

並列処理によるペントミノ箱詰めパズルの高速化

144218 ネットワークメディア学科 上野 公輔 (荻原研究室)

1 はじめに

ペントミノは 5 つの正方形を辺に沿ってつなげた図形で、ピースの形は回転・鏡像によって同じになるものを同一と考えると 12 種類ある。

箱詰めパズルとは、例えば 6×10 の大きさの箱に、12 種類全てのペントミノを隙間なく収める方法を求めるものである(図1)。箱の大きさなどで問題にはいくつものバリエーションがあるが、多くの問題において、解の総数を手で求めることは難しい。そのため、古くからコンピュータによって解の探索が行われてきた[1]。

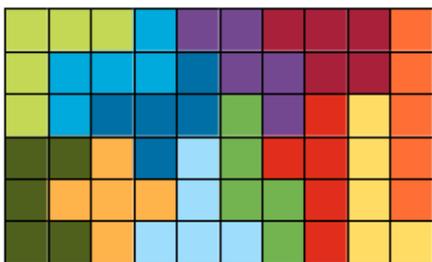


図1 12 種類のペントミノのピースと解の例

2 目的

箱詰め問題の一般的な解法はバックトラッキングである。ピースを回転、反転させながらボードの左上から可能な限り配置して行き、失敗した場合は後戻りを繰り返す。

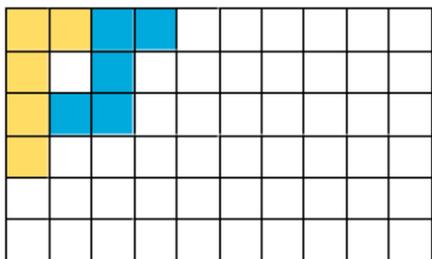


図2 間違った配置の例

しかし、この手法を単純に適用すると、明らかに間違ったピースの配置であってもそのまま探索が進んでしまう場合がある。たとえば、図2は埋められない小さな隙間ができてしまう例である。

本研究では、配置したピースが不適切なものかどうかを、並列動作する別のスレッドで調べることによって無駄な探索を排除する方法を提案する。

3 実験と実行結果

解を探索するスレッドと並列に、埋められない穴の有無を調べるスレッドを実行する(図 3)。探索スレッドと検査スレッドは、生産者・消費者問題のデザインパターン[2]を利用して連携させる。

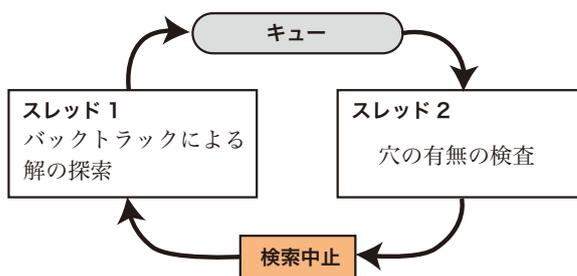


図3 実装したプログラムの処理の流れ

バックトラックのみの従来手法と、提案手法で箱詰め問題を解いた際の結果の例を示す(表1)。

表1 実行結果

	試行回数	処理時間(ms)
従来手法	127770343	8576
提案手法	12371446	28206

4 まとめ

試行回数を大幅に削減し、正しい解を得ることができたが、目標とした高速化は達成できなかった。データの受け渡しなどのオーバーヘッドを取り除いて行くことが必要と考えられる。

参考文献

- [1] 有澤誠, “プログラミング・レクリエーション ソフトウェア実習のガイド”, 近代科学社 (1978).
- [2] 結城浩, “Java 言語で学ぶデザインパターン 入門【マルチスレッド編】”, SB クリエイティブ (2006).

ARを利用したモバイル端末向け双六アプリの開発

学生証番号 144777 インテリジェントシステム学科 谷村 友優 (荻原研究室)

1 はじめに

ARの機能をモバイル環境でも利用できるようになってきたが、ゲームアプリでの使用例は多くない。ARとBluetooth通信を組み合わせれば、その場にいる人同士で楽しめるゲームが作成できると考えた。

2 研究の背景・目的

アプリケーションのテーマでは、昨年、一昨年の卒業研究として取り上げている[1][2]。AR双六アプリケーションは一昨年の「双六アプリケーション」を参考に行った。一昨年の双六アプリケーションは子供の機嫌を取るために勝たせるアプリケーションであった。

今回はARを取り入れることによって、双六の表現の幅が広がり、より多くの人を楽しませるアプリケーションが出来ると考えた。

3 手段

本研究では、Bluetoothをプレイヤー間でのゲーム進行に関わる情報のやり取りに使う。実装にはMultipeerConnectivityフレームワークを利用した。

ARを利用するためにmetaio社[3]が提供する無料のAR開発キットの中にあるiOS用のフレームワークを利用する。このフレームワークがあると、あらかじめアプリケーション内に用意した3Dモデルを様々な方法で出現させることが出来る。

4 アプリケーション概要

本研究では、iPadを3台使用して実験した。

各プレイヤーがiPadを持ち、カメラ越しにARマーカーを見ると盤面とキャラクターが表示され、ゲームを行う(図1)。

ゲーム中はBluetoothで通信が行われており、通信によってサイコロの出目や順番に関する情報のやり取りを行う。

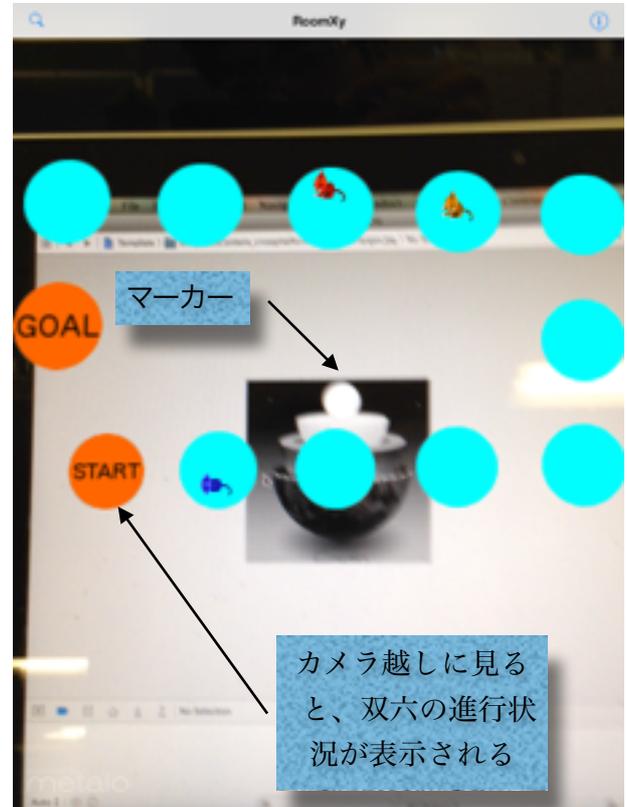


図1.ゲームの実行例

5 考察

iOSを一人1台もって行うため他の人に渡してまわす必要がなくなった。ARを用いるので、マーカーの角度で見え方が変わり、楽しめるので飽きにくい。しかし、一昨年の双六アプリと比べると、iOSのカメラやBluetoothの機能を使うので消費電力は激しい。

その場にいる人同士でこのAR双六アプリを利用する事は出来るが、通信の接続方法がややこしい。子供が遊べるアプリにするためには、特にこの点の改良が必要であろう。

参考文献

- [1] 湯川隆介, 子供と遊ぶ事を目的とした双六アプリケーション開発, 京都産業大学コンピュータ理工学部 卒業論文, 2013.
- [2] 呼思楽, Bluetooth通信を利用したカードゲームアプリの開発, 京都産業大学コンピュータ理工学部 卒業論文, 2014.
- [3] metaio, metaio | SDK Overview, <http://www.metaio.com/products/sdk/> (2015)

Clang を利用したリファクタリング支援ツールの作成

学生証番号 145181 コンピュータサイエンス学科 藤原 匡輝 (荻原研究室)

1 はじめに

リファクタリングは外部から見た動作を変える事なくソースコードを変更し、その複雑さを改善する為に行なわれる。しかし、既に正しく動作しているプログラムを手動で変更する事は、新たなバグの発生リスクを伴う。そこで必要とされるのが、自動でリファクタリングを行なうツールである。本研究では Clang[1] とそれに付属するライブラリを用いて C 言語のリファクタリングを支援するツールを作成した。

2 Clang

Clang は、Apple を中心として開発されている、C、C++、Objective-C 向けのコンパイラ・フロントエンドである。Clang にはコンパイラとしての機能の他に、コードを解析する機能を LibTooling と呼ばれる付属ライブラリとして提供している。これは Clang の字句解析・構文解析等の機能を直接呼び出し、利用するものである。本研究では、構文木に対して操作を行うので、構文解析の部分を使う。

3 ツールの実装

```

3 int f(int i) {
4   int j;
5
6   if(i == 1){
7     printf("hello");
8     j = i * 2;
9   }
10  else {
11    printf("hello");
12    j = i * 3;
13  }
14  return j;
15 }
16 }

```

```

3 int f(int i) {
4   int j;
5
6   printf("hello");
7   if (i == 1) {
8     j = i * 2;
9   } else {
10    j = i * 3;
11  }
12
13  return j;
14 }

```

図 1. if 文の重複コードの抽出

実装の手順は、リファクタリングの対象となる構文要素を構文木表示コマンドで調べ、その

要素を引数に取る関数を実装、という流れで行う。今回は参考文献[2]より if 文の重複コードの抽出(図 1)の他、独自に 3 種類のリファクタリングを実装した。また、ツールを Emacs から実行し、リファクタリングの部分適用を可能にした。

4 評価

本ツールは、コマンド一つで実行ができる。そのため、大きな関数を記述し終わった時や、コードを保存した時、リポジトリへコミットする直前など、所謂「一区切りついた」時にツールを実行すれば、それだけでコードを改善することが可能である。また、IDE に付属するツールと比較すると、本ツールのほうが Emacs 上で動作するぶん学習コストが低く、より簡単にツールの利用ができると考えられる。

5 まとめ

本研究では Clang 及びその構文解析ライブラリを使い C 言語の解析をし、またソースコードを書き換えることでいくつかの自動リファクタリングを実装し、その有用性を示した。急速な成長を続ける Clang を基盤とすることは、様々なパターンに対応したツールを実装する大きな手助けとなった。また、C++ に慣れているプログラマであれば、ツールへのパターンの追加は容易であり、拡張性は高いと言えよう。

参考文献

- [1] "clang" C Language Family Frontend for LLVM. <http://clang.llvm.org/>
- [2] FOWLER, M. Refactoring: Improving the Design of Existing Code. Addison Wesley Professional, 1999.

スマートフォン特有の操作性を生かした 計算機アプリの開発

学生証番号 145361 コンピュータサイエンス学科 安田 悠輝 (荻原研究室)

1 はじめに

本研究では、タップ(画面にタッチする)、スワイプ(画面にタッチしたまま指を移動する)など、スマートフォンの操作性を生かしたインターフェースによって利便性を向上させることを目的とし、計算機アプリの開発を行った。アプリ開発には、`enchant.js`[1]、`PhoneGap` (Apache Cordova)[2] を用いた。機能としては、四則演算の

ほか、計算式を記憶させて計算を楽に行う機能や、数値をメモリに追加するときに1つ1つ別々の値として保存できる機能を備えている。



図1. タップ操作によるボールの色の变化



図2. 青いボール(5)を黄色いボール(7)にぶつけて
黄色いボールの値を変化させる(7-5=2)

2 アプリの概要

数値などの情報が入ったボールを移動させたり、ぶつけたりして計算を行う。四則演算機能(図3, 4)、計算式作成/使用機能(図5)、メモリ機能(図5)、税込計算機能が存在する。使用例としては、買い物時の会計の概算や、簡単な統計データを用いた数値計算などがある。

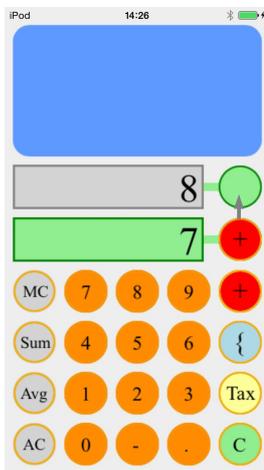


図3. 8+7の計算方法

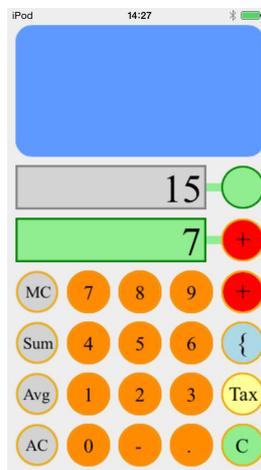


図4. 8+7の計算結果

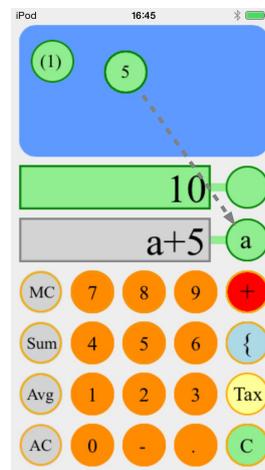


図5. 計算式 a+5 にメモリプール内の数値 5 を代入

3 評価、考察

計算結果を後で計算に使いたいとき、同じような計算を何度か行うときに有効である。しかし、アプリの画面を見ただけでは操作方法が分かりにくく感じる場合があるので、ボールをタップした時にどこにスワイプすればよいかなどの指示を出す設定をするといった改善点が挙げられる。

4 まとめ

独特のメモリ機能や、簡単な計算式を作成して変数に数値を代入するだけで計算ができる機能などが実現できた。しかし、誰にでも使いやすいかという点については、何人かの人に実際に使ってもらい、評価を受ける必要がある。

参考文献

- [1] 株式会社ユビキタスエンターテインメント：“enchant.js”，
<http://enchantjs.com/ja/>, (2014)
- [2] Asial Corporation：“PhoneGap Fan”，
<http://phonegap-fan.com/about/>, (2012)

双方向コミュニケーションが可能な プレゼンテーション支援ツール

学生証番号 145505 コンピュータサイエンス学科 和田 祥吾 (荻原研究室)

1 はじめに

従来のプレゼンテーションの方式では、発表者が一方的に情報を発信するだけで、発表者と聴衆が双方向にコミュニケーションを取る手段が存在せず、双方向性の課題が存在している。

本研究では、発表者が聴衆の意見や考えを把握できる機能を備えたプレゼンテーションツールを開発した。聴衆は意見を述べたり、質問することでプレゼンテーションに参加できる。

2 実装した機能

類似研究として、黒河ら[1]は、プレゼンテーションでの意識と情報の共有を目的として、スライド同期とコメント機能を実装している。これらの機能は本研究の目的にも利用できるため、本ツールにも実装することとした。実装には Node.js[2]とそのモジュールであり、幅広いブラウザで動作する Socket.IO[3]を利用した。

本ツールの主な機能の概要を示す。

- スライド同期
 - 発表者がスライドのページを進めると聴衆の資料も同様に進む機能。
- コメント機能
 - 聴衆が不明点などを質問できる機能。
- アンケート機能
 - アンケートを取り、集計する機能。
- 感想機能
 - 聴衆が発表者のプレゼンテーションに対する評価や意見を行う機能。

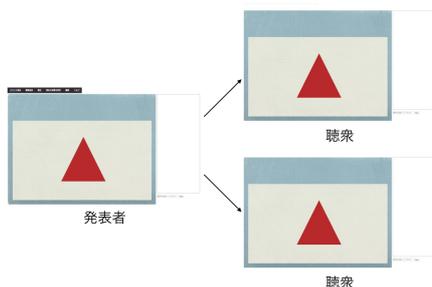


図1 スライド同期

- [JEWjk8HiQj2CK0CUAAAB] : ○○の部分について、もう一度説明してくれませんか？
- [uzdbkm0mmepbdkfHAAAC] : ○○がよく分かりません。
- [q0YgbPMn4t64NB0zAAAD] : スライドに誤字がありますよ。

図2 コメント機能

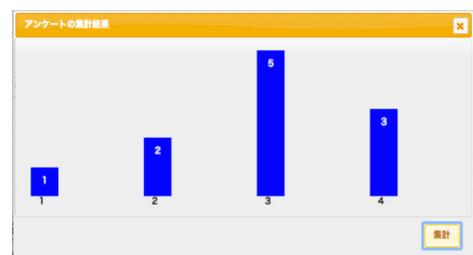


図3 アンケート機能

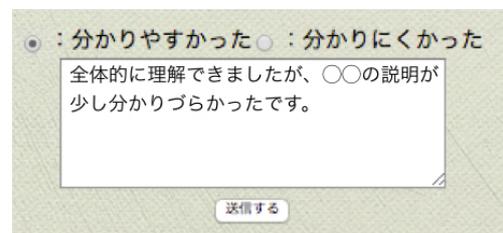


図4 感想機能

3 まとめ

従来のプレゼンテーションの方式では難しかった、発表者と聴衆が双方向にコミュニケーションを取れるツールを開発した。しかし本研究では大人数で本ツールを利用する実験を実施しておらず、サーバの負荷などを確認する必要がある。

参考文献

- [1] 黒河, 藤枝, 福本, 関戸, 服部, 萩野 : "Web を用いた参加型プレゼンテーション", 全国大会講演論文集 第 70 回平成 20 年(4), 4-371 - 4-372, (2008)一般社団法人情報処理学会
- [2] Joyent, Inc : Node.js <http://nodejs.org>
- [3] LearnBoost : Socket.IO <http://socket.io>