

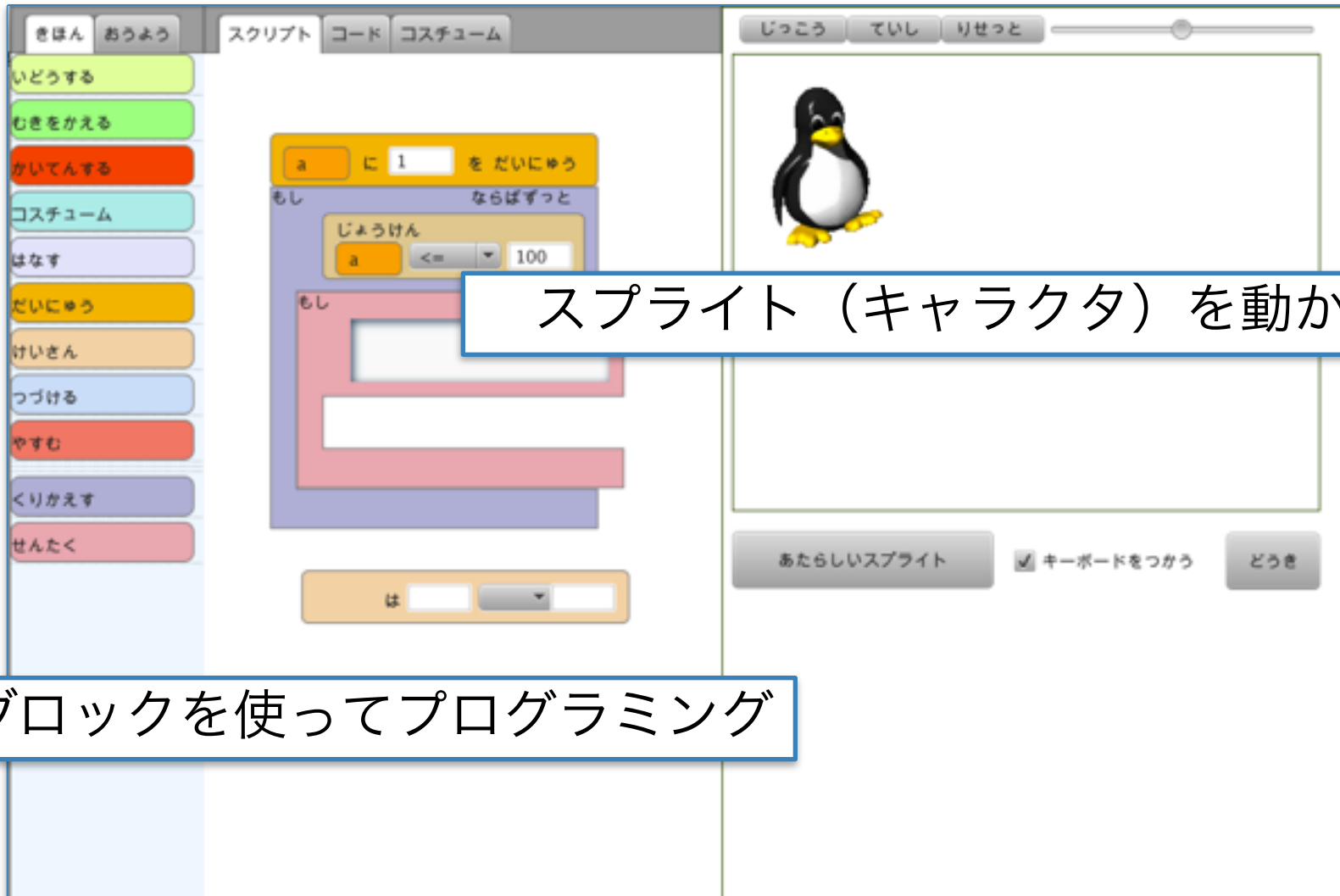
子ども向け プログラミング学習ツール 「Jointry」の開発

秋口PT
金本真左也、角征典

Jointry（ジョイントリ）の概要

- 「Jointry」は、小学校高学年向けのプログラミング学習ツールであり、キーボードを使わずにブロックを組み立ててプログラミングを可能にするソフトウェアである。
- 既存のプログラミング学習ツール（Scratch等）とは違い、友達や先生と一緒に学習できる通信プログラミング（共同プログラミング）機能が備わっている。

Jointryの画面イメージ



スプライト（キャラクタ）を動かす

ブロックを使ってプログラミング

実際の動作



開発の背景

Let's teach kids to code

子供達にプログラミングを教えよう

http://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code.html



“若い人たちが新しい技術を
上手に使って「書ける」ようになることを
どう助けられるかに 私は興味を持っています”

プログラミング教育の重要性の高まり

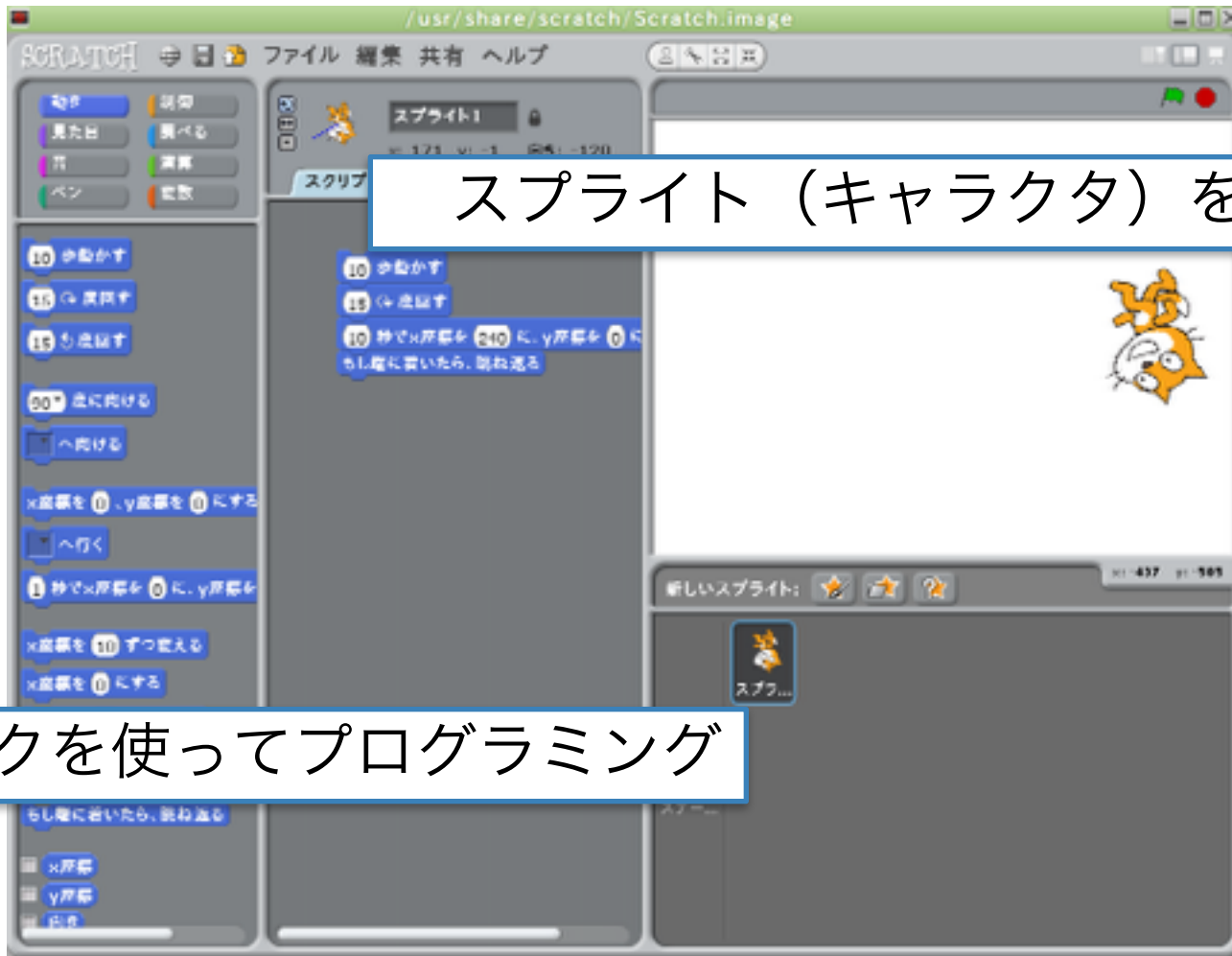
- オンライン学習 (MOOCs)
 - Code.org, Codecademy, Coursera, EdX, Khan Academy, Udacity
- イベント
 - CodingDojo, Tech Kids CAMP
 - Girls Who Code, Black Girls Code, Rails Girls
- ニュース
 - エストニア、小学1年生からプログラミングの授業実施へ
 - イギリスは義務教育 (5-16歳) 公式カリキュラムにプログラミング教育を導入
 - ドラッグ&ドロップで簡単プログラミング 文科省が子ども向けに公開

プログラミング教育の歴史

- AMBIT ('64)
- LOGO ('67)
シーモア・パパート
- 「すべての年齢の子どもたちのための
パーソナルコンピューター」 ('72)
アラン・ケイ
- HyperCard ('87)
ビル・アトキンソン
- Scratch ('03)
ミッチェル・レズニック



Scratchの画面イメージ



ブロックを使ってプログラミング

わくわくプログラミング (2013-07-25)

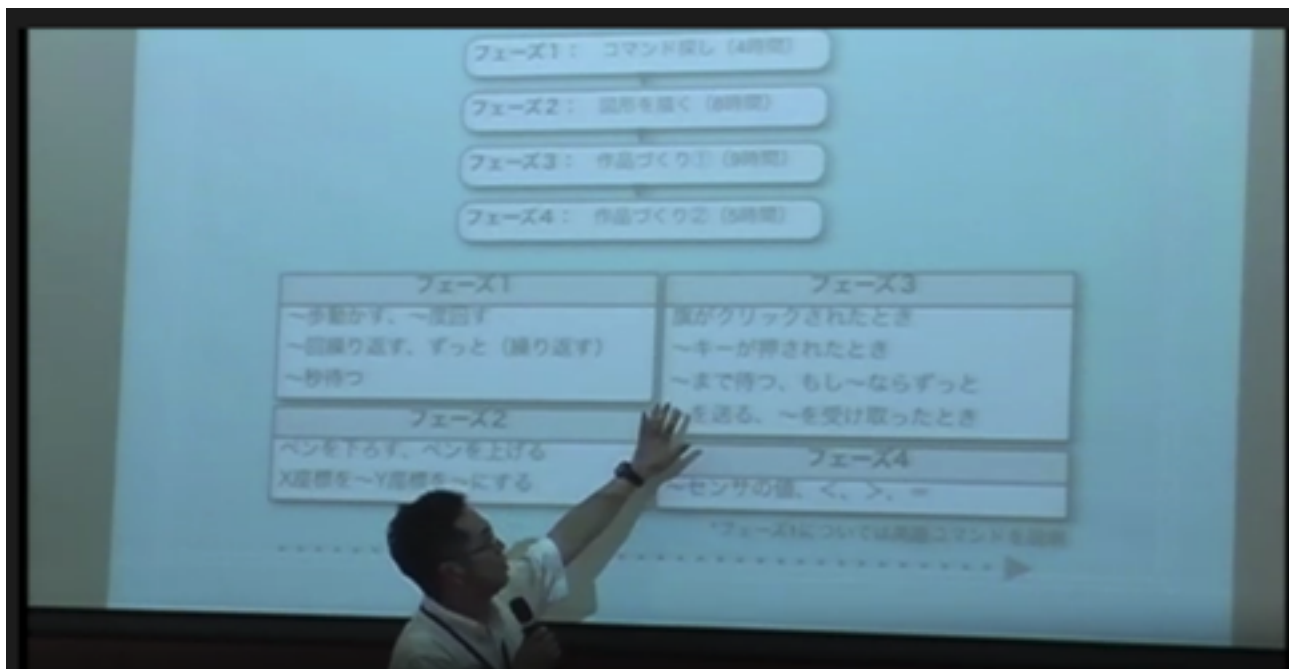


Amazon.co.jp 「コンピュータ・IT > プログラミング」 カテゴリ1位

現状の課題と考察

Scratchを用いた小学校授業の実践

森 秀樹, 杉澤 学, 張 海, and 前迫 孝憲, “Scratchを用いた小学校プログラミング授業の実践：小学生を対象としたプログラミング教育の再考(教育実践研究論文),” 日本教育工学会論文誌, vol. 34, no. 4, pp. 387–394, 2011.



Scratch Day 2011 in Tokyo
大阪大学大学院人間科学研究科 森秀樹 「Scratchを用いた小学校授業の実践」

<http://www.ustream.tv/recorded/14850693>

Scratchを用いた小学校授業の実践

- 小学4年生38名を対象に計26時間の授業
- ブロックの理解度はいずれも高い結果
 - 「特にスプライト制御と繰り返しについては、殆どの生徒が5段階中の5と最高評価」
- 楽しかったか？ → 平均 4.78 (SD : 0.76)
- 簡単だったか？ → 平均 3.22 (SD : 1.27)
- "初めの学習では苦戦 (中略) 友だちや先生から教えてもらって (中略) 楽しむことができました"

森 秀樹, 杉澤 学, 張 海, and 前迫 孝憲, "Scratchを用いた小学校プログラミング授業の実践 : 小学生を対象としたプログラミング教育の再考(教育実践研究論文)," 日本教育工学会論文誌, vol. 34, no. 4, pp. 387–394, 2011.

現状の課題と考察

●現状

Scratch 等のビジュアルプログラミングの人気の高まっている



●課題

ただし、ひとりでは習得が難しいという研究報告もある



●考察

複数人での操作を可能にすることで、習得を容易にできないか



「友だちや先生から教えてもらって
(中略) 楽しむことができました」

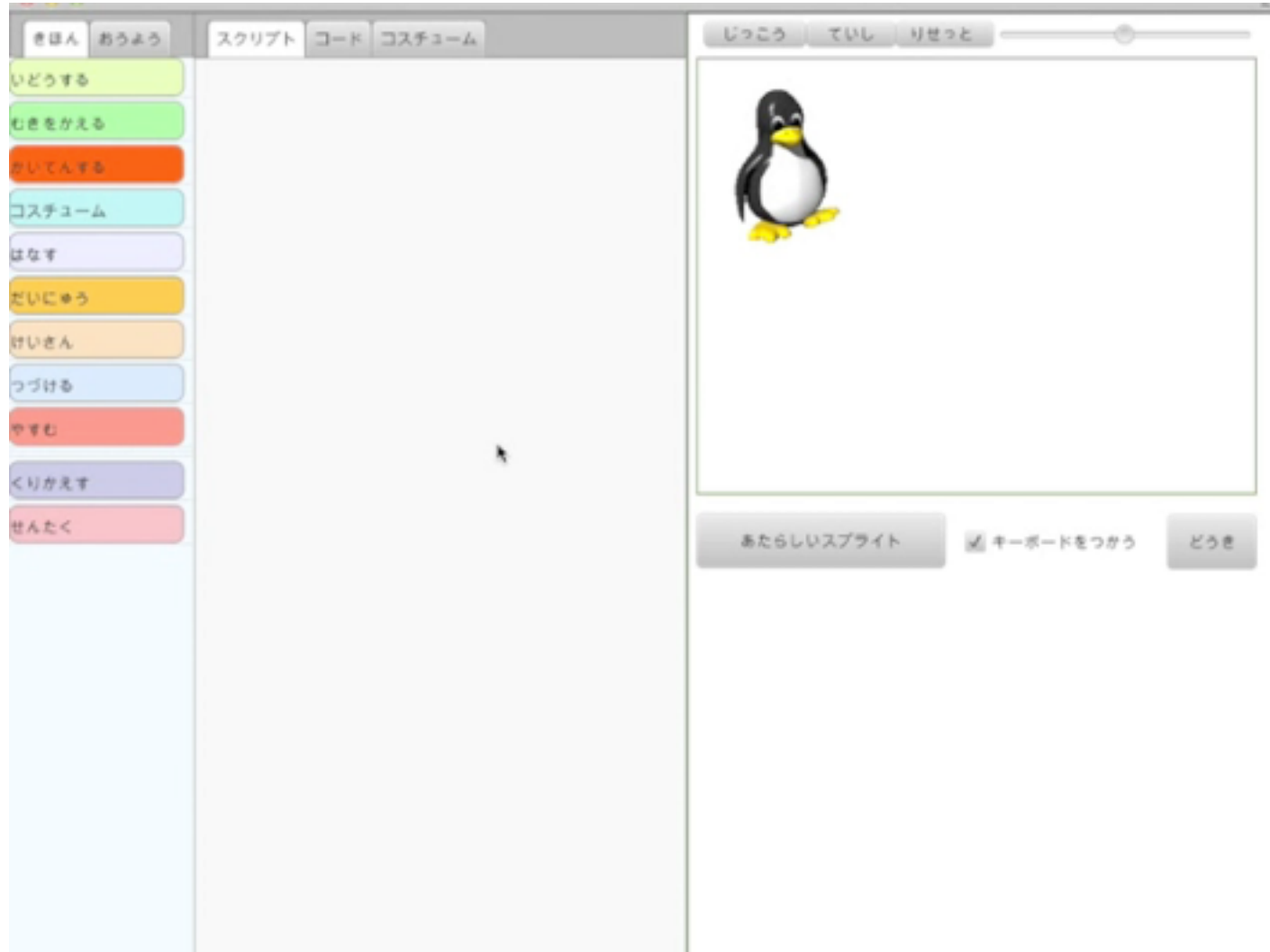
ソリューション

Jointryの3つの特徴

1. 「理解しやすい」「楽しい」ブロックプログラミングはそのまま踏襲
2. ともだちと一緒にできる通信プログラミング（共同プログラミング）の機能を搭載
3. （さらなる学習のために）ブロックで作成したプログラムを文字で確認できる機能
[松澤他、2014]

※ ただし、自分たちの学習のために新規でプロダクトを開発する

1. ブロックプログラミング



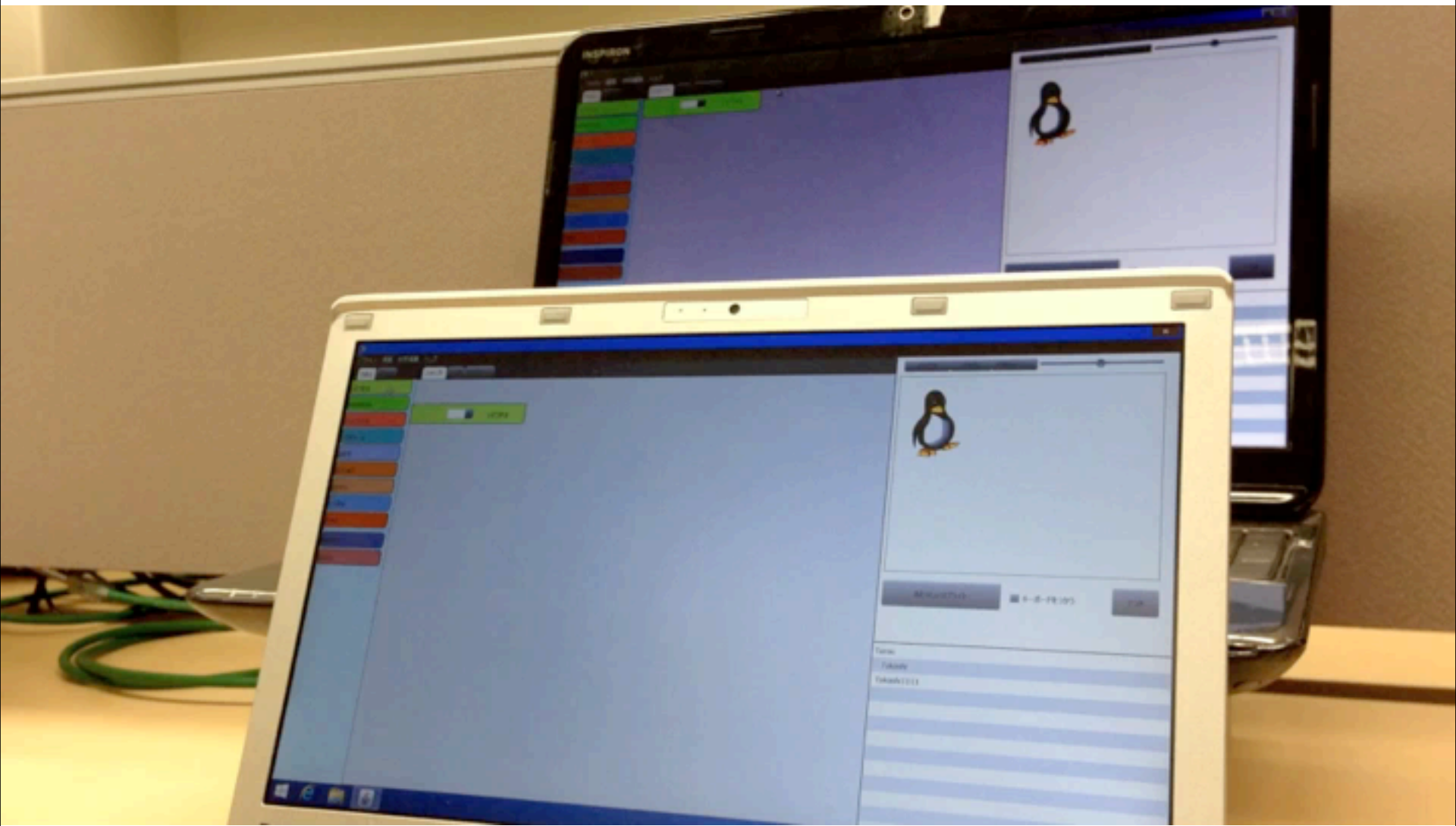
1. ブロックプログラミング (キーボード)



1. ブロックプログラミング (FizzBuzz)



2. 通信プログラミング




3. 文字で確認できるプログラム

きほん おうよう スクリプト コード コスチューム

いどうする
むきをかえる
かいてんする
コスチューム
はなす
だいにゆう
けいさん
つづける
やすむ
くりかえす
せんたく

```
i = 1
while i <= 100 {
  speech i
  fizz = (i % 3)
  if fizz == 0 {
    speech "Fizz"
  }
  buzz = (i % 5)
  if buzz == 0 {
    speech "Buzz"
  }
  i = (i + 1)
  sleep 5
}
```

じっこう ていし りせっと



あたらしいスプライト ☒

プロダクトの評価

プロダクトの評価（の前に）

大きな問題

周囲に小学生がいない……。

「もしも小学生が使うなら？」
という前提で大人に感想をいただく

プロダクトの評価方法

- 期間：2014年1月～2月
- 人数：7人（AIIT関係者）
- 調査方法：
 - 偶然出会った人に使ってもらう
 - インタビュー形式でヒアリング
 - 「楽しさ」と「難しさ」で評価

利用者の声

- インターフェイスが洗練されていない
 - ブロックの操作がわかりにくい
 - ブロックが多すぎて画面が複雑
 - 無機質でゲーム性が足りない
- 通信プログラミングは便利そう
 - 誰かと一緒にできるのは楽しい
 - 今後は「共創」が重要になるはず

利用者の声を受けて

- ブロックの操作性の改善
 - Scratchと同じところを変えない
 - ブロック作成時の登場位置を変更
 - 接続時の挙動を調整
- 通信プログラミングの改善
 - 通信設定の簡略化
 - GUIで起動／停止を可能に

技術詳細

1. ブロックプログラミング

- GUIはJavaで作成
 - Java 7
 - JavaFX 2.2
 - NetBeans 7.4
 - JavaFX Scene Builder



NetBeans

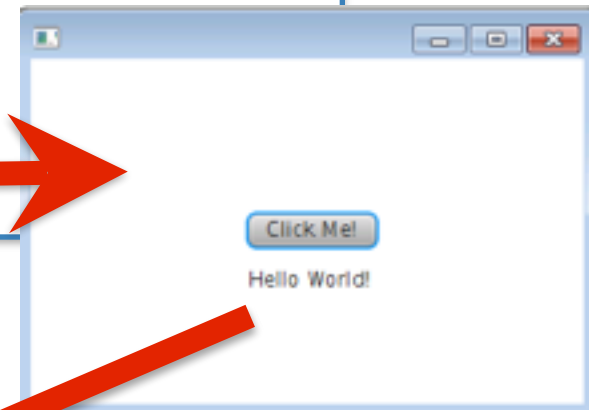
1. ブロックプログラミング

```
<!-- view.fxml -->
```

```
<Pane fx:controller="MyController" ... >
```

```
  <Label fx:id="message" />
```

```
:
```



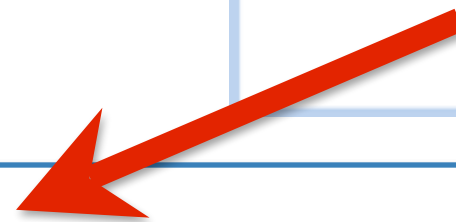
```
// MyController.java
```

```
public Class MyController implements Initializer {
```

```
  @FXML
```

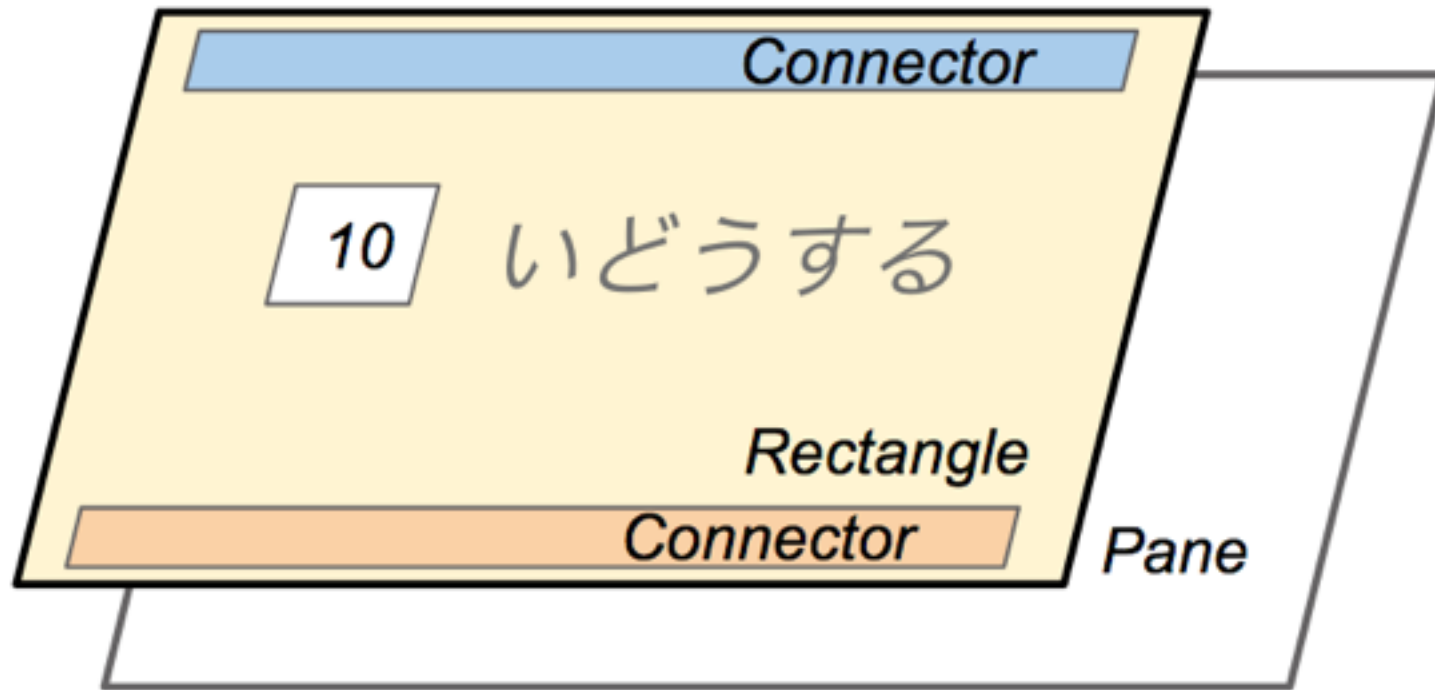
```
  private Label message;
```

```
:
```



1. ブロックプログラミング

Connector 同士の当たり判定で接続処理



2. 通信プログラミング

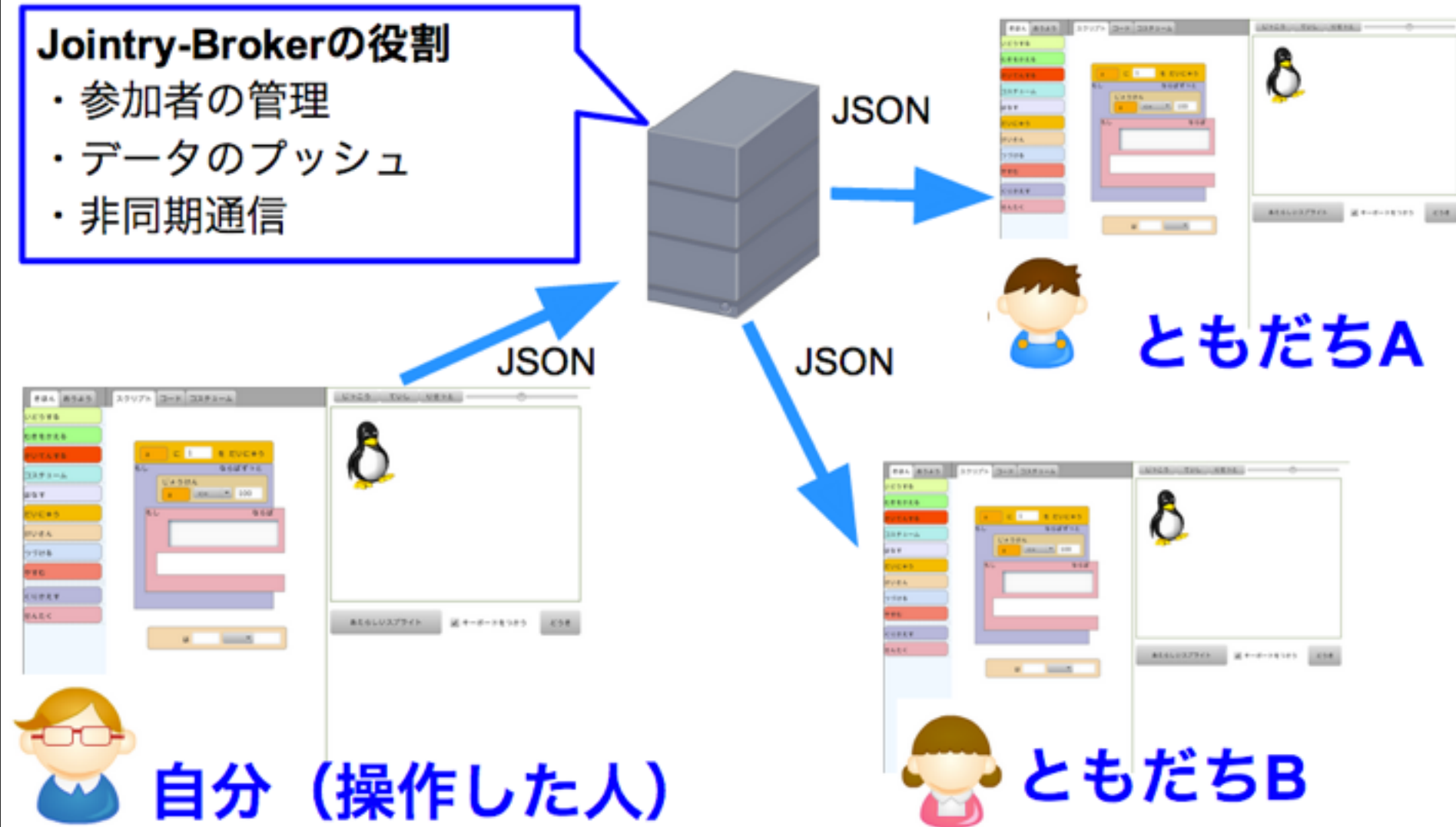
- 通信プロトコル
 - HTTP／JSON
 - JSON：JavaScriptのオブジェクト表記をベースにしたデータ構造
- 双方向通信ライブラリ
 - Jointry-Broker（独自ライブラリ）

<https://github.com/jointry/jointry-broker>

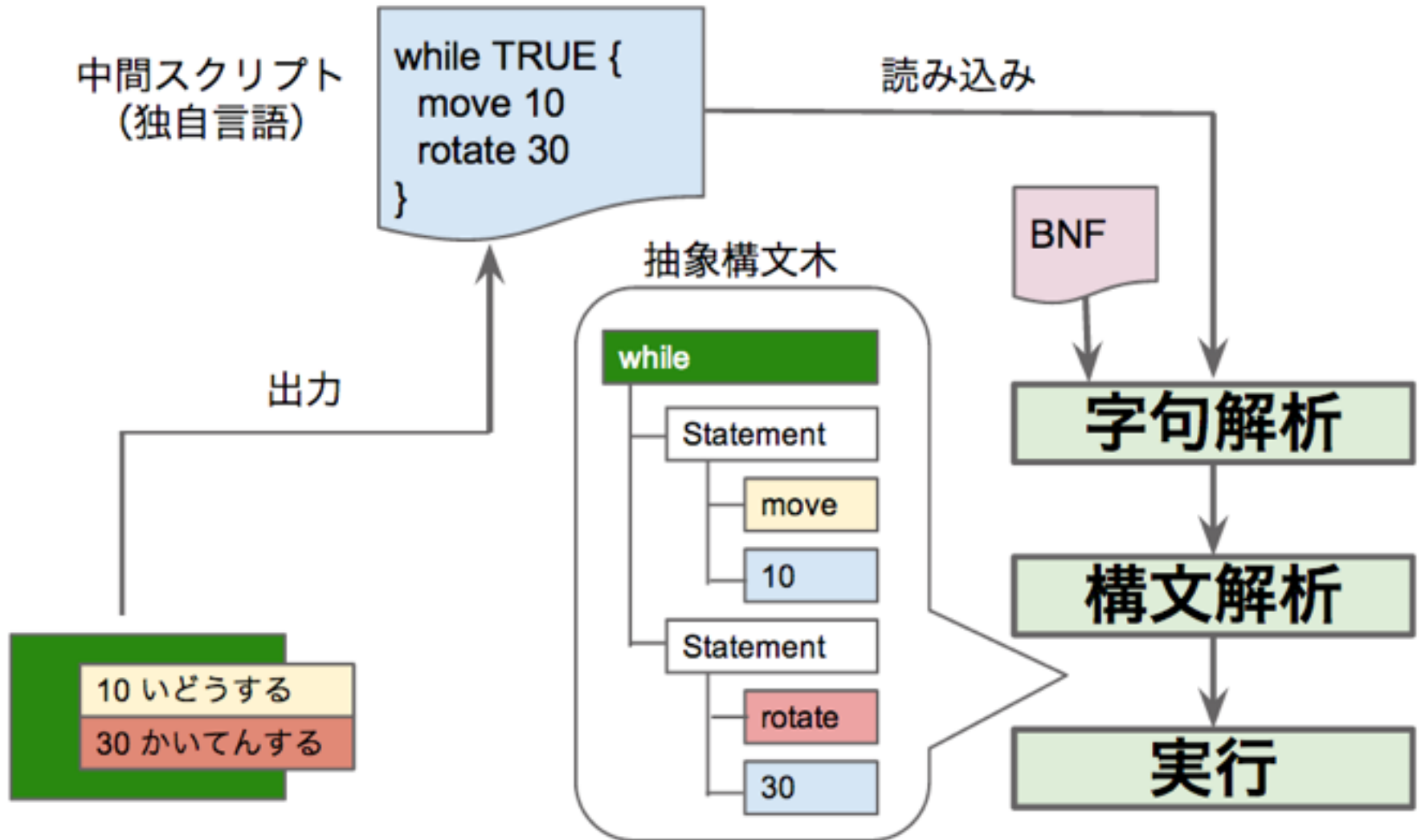
2. 通信プログラミング

Jointry-Brokerの役割

- ・参加者の管理
- ・データのプッシュ
- ・非同期通信



3. 文字で確認できるプログラム



3. 文字で確認できるプログラム

```
primary      : ( "[" [ elements ] "]" | "(" expr ")"  
              | NUMBER | IDENTIFIER | STRING ) { postfix }  
factor       : "-" primary | primary  
expr         : factor { OP factor }  
block        : "{" [ statement ] { ( ";" | EOF ) [ statement ] } "  
simple        : expr [ args ]  
statement    : "if" expr block [ "else" block ]  
              | "while" expr block  
              | "rotate" expr  
              | "move" expr  
              | "costume" expr  
              | simple  
program      : [ defclass | def | statemnet ] ( ";" | EOL )
```

独自のステートメントを定義していくことで、
ビジュアルプログラミングらしい構文が作成できる (DSL)

プロジェクトの進め方

プロジェクトの目的

1. 自ら課題を設定し、ソフトウェアの企画力を向上させる
2. 高度なプログラミング技術を習得する
3. 不確実性の高いプロダクトの開発手法を習得する

スケジュール



- ・ 1Q

- ・ 企画、技術調査
- ・ アーキテクチャ設計

- ・ 2Q

- ・ ブロック機能
- ・ コード生成機能

- ・ 3Q

- ・ 通信機能
- ・ ファイル保存機能

- ・ 4Q

- ・ コードの洗練
- ・ ドキュメント作成
- ・ 利用者インタビュー

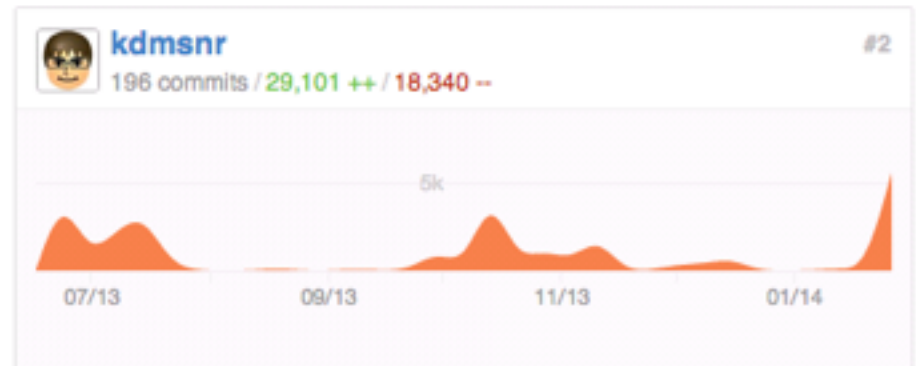
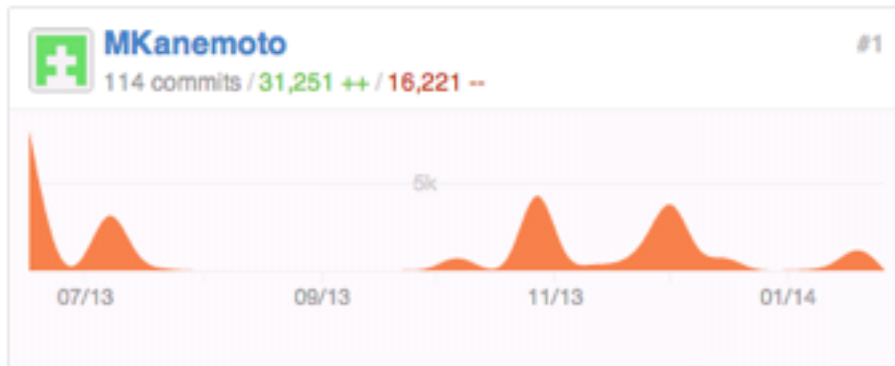
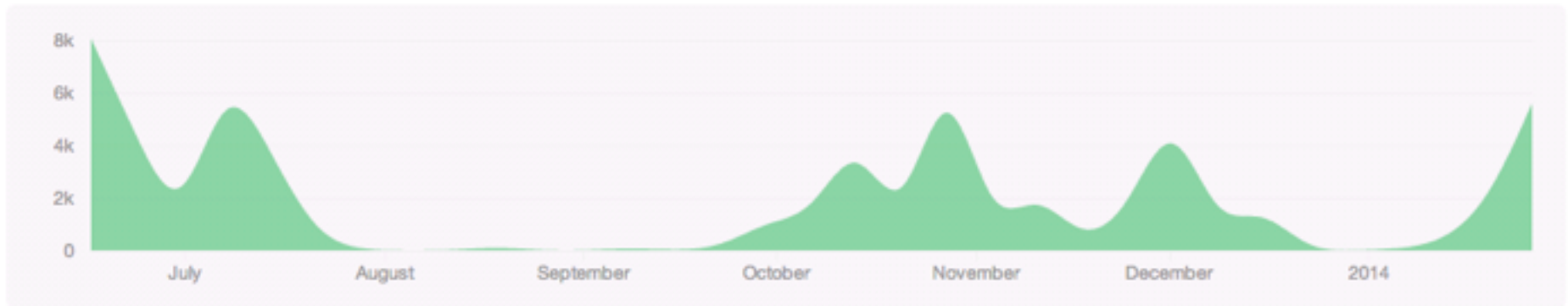
開発プロセス

- 年間と四半期の大まかな計画を立案
 - 必要と思われるタスクを列挙
 - 手のあいた人がタスクにサインアップ
- 週2回のコアミーティング（水曜日と土曜日）
- 土曜日に進捗報告（3つの質問）と改善
 - 1. 今週やったこと（+ 動作デモ）
 - 2. 来週やること
 - 3. 問題点と今後の見通し
- 利用者インタビュー

開発データ

コード行数：9,860行（約7.5K LOC）

クラス数：105クラス



<https://github.com/jointry/jointry>

プロジェクトの評価

- よかったこと

- 前例のないプロダクトを企画できた
- 評価可能なレベルのプロダクトが完成した
- 新しい技術に取り組むことができた

- 改善できるところ

- 機能単位で開発したために、アーキテクチャ設計がうまくできていない
- 本来の対象者（小学生）に利用してもらっていない

プロジェクトの改善案

1. 事前にプロトタイプを複数作成し、技術的な選択肢を吟味する時間を設ける。
2. 利用者の声を早い段階から取り入れて、プロダクトを改善するプロセスを実施する。
3. 特に、デザインやユーザビリティの向上については、ユーザー評価前にスケジュールを確保する。

参考文献

1. 森 秀樹, 杉澤 学, 張 海, and 前迫 孝憲, “Scratchを用いた小学校プログラミング授業の実践：小学生を対象としたプログラミング教育の再考(教育実践研究論文),” 日本教育工学会論文誌, vol. 34, no. 4, pp. 387–394, 2011.
2. 松澤 芳昭, 保井 元, 杉浦 学, and 酒井 三四郎, “ビジュアル-Java相互変換によるシームレスな言語移行を指向したプログラミング学習環境の提案と評価,” 情報処理学会論文誌, vol. 55, no. 1, pp. 57–71, 2014.
3. 伊藤 一成, 阿部 和広, and 新目 真紀, “世代や組織を超えた相互学習を促進するためのプログラミング導入教育の実践報告,” 情報処理学会研究報告. コンピュータと教育研究会報告, vol. 2012, no. 14, pp. 1–7, 2012.
4. 阿部 和広, “小学生からはじめるわくわくプログラミング,” 日経BP社, 2013.

レポジトリ情報

<https://github.com/jointry/jointry>

