

ぼくらのプログラミングから、

From “programmers are people, too”

みんなのプログラミングへ

to “people are programmers.”

国立研究開発法人 産業技術総合研究所

加藤 淳

<http://junkato.jp/ja> (@arcatdmz)

2015/9/20 第48回 情報科学若手の会 若手特別講演

digitalmuseum

<http://digitalmuseum.jp>

'00-

- 主宰; 高校の頃から運営
- フリーウェアやデザインした作品を展示
- 友人との創作活動の拠点
- 産学交流の場づくり



{ぼくらの→みんなの}プログラミング

立花隆ゼミ (SCI/サイ)

<http://sci.gr.jp/>

- 幹事、デザイナー、プログラマ; 学部生活を捧げた
- ジャーナリストの立花隆氏と学部生で作るゼミ
- 最先端の科学技術に関する取材記事を配信
- 企画→参加募集→取材→執筆→校正→Web公開
- 誰でも記事をWebページ化できるシステムの開発



おうちハック

<http://kadecot.net/ouch-hack/>

- 実践中! (リズム家電, なめこカメラ, ...)
- 画像処理のためのビジュアル環境を研究



{ぼくらの→みんなの}プログラミング

OpenPool

<http://openpool.cc/>

- 元メインプログラマ; 一昨年グッドデザイン賞
- 大学至近の学生向けカフェ発、ビリヤード台を拡張する企画
- オープンソースでSDK公開
- API設計、独自ハード用ライブラリ実装などを担当

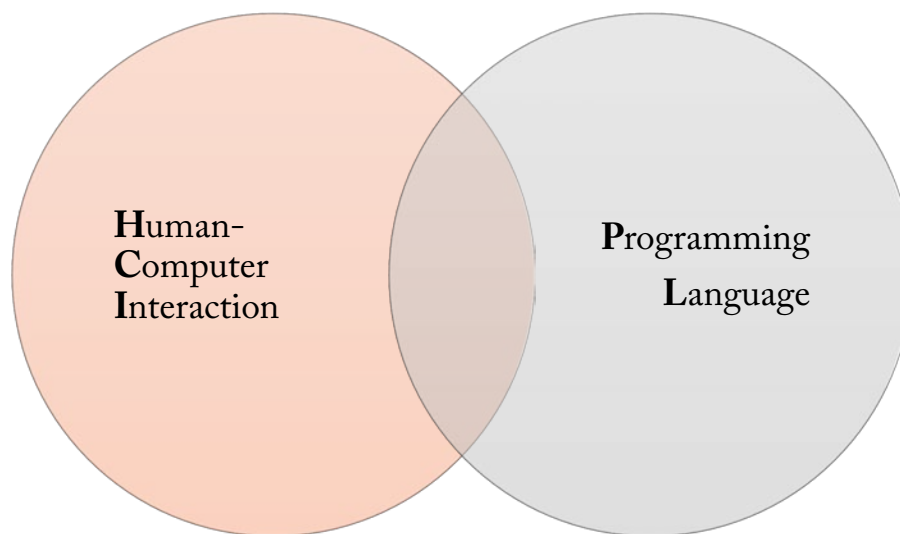


{ぼくらの→みんなの}プログラミング

研究分野:

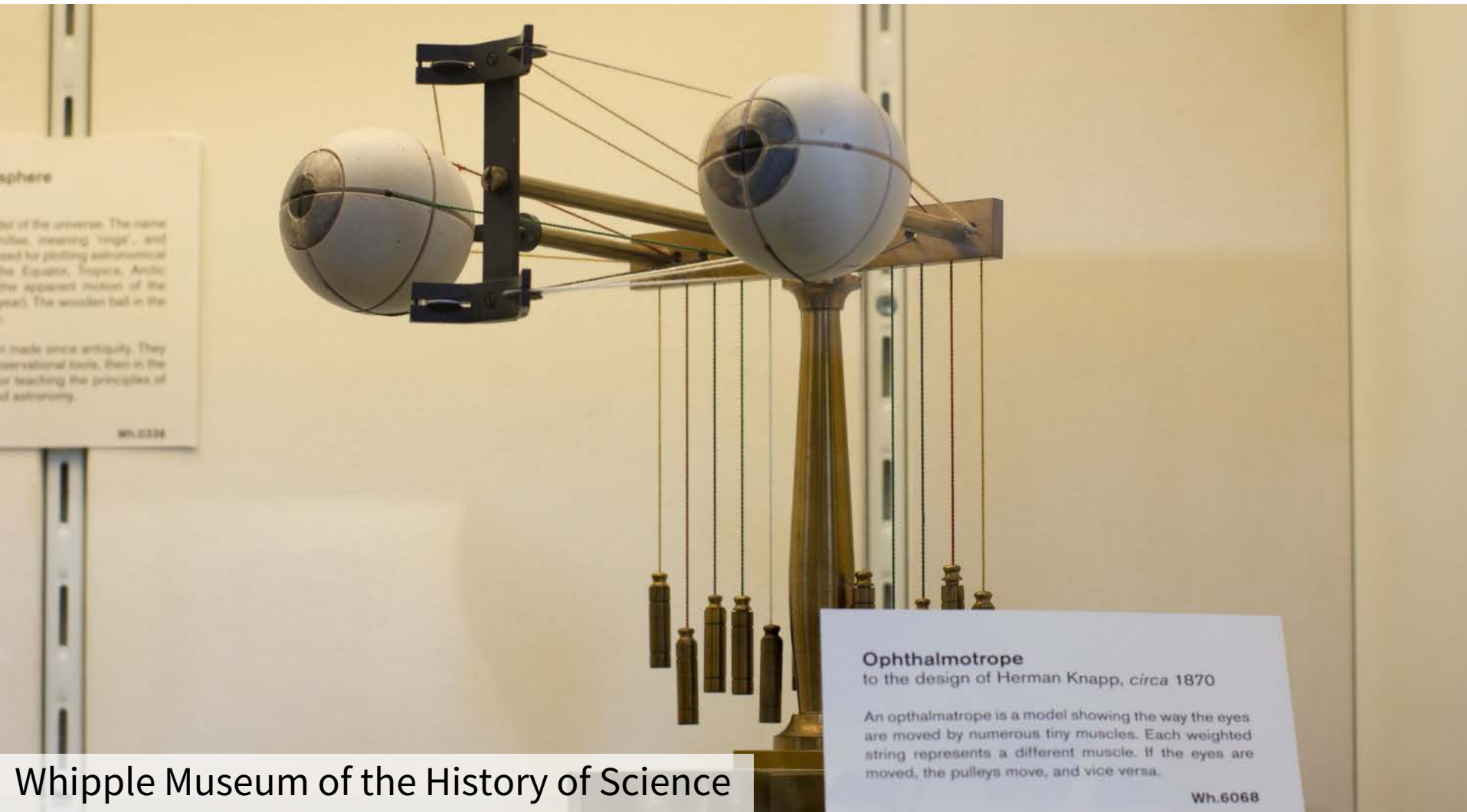
Human-Computer Interaction

- 人とコンピュータの関係改善
- とくに、プログラミング体験の改善



人々が快適に楽しく過ごすための**仕組み**作り

仕組みに萌える



Whipple Museum of the History of Science

{ぼくらの→みんなの}プログラミング

仕組みに萌える



Whipple Museum of the History of Science

{ぼくらの→みんなの}プログラミング

仕組みを疑う・考える・作る



異質な人が集まって
何かが起きる仕組み(場)をつくる

誰でも創造性を発揮できる・楽しめる
仕組み(システム)をつくる

研究分野:

Human-Computer Interaction

人々が快適に楽しく過ごすための**仕組み**作り

{ぼくらの→みんなの}プログラミング

今日の講演の目標

- プログラミングに使っている**仕組み**（開発環境）を疑ってみてほしいです！
- プログラミングに使っている**仕組み**の過去を知り、少し先の未来を知ってほしいです！！
- プログラミングの未来、社会の未来について、考えてみてほしいです!!!



よろしくお願いします！

ぼくらのプログラミング

みんなのプログラミング

{ぼくらの→みんなの}プログラミング

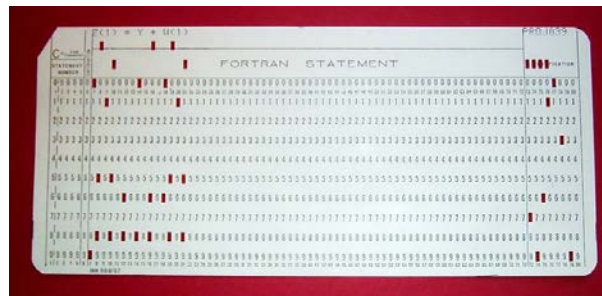
1960年代前半までの開発環境

- コンピュータは自動計算のためのもの
- 「コーディング」はカードにパンチすることを指した



記録する内容は
自力で計算

機械語をカードに記録



実行



プログラミングは全くインタラクティブじゃなかった😞

統合開発環境の登場

- コンピュータ上でインタラクティブに開発できる
- ワークフロー全体を支援してくれる

```
public class HelloWorld {  
    // Hello World!  
    public static void  
        main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

スクリーン上で
高級言語を記述

機械語を電子的に保存

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	0123456789ABCDEF
000000	CA	FE	BA	BE	00	00	00	33	00	14	07	00	02	01	00	0A	3.....
000010	48	85	6C	6C	6F	57	6F	72	6C	64	07	00	04	01	00	10	HelloWorld.....
000020	6A	61	76	61	2F	6C	61	6E	67	2F	4F	62	6A	65	63	74	java/lang/Object
000030	01	00	06	3C	69	6E	69	74	3E	01	00	03	28	29	56	01	...<init>...()V.
000040	00	04	43	6F	64	65	0A	00	03	00	09	0C	00	05	00	06	..Code.....
000050	01	00	0F	4C	69	6E	65	4E	75	6D	62	65	72	54	61	62	...LineNumberTab
000060	6C	65	01	00	12	4C	6F	63	61	6C	56	61	72	69	61	62	le...LocalVariab
000070	6C	65	54	61	62	6C	65	01	00	04	74	68	69	73	01	00	leTable...this.
000080	0C	4C	48	65	6C	6C	6F	57	6F	72	6C	64	3B	01	00	04	..LHelloWorld;..
000090	6D	61	69	6E	01	00	16	28	5B	4C	6A	61	76	61	2F	6C	main...([Ljava

実行

Console

<terminated> HelloWorld
Hello World!

エディタ

コンパイラ

デバッガ

インタラクティブにプログラムを作れるようになった☺

文字ベースの統合開発環境 (1/2)

Dartmouth Time-Sharing System [1964-]

- ソースコードの読み込み・保存、編集、コンパイル、実行が一通りできた初めての統合開発環境

```
GW-BASIC 3.23
(C) Copyright Microsoft 1983,1984,1985,1986,1987,1988
60300 Bytes free
Ok
```

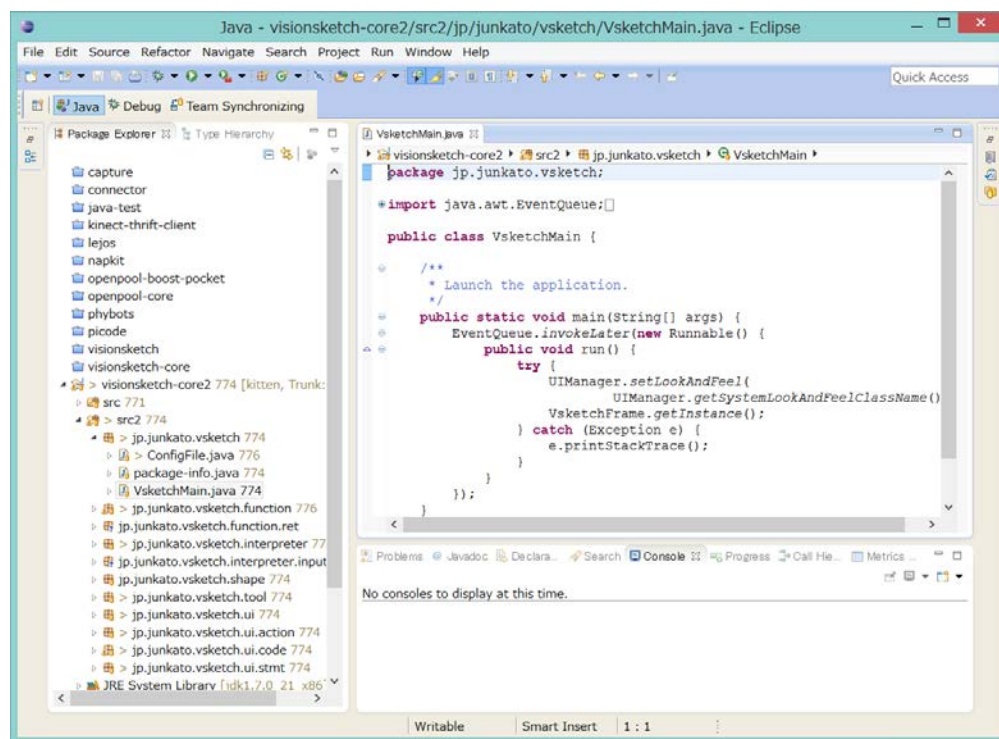
```
1LIST 2RUN← 3LOAD" 4SAVE" 5CONT← 6,"LPT1 7TRON← 8TROFF← 9KEY 0SCREEN
```

GW-BASIC

DTSSに似たMicrosoft製方言

文字ベースの統合開発環境 (2/2)

Visual Studio (Microsoft C) [1983-], Eclipse (IBM VisualAge) [1984-], Xcode [2003-], ...



- 文字列の高級言語を編集できるエディタ
- メモリ内容を文字列で表示できるデバッガ
- ...

インタフェースの大部分が
文字ベースという点では
40年前と同じ

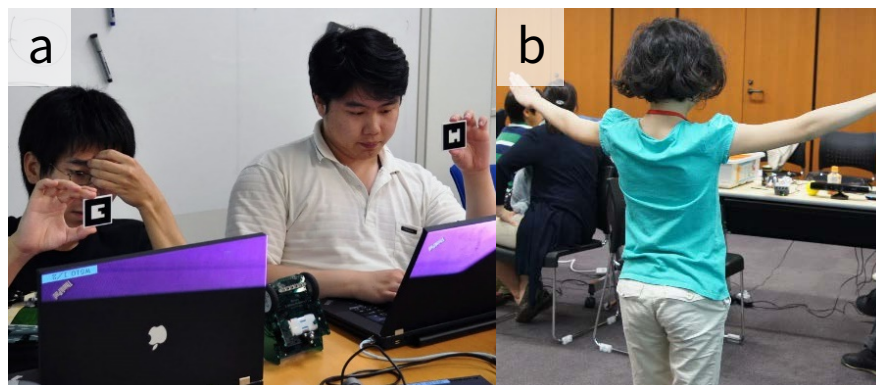
実世界データを伴うプログラムの増加

実世界入力

(センサの生データ)

a. RGBカメラ → 映像情報

b. 深度センサ → 姿勢情報



実世界出力

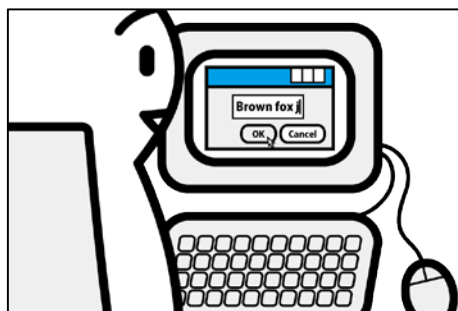
(アクチュエータの生データ)

c. 姿勢情報 → ロボット

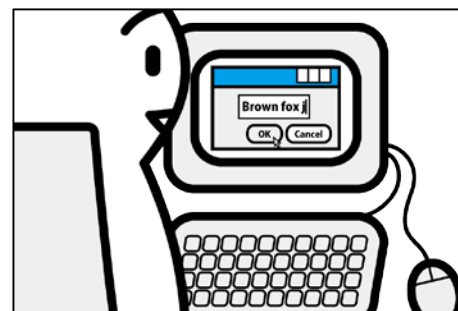


実世界データを伴うプログラム開発の特徴 (1/2)

従来型入出力:

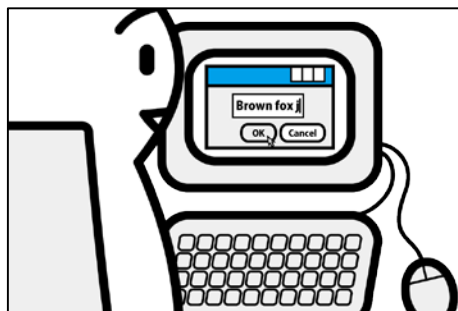


開発環境

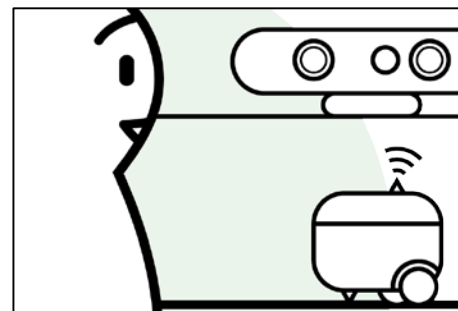


実行環境

実世界入出力:



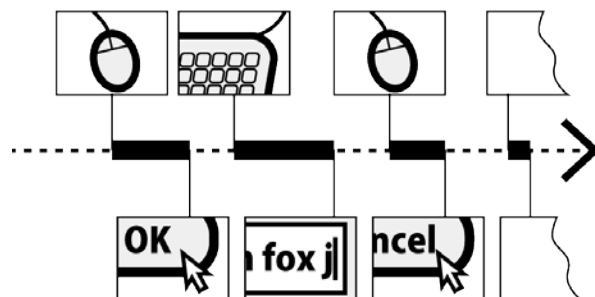
開発環境



実行環境

実世界データを伴うプログラム開発の特徴 (2/2)

従来型入出力:

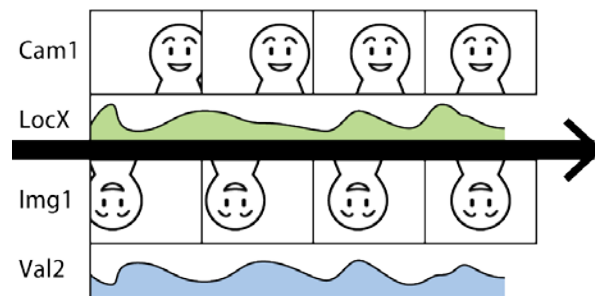


シンプルで断続的

numAbc	int, 26983
message	string, "Quick..."
isClicked	boolean, false

文字表記が容易

実世界入出力:



複雑で連続的

numAbc	int, 26983
message	string, "Quick..."
isClicked	boolean, false

文字表記が困難

Picode

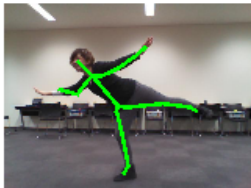
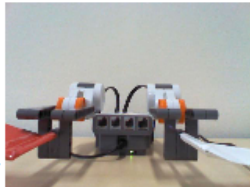
[[Kato, ACM CHI '13](#)]

BallShooter | Picode

Sketch File View

Run Stop

BallShooter

```
if (pose.eq()) {  
  showText("Got the command!");  
  
  ;  
  nxt.setPose(  
  }  
}
```

71 lines

Human Robot

My NXT (MindstormsNXT) Edit

both

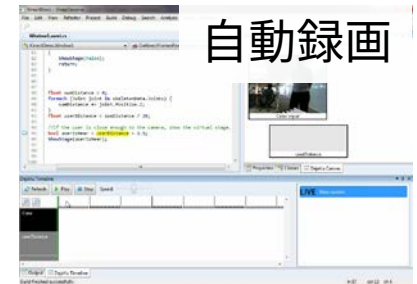
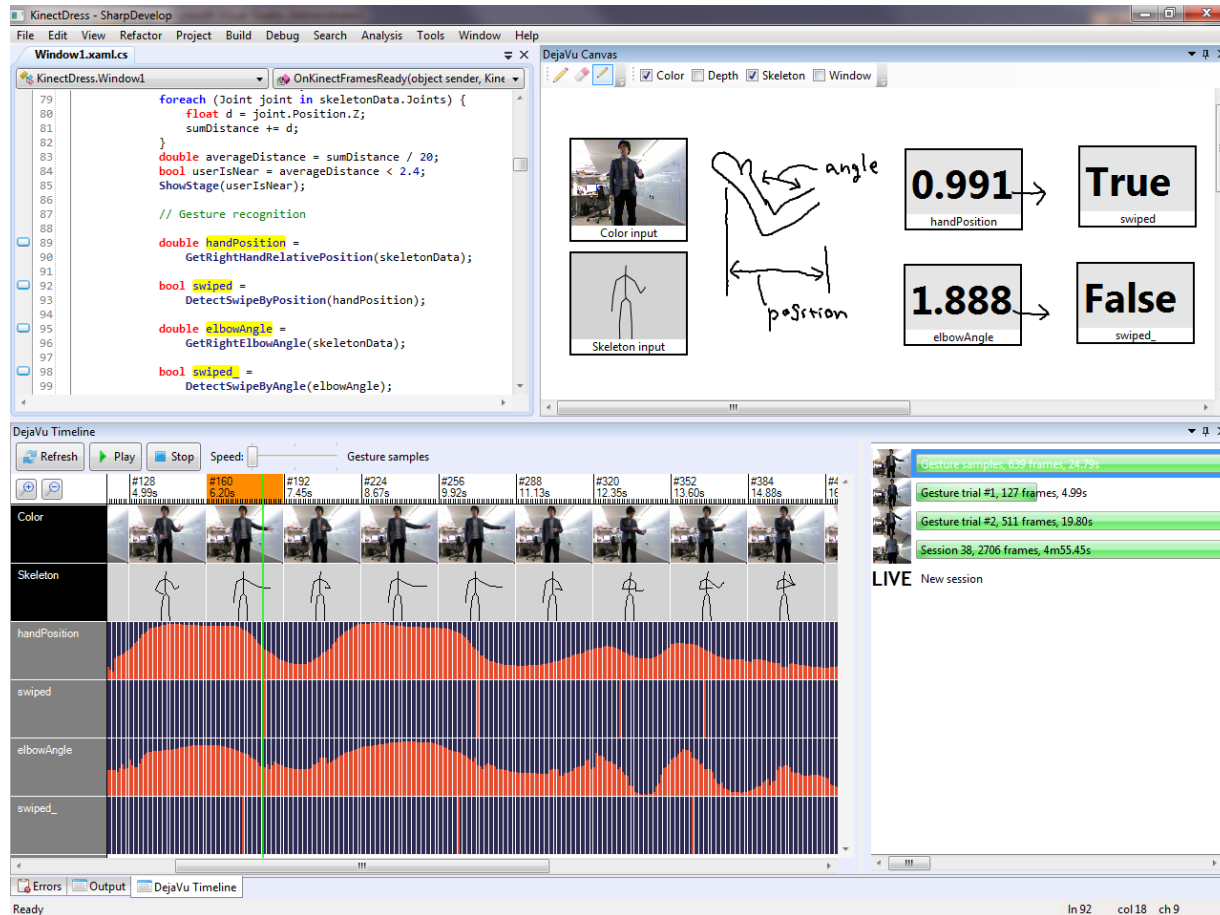
left

Capture Delete

Undo Redo

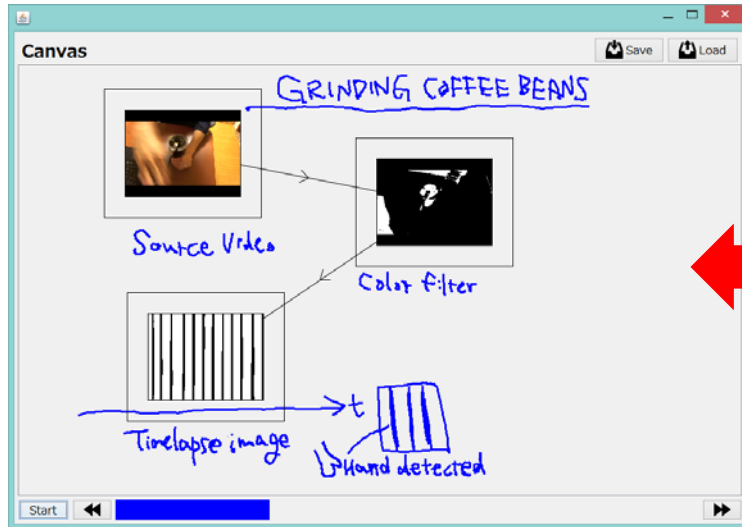
DejaVu

[Kato, ACM UIST '12]

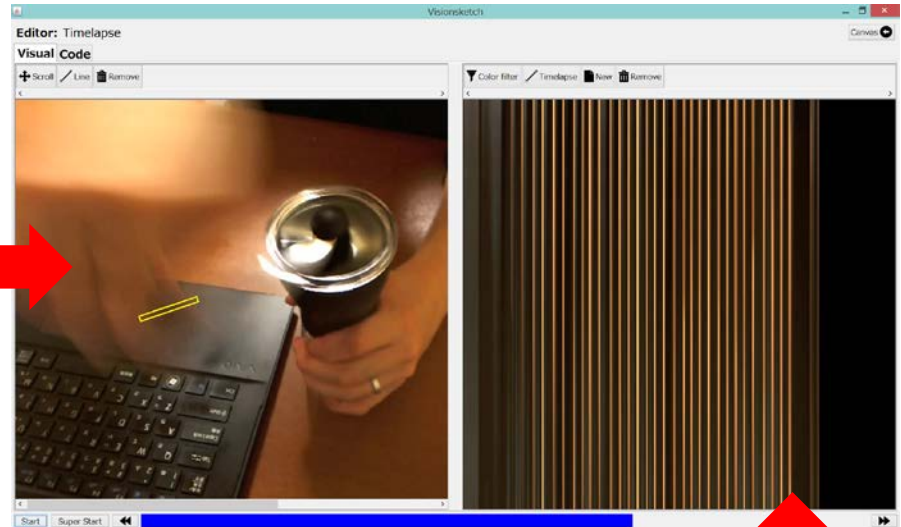


VisionSketch

[Kato, GI '14]

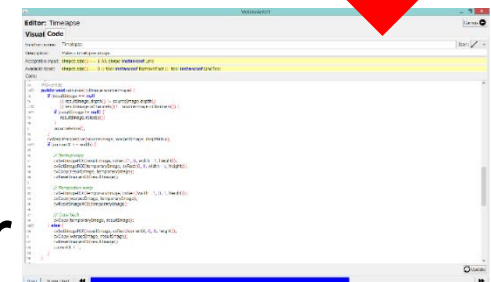


Canvas



Visual Editor

Code Editor



これからのプログラムとデータ

さまざまな種類のデータがあふれる時代、
ロジックだけを文字で開発する環境では**不十分**



- プログラムとデータの主従関係逆転
- 開発(プログラム)と制作(データ)の境界消失

これからの統合開発環境

- プログラムとデータの主従関係逆転
- 開発(プログラム)と制作(データ)の境界消失



- 各データの扱いに特化した開発環境が必要
- 制作・編集支援ツールの知見を融合するべき
- (視覚以外も)五感をフル活用するインタフェース



メディアコンテンツのための
統合開発デザイン環境がウェブ上に現れる

作りました

みんなのプログラミング

ぼくらのプログラミング

[\[Kato, ACM CHI '15 & ICLC '15\]](#)

{ぼくらの→みんなの}プログラミング

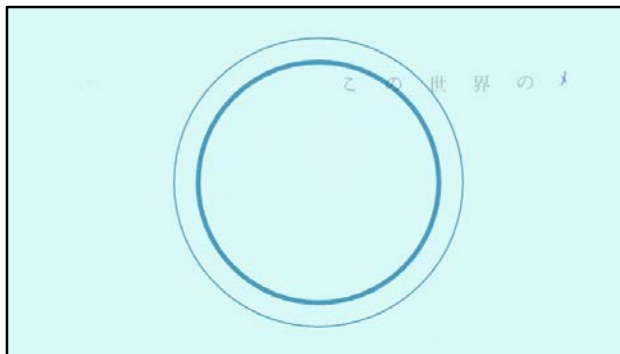
歌詞アニメーション



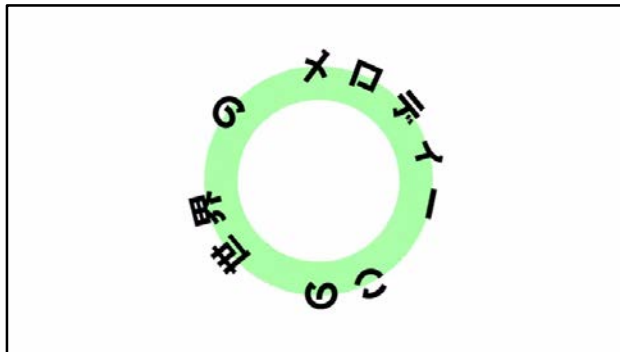
この世界の

この世界の **X**

この世界の **X**



この世界の **X**ロディー



more at <http://textalive.jp>

歌詞アニメーションの魅力

カラオケ表示

失敗を繰り返し越えて行くんだよ

- どの楽曲でも同じ見た目
- 発声に合わせて色が変わるだけ

歌詞アニメーション



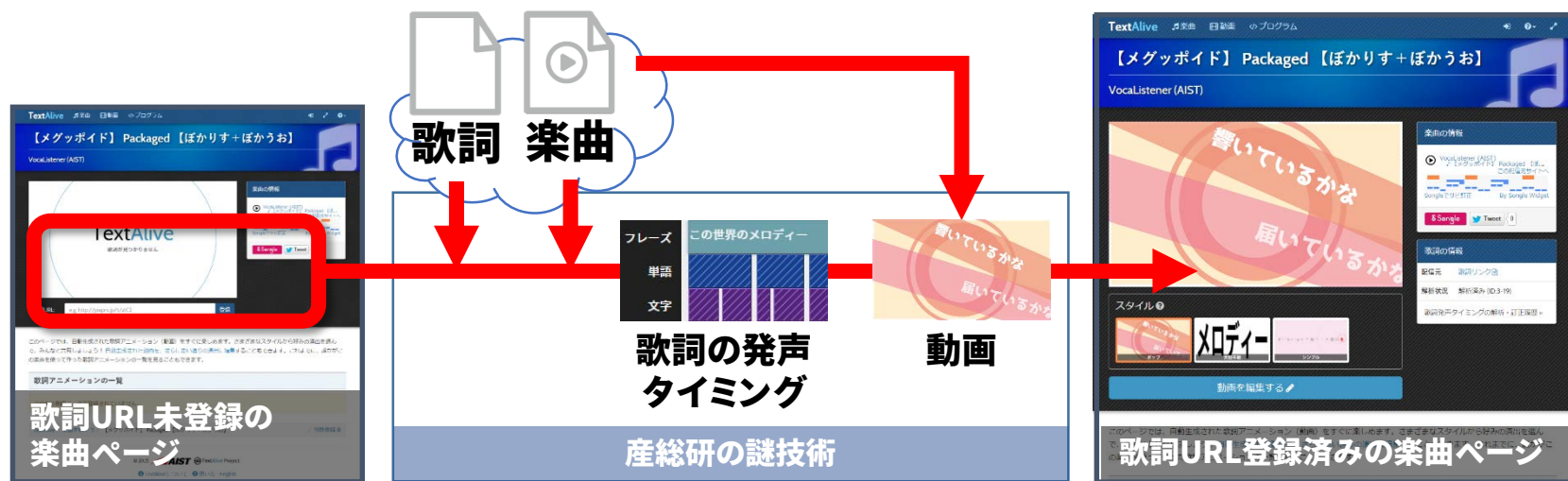
- **楽曲ごと**に歌詞を魅力的に演出
- 動画共有サービスで増加中



歌詞アニメーションは音楽の楽しみを広げてくれる

歌詞アニメーションの自動生成

- 産総研の謎技術でウェブ上の歌詞・楽曲から歌詞アニメーションを自動生成
- スタイルを選んで演出を瞬時に切り替えられる



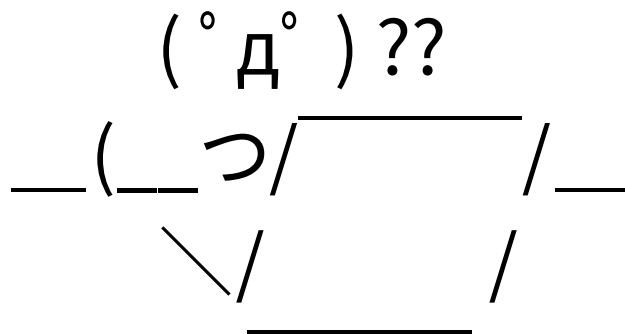
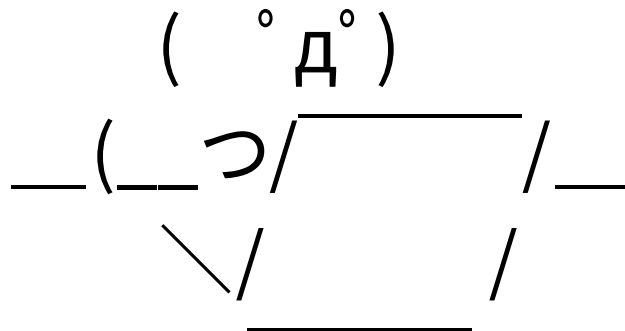
{ぼくらの→みんなの}プログラミング

歌詞アニメーションの編集



{ぼくらの→みんなの}プログラミング

みんなの「プログラミング」?



動画vは時間の関数f(t)

for each (全描画対象オブジェクト)

描画のためのパラメタを操作

for each (グラフィック)

座標系操作と描画

for each (フレーズ)

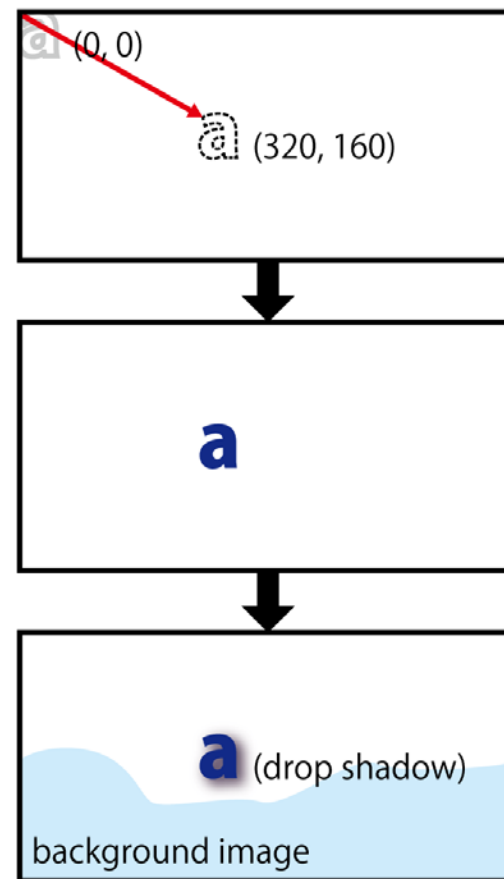
座標系操作

for each (単語)

座標系操作

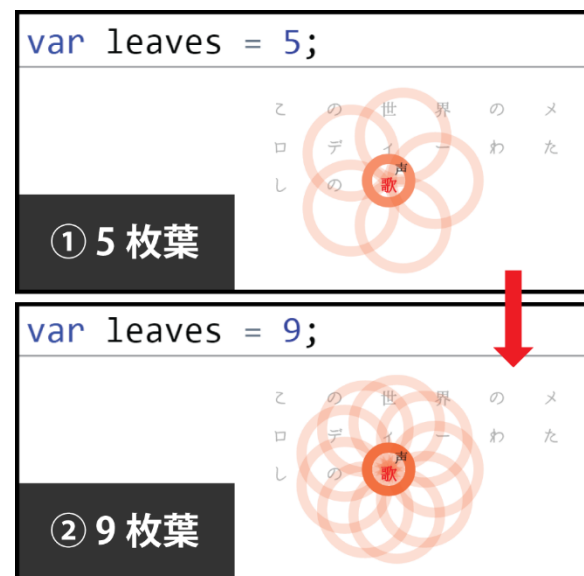
for each (文字)

描画



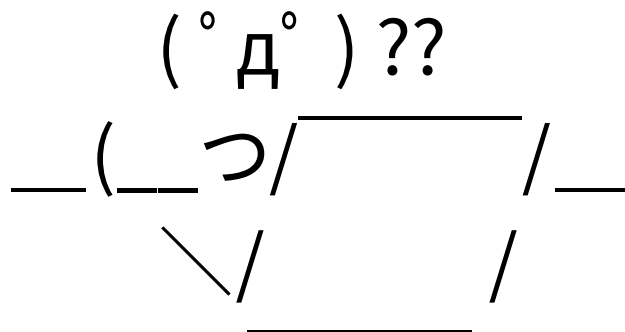
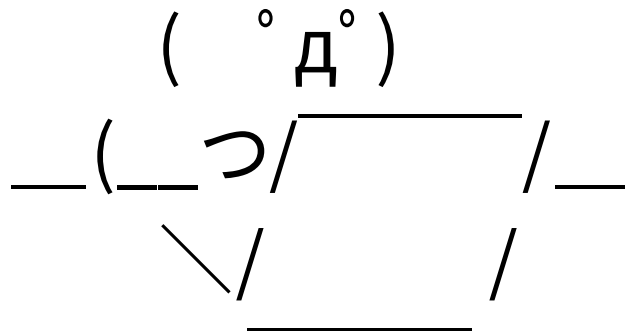
演出のライブプログラミング

- 文字の動きなどの演出機能をプログラミングできる
- プログラミングした内容がすぐに動画に反映される



TextAliveは、ユーザがプログラミングで
機能を強化できる

「みんなの」プログラミング?



みんな幸せなプログラミング

- 成果をみんなが使える
- みんなが好きに調整できる

```
var leaves = 5;
```

この世界のメ
ロディーわ
た

① 5 枚葉

```
var leaves = 9;
```

この世界のメ
ロディーわ
た

② 9 枚葉

```
// @ui Slider(1, 20)  
// @title 花びらの数  
this.leaves = 5;
```

テンプレートの調整 (花びらの)

花びらの数

この世界のメ
ロディーわ
た

③ n 枚葉 (13 枚に調整)

{ぼくらの→みんなの}プログラミング

プログラマ的方法の一般化

例えば…

バージョン管理

TextAlive 楽曲 動画 プログラム 編集 ログイン フルスクリーン ヘルプ

歌詞アニメーション（動画）の情報

更新履歴 貢献者 埋め込み

この歌詞アニメーションの作成に貢献した人の一覧です。

動画制作

arc@dmz

Anonymous

テンプレート開発

arc@dmz

isyda

ID: 6

Yutaka Obuchi

ID: 15

ともき

遊牧家族/yuuboku@UTABO7

Anonymous

このページでは、歌詞が楽曲に合わせてタイミングよく動く歌詞アニメーション（動画）を見ることができます。TextAliveでは、この歌詞アニメーションを編集して派生コンテンツを作ったり、同じ楽曲の歌詞アニメーションを視聴して楽しんだりできます。他の楽曲を検索して新しく歌詞アニメーションを作ることできます。

TextAlive / 動画を見よう / 妄想スケッチ (ラストまで) / 削除依頼

© 2015 AIST TextAlive Project

貢献度の可視化

TextAlive 楽曲 動画 プログラム 編集 ログイン フルスクリーン ヘルプ

歌詞アニメーション（動画）の情報

更新履歴 貢献者 埋め込み

この歌詞アニメーションの作成に貢献した人の一覧です。

動画制作

arc@dmz

Anonymous

テンプレート開発

arc@dmz

isyda

ID: 6

Yutaka Obuchi

ID: 15

ともき

遊牧家族/yuuboku@UTABO7

Anonymous

このページでは、歌詞が楽曲に合わせてタイミングよく動く歌詞アニメーション（動画）を見ることができます。TextAliveでは、この歌詞アニメーションを編集して派生コンテンツを作ったり、同じ楽曲の歌詞アニメーションを視聴して楽しんだりできます。他の楽曲を検索して新しく歌詞アニメーションを作ることできます。

TextAlive / 動画を見よう / 【メグツポイド】Packaged 【ばかりす+ほかうお】 (使っていないテンプレートを追加) / 削除依頼

© 2015 AIST TextAlive Project

{ぼくらの→みんなの}プログラミング

全部が関数=解像度無限大



宣伝



TextAlive

テキストアライブ

<http://textalive.jp>

ぜひ遊んでみてください！

{ぼくらの→みんなの}プログラミング

さまざまな種類のデータがあふれる時代、
ロジックだけを文字で開発する環境では**不十分**



- プログラムとデータの主従関係逆転
- 開発(プログラム)と制作(データ)の境界消失



- 各データの扱いに特化した開発環境が必要
- 制作・編集支援ツールの知見を融合するべき
- (視覚以外も)五感をフル活用するインタフェース



メディアコンテンツのための
統合開発デザイン環境がウェブ上に現れる

メディアコンテンツのための 統合開発デザイン環境がウェブ上に現れる

- **プログラミングできる**

- プログラミングでツールそのものの機能を拡張できる
- 機能拡張の結果をみんなで共有・活用できる

- **プログラマ的方法で運用される**

例) バージョン管理、貢献が可視化される、再利用性

- **プログラミングの利点が活かされる**

例) 解像度無限大、転送サイズがコンパクト、etc.

歌詞アニメーション
以外にも
応用可!

設計思想

プログラミング = 

プログラミング=

- 昔: プログラミングは{どうして|どこが|なぜ}  ?
- 今: 確かに  っぽい を始点に議論できる
- みんなで  プログラミングしましょう

ぼくらの(苦しい)プログラミング

Programmers are people, to 

みんなの(楽しい)プログラミング

People are programmers 

今日の講演の目標

- プログラミングに使っている**仕組み**（開発環境）を疑ってみてほしいです！
- プログラミングに使っている**仕組み**の過去を知り、少し先の未来を知ってほしいです！！
- プログラミングの未来、社会の未来について、考えてみてほしいです!!!



ありがとうございました！