

攻茶花電

制作者：由井 卓哉

はじめに

2016年7月6日、アメリカ・オーストラリア・ニュージーランドの限定公開から始まった『Pokémon Go』（ポケモン GO）は、世界的に大ヒットアプリケーションとなった。ギネス世界記録によると、

1. 「初月にもっとも収益を上げたモバイルゲーム(\$2億650万)」
2. 「初月にもっともダウンロードされたモバイルゲーム(\$1.3億回)」
3. 「初月に国際ランキングでもっとも多くのダウンロードトップを動じ獲得したモバイルゲーム(70カ国)」
4. 「初月に国際ランキングでもっとも多くの収益トップを動じ獲得したモバイルゲーム(55カ国)」
5. 「もっとも早く収益\$1億をあげたモバイルゲーム (20日)」

と幾つもの記録を打ち立てている。

この様な Pokémon Go の素晴らしい成功にあやかって開発の成功と大会での好成績を願いたい。

本開発が将棋思考 Engine (AI player) であり、複雑で高速な処理が求められるため、開発に用いるプログラミング言語の選択は重要な検討事項である。

プログラミング言語は、設計思想、利用目的、求める性能、言語文法、開発環境などのトレードオフをどの様に図るかで多種多様な選択があり、そのためのアイデアによって、数多くのプログラミング言語が作成されている。

コンピュータ将棋関係でも多様なプログラミング言語が採用されており、『Bonanza』や『Nine Days Fever』の用いている『C』言語や『Ponanza』や『Apery』『GPS 将棋』『やねうら王』『技巧』をはじめ数多くのコンピュータ将棋で選択された『C++』言語など、開発効率よりもパフォーマンス重視の選択が広く見受けられる。

しかしながら、multicore 環境が一般になり、近い将来、さらに多くのコア数を対象とした開発が迫られる事は想像に難くない。

この様な中では、当初から multicore を前提とした言語設計の優位性も考えられ、他のプログラミング言語の選択も一考に値する。

2016 年 8 月 18 日に、Version 1.7 が公開されたばかりのプログラミング言語に『Go』がある。

Pokémon Go の成功を受け、本開発のプログラミング言語としてこの『Go』を選択し、『Go』言語の設計思想にそった将棋プログラミングを行う。

1. プログラミング言語『GO』

プログラミング言語『Go』(golang)は Google 社の Robert Griesemer (ロバート・グリズマー) , Robert C. Pike (ロバート・C・パイク;Rob Pike) そして Kenneth Lane Thompson (ケネス・レイン・トンプソン;Ken Thompson) らによって設計・開発された言語である。

Robert Griesemer は、Google Chrome を陰で支える『V8』の開発関係者であり、Ken Thompson と Rob Pike は、世界中の言語を扱うためのコンピュータでの文字規約 UTF-8 の考案・策定者でもある。

また Rob Pike は、現在一般化した GUI の window system を Unix 向けに作成した事で知られ、『Plan9』OS や『Inferno』OS、そして『Limbo』言語の開発者として名高い。

そして Ken Thompson は、計算機科学のみならず幅広く IT 業界で有名で、特に『Unix』『B 言語』『C 言語』など、

Dennis Ritchie と共に現在のコンピュータ及びその環境の基礎の設計・開発で知られる。

また、Joseph Condon と主に『Belle』のハードウェア及びソフトウェアの開発を行った事は、電王戦としては特にここで記述するに値するだろう。『Belle』は 1970 年代にコンピュータチェスのワールドチャンピオンなり、また、4-6 駒の完全な endgame tablebases の列挙を完成させた。さらに、Ken Thompson は、1983 年に Turing Award の受賞など歴史に名を残す計算機科学者として知られる。

2007 年に 3 者によって『Go』が生み出され、2009 年 11 月に発表された。バージョン 1.0 が公開されたのが 2012 年 3 月と比較的最近の言語であり、近年のコンピュータを取り巻く世界観の反映や、プログラミングの流行や変化を取り込んだ設計になっている。

たとえば、『Go』の公式 FAQ では、プロジェクトの目的の説明に『C』や『C++』のコンパイル問題や multicore 計算化による混乱や心配の台頭を挙げ、これらの問題に挑戦する新しい言語の価値を考え推進された事が書かれている。また、Rob Pike は、2012 年の SPLASH conference の基調講演で、Google 社での『Go』以前の C++, Java, Python を用いた開発に伴う困難や懸案事項として現実問題例を挙げながら、苦痛点を列挙した。「ビルドの遅さ」「制御できない依

存関係」「プログラマー各々が用いる言語の部分の違い」「貧弱なプログラム理解（コード読解の困難、貧しいドキュメント化など）」「努力がダブる」「update のコスト」「バージョンの歪み」「自動化ツール製作の困難さ」「言語をまたぐビルド」とまとめている。

それ以外にも、『Go』言語では並行計算処理や、**garbage collection** など重要機能の言語レベルでのサポートを取り入れ、手軽で高効率な開発、言語習得の容易化を行いながらも『C』や『C++』に迫る性能の実現を追求している。

2. 並行性

『Go』の特色として外せないモノとして並行性のサポートがある。

これは、『C』言語などが生まれた時代背景と異なる現代の計算機ハードの事情や、広くネットワークを介した処理が求められるなど、コンピュータの利用方法・利用目的の変遷を反映した言語特色である。

前提として『C』や『C++』では逐次的に、一度に一つのみコツコツと処理をする事を基本として設計されている。これは『C』が開発された当初では **singlecore** でない事を積極的に考慮する必要が無かったことから、高い妥当性と言語開発の面からも良かったと思われる。

しかしながら、時代を経るに従い単純な計算から複雑な処理過程が求めたりして、処理の順序が入り組み非明瞭になったりした。また、**multicore** の性能を引き出すためには、一度にいくつもの事を同時に処理する事が求められる様になり、並行性の重要性が増した。

並行処理は、逐次処理から単純には展開できない非常に難しい問題を孕む。例えば、『食事する哲学者の問題』や『居眠り床屋問題』など並列性に関しての問題が広く知られている。これらの問題は、並行処理では容易に予想外の結果を引き起こし、また、その結果が実行毎に異なる事がありえるなど処理結果の不安定さをもたらす。そして、それ故にバグなどがあっても問題点に気が付きにくくなってしまう。

これら並行処理特有の問題の難しさについては、様々な解決法や理論的な基礎が構築されて来た。『Go』では並行性に関する理論の一つである **Communicating Sequential Processes (CSP)** を元に **CSP** 風の並列性のサポートを言語レベルで行っている。

CSP に基づき、『Go』では並行処理での効率を高めるいくつかの仕組みを持ち、**channel** や **goroutine** など、簡単で安全性・安定性の高いプログラム作りが行える。実際に、Web サーバーなどの分野などでは、旧来の別言語製の元に比べ非常に高い性能を誇る実績がある。

3. 『GO』を取り巻く環境

開発において使用を試みる言語の利用状況などは一考の価値のある対象である。

その点では、『Go』は『C』や『C++』よりも利用され始めてから遙かに短い。

しかしながら、Google 社製で Google や Youtube での利用実績をはじめ、facebook や Twitter, Twitch, dropbox, bitly, eBay, Mozilla, Netflix, SpaceX, Tumblr, Uber など新聞でも見かけるような大企業や注目新興企業でも利用されており、さらに興味深いところでは GOV.UK とイギリス政府での利用もある。

また、開発が断続的かつ安定を指向している事も選択の良い材料となる。今回の ver1.7 への version up も含め、定期的に修正や改良が含まれる安定した更新が計画・実施されている。

そして、open source として仕様及び実装が公開されている。

4. コンピュータ将棋のハード環境

第 26 回世界コンピュータ将棋選手権を参考に近年のコンピュータ将棋で利用されているコンピュータハード環境についてまとめる。

『GPS 将棋』の 606core をはじめ、『技巧』の 294core 『YSS』の 288core など、manycore 化が顕著に見られる。

また、決勝戦に残ったハード構成では、『読み太』の 4core、構成が特殊で詳細不明な『大合神クジラちゃん』を除く、6 種のプログラムが 18core 以上のハード構成となっている。

さらに、優勝の『Ponanza』80core、準優勝『技巧』294core となっている。

5. 『GO』とコンピュータ将棋

前項で見たように、近年のハード構成として multicore/manycore 化に対応する事はコンピュータ将棋の重要な技術になると思われる。

その点で、『Go』は言語レベルでの平行性サポートや、その実績など優れた選択肢として考えられる。

また、第 26 回世界コンピュータ将棋選手権には『GPS 将棋』及び『HoneyWaffle』が『Go』のプログラミング言語としての採用を表明している。

本開発も上記の 2 作を追う形で、『Go』言語の全面採用を考えている。

6. 本開発の特徴

本開発では、近年一般化した評価関数の学習とそれを用いた局面探索の枠組みを採用。

また、前述の通り『Go』言語でのフルスクラッチと、CSP 風の並行指向を用いた設計で将棋思考 Engine の実装。