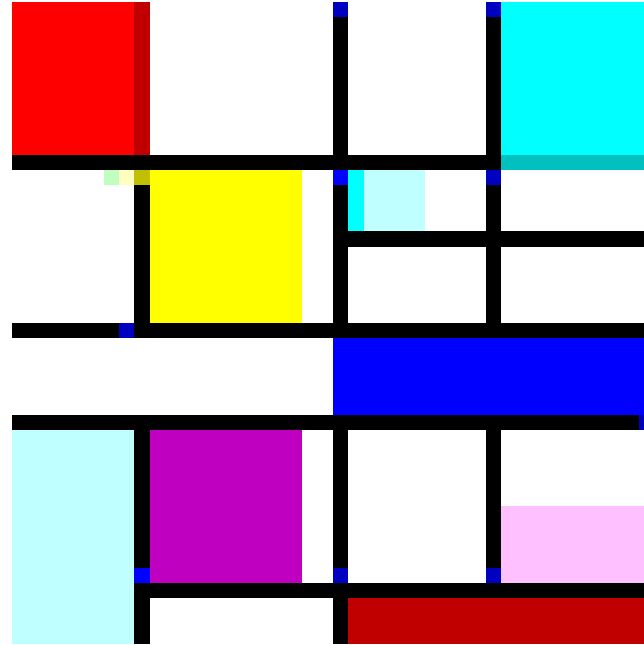
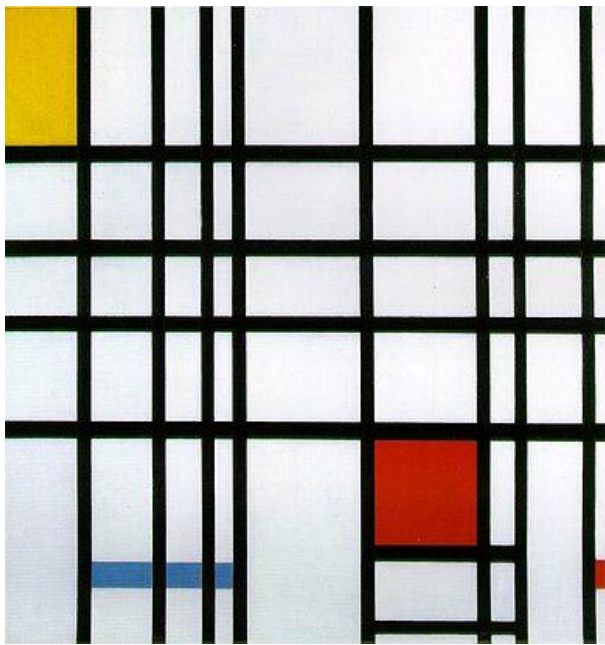
Pojmenován po Pietu Mondrianovi, jednom ze zakladatelů geometrického abstraktního umění. Programy by měly být psány tak, aby toto umění připomínaly.



Kompozice se žlutou, modrou a červenou, 1937-1942.
Vystaveno v Tate Gallery.

Novost přístupu

Pojmenován po Pietu Mondrianovi, jednom ze zakladatelů geometrického abstraktního umění. Programy by měly být psány tak, aby toto umění připomínaly.



Srovnání:
Program
v jazyce *Piet*.

Novost přístupu

V současné době neexistuje žádný jazyk, který by bylo možné prohlásit za použitelný (ve smyslu složitějších aplikací) a zároveň za zástupce vizuálního programování. Chybí zejména podpora pro programování (IDE, debuggery, profillery, ...)

Novost přístupu

V současné době neexistuje žádný jazyk, který by bylo možné prohlásit za použitelný (ve smyslu složitějších aplikací) a zároveň za zástupce vizuálního programování. Chybí zejména podpora pro programování (IDE, debuggery, profily, ...)

Tedy zatím... :)

Cíle vizuálního programování

- umožnit vytvářet obrázky s přidanou hodnotou
- abstrahovat od použití konkrétního přirozeného jazyka
- snaha o platformní nezávislost výsledných programů (u samotných obrázků zaručeno), záleží na interpretech
- přinést nový pohled na programování, důraz kladen na estetičnost

Cíle vizuálního programování

- přivést k programování designéry, grafiky..., ale zároveň neodradit programátory
- umožnit používat běžné konstrukty (`if`, `while`, `switch`, ...) spolu s novými možnostmi řízení toku („2D programování“)

Různé přístupy: varianta 1

- zdrojový kód zapsán (nakreslen) ve formě obrázku a následně přeložen do spustitelné podoby
- ztrácí se základní myšlenka vizuálního programování, spíše ezoterický přístup
- ztráta platformní nezávislosti, nutnost vytvořit minimálně back-endy překladačů pro více operačních systémů a platforem
- rychlejší běh programů, skrytý zdrojový kód

Různé přístupy: varianta 2

- zdrojový kód zapsán jako text a přeložen do podoby interpretovatelného obrázku
 - základní myšlenka vizuálního programování částečně zachována
 - vstupní programovací jazyk může být zcela nový, lze ale použít libovolný již existující
 - nutné řešit mapování konkrétního jazyka do podoby obrázku (anotace, metadata), poměrně složité

Různé přístupy: varianta 2

- zdrojový kód zapsán jako text a přeložen do podoby interpretovatelného obrázku
- musí existovat překladač jazyka do obrázku pro každý takto použitý jazyk
- musí existovat interpret obrázků (obdoba VM), případně dekompilátor obrázku zpět do původního jazyka a jeho následné zpracování konkrétním interpretem (překladačem), podobně jako v rámci platformy .NET

Různé přístupy: varianta 2

- zdrojový kód zapsán jako text a přeložen do podoby interpretovatelného obrázku
- výhodou je možnost použít známé jazyky a již existující nástroje (IDE, profilery, ...)
- nevýhodou je složité mapování kódu do podoby požadovaného obrázku, pomalejší zpracování než v případě první varianty

Různé přístupy: varianta 3

- zdrojový kód je nakreslen přímo jako interpretovatelný obrázek
- plně splňuje myšlenky vizuálního programování
- vyžaduje vizuální programovací jazyk a interpret
- přináší prakticky neomezené možnosti použití v závislosti na daném jazyku
- zatím neexistuje žádný takový jazyk ani vývojové nástroje

Různé přístupy: varianta 3

- zdrojový kód je nakreslen přímo jako interpretovatelný obrázek
- podobně jako varianta 2 je pomalejší (samotná režie zpracování obrázku a jeho interpretace)

Shades

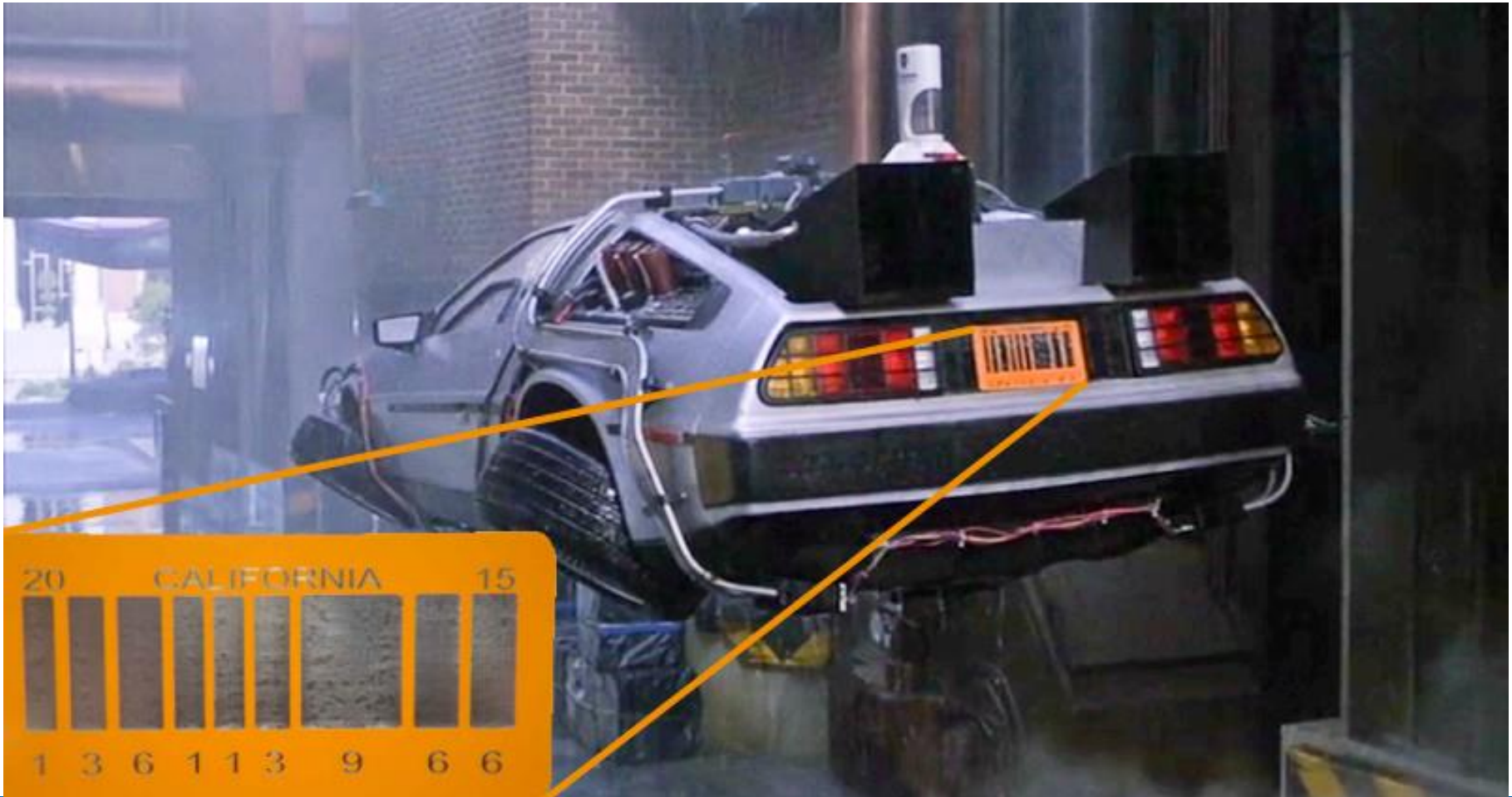
- připravovaný vizuální programovací jazyk (momentálně uzavřené testování)
- spolu s ním připravována sada vývojových nástrojů
- programy v jazyce Shades cílí na možnost co nejvíce napodobit libovolný obrázek
→ uplatnění např. v rámci korporátní identity
- využívá velké množství barev, spíše nevhodné pro tisk
- možnost vložit kód do existujícího obrázku

Shades

- pro dosažení co nejlepší aproximace existujícího obrázku použít downsampling (obdobně jako při antialiasingu na GPU)
- možnost použít umělecké ztvárnění kódu (zvětšení jednotlivých barevných ploch bez ovlivnění sémantiky – mozaikový efekt), použito v jednotlivých snímcích této prezentace

Závěr

Vizuální zpracování dat a programování má budoucnost.





Děkuji za pozornost