Pressing: 打鍵の強さで出力が変わるビジュアルインタプリタ

加藤 淳[†,††] 五十嵐 健夫[†,††]

Pressing: A Pressure-sensitive Interpreter with Visual Feedbacks

Jun Kato[†,††] and Takeo Igarashi[†,††]

1. はじめに

キーボードは，Human-Computer Interaction（HCI）研究が進歩して様々な入力装置が提案されてもなお，コンピュータへ精確に指示を出せるインタフェースとして使われ続けている．そこで，打鍵に付随する情報を取得してインタラクションに活かす研究が進められてきた．例えば，打鍵の強さからユーザの感情を推論したり，個人認証に用いるアイデア[1]が提案されてきている．一方で，テキストベースのプログラミングにおいては，一般に正確な出力を得られるという信頼性が重視されるため，敢えてテキストに付加情報を与える試みは見られない．しかし，LOGO[2]に端を発する教育用途のビジュアルインタプリタにおいては，プログラミングの本質を損なわない範囲で，ユーザに面白いプログラミング体験を提供することが求められる．

我々は，テキストが表す精確な情報に打鍵力というファジィな情報を加えることによって新しいインタラクションを実現できると考え，プログラミング環境“ Pressing ”を開発した．Pressing は，物理シミュレーションによる視覚的フィードバックがあり，打鍵の強さが出力に影響を与えるインタプリタを内蔵したプログラミング環境である．マウスカーソル（マウス，タブレット PC またはタッチパネル）による補助的な操作をサポートしている．

† 東京大学 情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo
†† JST ERATO 五十嵐デザインインターフェースプロジェクト
JST ERATO IGARASHI Design UI Project

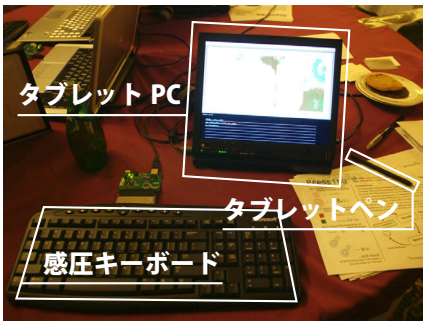

図 1 Pressing の動作環境

2. Pressing

2.1 概 要

Pressing を起動すると，仮想環境“ Hakoniwa ”とインタプリタ“ Kotosaka ”の入力欄が表れる．Kotosaka にスクリプトを入力すると，各文字を打鍵した強さの情報が保持され，スクリプトの実行時に Hakoniwa で打鍵力に応じた出力が得られる．例えば，*new Ball()* と強く打つと大きく，弱く打つと小さいボールが現れる．我々は，**図 1** に示す通り打鍵力の検出に Microsoft 社の研究用プロトタイプ Pressure-sensitive keyboard[3]を用いたが，ノート PC の内蔵加速度センサを用いる手法[1]でも同様のデモ環境を構築できるだろう．

Hakoniwa は 2D 物理シミュレータ Box2D[4]の Java 版実装を内蔵している．また，Kotosaka は JavaScript に影響を受けた簡易手続き型言語を解釈する．実装の詳細については公式サイト☆を参照されたい．

2.2 Pressing の機能と使い方

2.2.1 物体を表すクラス

Ball と Box は物体を表し，インスタンス化するとそ

☆ Pressing: http://digitalmuseum.jp/software/pressing/

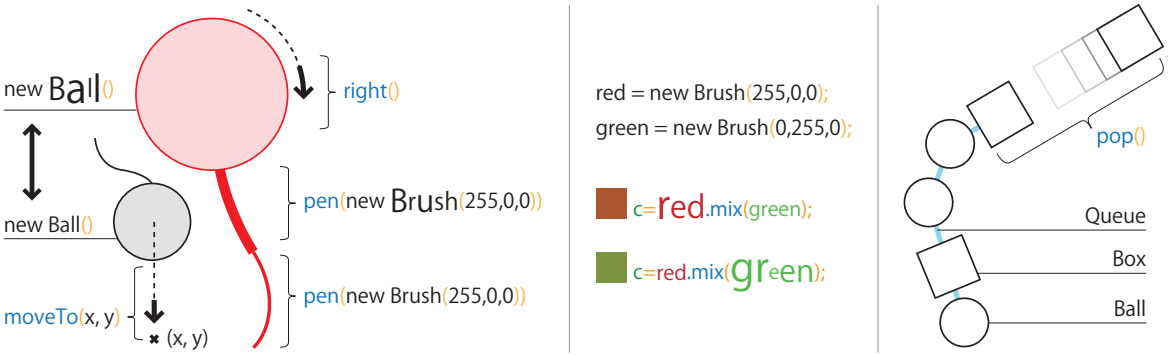

図 2 Pressing の使い方例

れぞれ円と矩形が Hakoniwa に現れる．そのときの打鍵が強ければ大きな物体が，弱ければ小さな物体が現れる．これらのクラスは次のようなメソッドとフィールドを持つ．

forward(), backward(), left(), right() 物体を前後進，左右回転させる．打鍵の強弱で速度が変わる．

moveTo(x, y) 物体を指定したスクリーン座標へ徐々に移動させる．打鍵の強弱で速度が変わる．

pen(brush), nopen() 物体の軌跡を描画するブラシを設定する．引数を与えないとデフォルトのブラシが使われる．打鍵の強弱でブラシの太さが変わる．*pen(null)* または *nopen()* が呼ばれるとそれ以上軌跡を描画しない．ブラシについてはは次節で詳述する．

behavior シミュレータが進むたび，引数に物体が渡された状態で呼ばれるコールバック関数．これは物体の恒常的な振る舞いを指定するために用意されたフィールドであり，例えば，物体を円を描くように移動させるアニメーションが可能となる．

2.2.2 ブラシを表すクラス

プリセットクラスの Brush は物体の軌跡を描くブラシを表す．ブラシは色と太さという 2 つの属性を持つ．インスタンス化の引数で色が，打鍵の強さで太さが決まる．

ブラシの *mix(brush)* メソッドを用いると，既存の二つのブラシを混ぜて新しいブラシを作れる．すなわち，二つのブラシの中間の色と太さを持つブラシのインスタンスが返される．メソッドの主体と引数のどちらに近いブラシが生成されるかは，それぞれの名前をどれだけ強く打鍵したかによる．

2.2.3 データ構造を表すクラス

プリセットクラスの Stack，Queue，BinaryTree はデータ構造を表す．インスタンス化するだけでは何も表示されないが，複数の物体を格納すると，物体同士を結ぶ半透明のバネとして描画される．

データ構造のインスタンスは，データ構造に固有の，要素を出し入れするメソッド（*push(e), pop(), insert(e), remove* など）を持っている．要素がデータ構造に挿入されるときは引力が，取り出されるときは斥力が，要素と構造の間に働く．メソッド名を打つときの打鍵が引力または斥力の大きさを決める．すなわち，Queue で *pop()* と強く打つと高速に，弱く打つと低速に，要素がデータ構造から打ち出される．

2.2.4 マウス操作

Pressing は，キーボード操作に加え，マウスカーソルで直感的に Hakoniwa とインタラクションすることができる．例えば，物体をドラッグしたり，クリックして選択したのち Kotosaka で *selected()* と打ってその選択結果を利用することができる．

3. おわりに

我々は，ビジュアルインタプリタに暗黙の打鍵力パラメタを追加することで，ユーザの力加減を出力に反映できるプログラミング環境を開発した．物理シミュレーションを用いて，ユーザが仮想世界と直感的にインタラクションできることを目指した．

参考文献

1) 岩崎健一郎，味八木崇，暦本純一：ExpressiveTyping: 本体内蔵型加速度センサによる打鍵圧センシングとその応用，WISS2008 論文集，pp. 91–94 (2008).
2) Logo Foundation. `http://el.media.mit.edu/logo-foundation/`.
3) Dietz, P., Eidelson, B., Westhues, J. and Bathiche, S.: A practical pressure sensitive computer keyboard, *Proc. UIST'09*, ACM, pp. 55–58 (2009).
4) Box2D. `http://www.box2d.org/`.