

仮想ゲーム空間を利用した プログラミング学習環境における言語処理系

A Programming Language Processor for a Programming Learning Environment Using a Virtual Game Space

赤川 啓行^{*1}, 竹内 章^{*2}, 國近 秀信^{*2}

Hiroyuki AKAGAWA^{*1}, Akira TAKEUCHI^{*2}, Hidenobu KUNICHIKA^{*2}

^{*1}九州工業大学知能情報工学科

^{*1} Department of Artificial Intelligence, Kyushu Institute of Technology

^{*2}九州工業大学大学院情報工学研究院

^{*2} Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology

Email: akagawa@minnie.ai.kyutech.ac.jp

あらまし:我々は、初学者に実践的プログラミング能力を獲得させるため、課題解決型学習をベースとした学習環境の実現を目指している。本稿では、学習環境の構成要素である言語処理系について述べる。本学習環境におけるプログラムは、タイルの組み合わせによって表現される。タイルは階層構造を持ち、学習者の状況に応じたプログラミングが可能である。

キーワード:プログラミング学習, 言語処理系, 仮想空間, ゲーム

1. はじめに

初学者を対象に広く用いられているプログラミング教育の一例として、基礎的知識を系統立てて配置した物をカリキュラムとし、講義や演習を通して対象者に順番に学習させる系統的学習がある。この手法は限られた時間や空間的制約において効率的に知識を伝達する事に長けている反面、現実的な条件との乖離による実践力の欠如を生む可能性がある。この問題を解決するため、本研究では、課題解決型学習をベースとした初学者用プログラミング学習環境の実現を目指す。本稿では、本学習環境の構成要素である言語処理系の実現について述べる。

2. プログラミング学習環境

問題発見から解決に至るまでの一連の行動はゲームの一環として行われ、全て学習者によって自主的に行われる。課題解決型学習の実施時に生じる主な課題⁽¹⁾とその解決方法を以下に示す。

2.1 学習意欲の維持

課題解決型学習では、より現実的な問題を扱う性質上、学習者は比較的長期間に渡り同一の課題に取り組み続けるため、その間学習意欲を維持する事が求められる。本研究では、プログラミングをゲーム内における行動を自動化するための方法として位置付けており、仮にプログラミングに対する明確な学習意欲がない場合でもゲーム内で優位に立つための手段として自然に課題解決に取り組むことができる。

2.2 前提知識不足による影響の緩和

学習者の知識不足が原因で十分な学習効果が得られない場合が考えられる。特に初学者を対象にする場合、プログラミングに必要な前提知識が不足している可能性がある。本研究では、タイルスク립ティング環境の導入によりプログラミングに必要とさ

れる前提知識を極力抑える。また、能力に応じた課題を適宜選択できる様、仮想空間内の課題を学習者自身に発見させる形式を採る。仮想ゲーム空間のイメージを図1に示す。

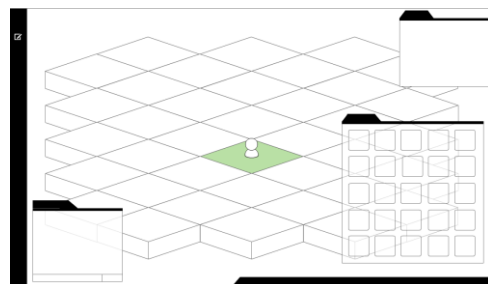


図1 仮想ゲーム空間のイメージ

3. タイルスク립ティング環境

学習者が課題を解決するための手段として、タイルスク립ティング言語の記述・実行環境を備えた開発環境を実現・提供する。本研究では、タイルスク립ティング言語の仕様策定を行った。策定に際しては、プログラミングに必要とされる前提知識を極力排除した。主な特徴として、処理の表現にタイルを使用して必要な操作を極力抑えた事による記述の容易化、それに伴う学習コストの引き下げ、および、制御フローのリアルタイム可視化が挙げられる。

4. 言語処理系の実現

本研究では、言語仕様の策定と併せて上述の言語を実行し得る言語処理系を実現した。

4.1 言語仕様

プログラムは図2に示すタイルによって表現され、記述はタイルの配置、タイル同士の接続、属性値の設定によって行われる。タイルは構文タイルと命令

タイルの2つに大別される。構文タイルは、条件分岐、変数操作、演算子など、プログラミング言語が備える基本的な構文に対応する。命令タイルは、学習者がゲーム空間に干渉、もしくは空間内の状況を監視する直接的な手段に対応する。構文タイルとして25種のタイルを定義しており、内24種のタイルが実装済みである。評価値の参照や継承関係を始めとするタイル同士の関係性は、タイルのコネクタ間の接続により表現される。タイルは0個以上のコネクタを持ち、コネクタは自身の親を含むタイルに接続される。

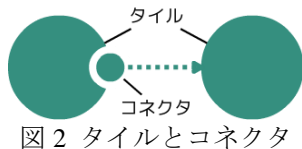


図2 タイルとコネクタ

命令タイルは、図3のような処理の抽象度に応じた複数の階層を持つ。これは、各学習者が持つ前提知識や取り組む課題の難度に応じた層のタイルを用いることで、徐々に難度の高い課題に取り組み、より高度なプログラミングに関する知識を獲得できるようにするためである。上層に属する命令タイルの例として、キャラクターを右に向かせる役割を持つタイル「右を向く」について説明する。このタイルを実行する際、内部的にはキャラクターの方向を操作する「向きを変える」というタイルに整数型の引数を渡し実行している。さらに下層を辿ると「キャラクターのインスタンスを取得する」等、仮想ゲーム空間の内部実装に極めて近いタイルを操作している。この様に命令タイルは上層ほど学習者の要求に近く扱い易いタイルが属しており、下層ほど内部実装に近く表現力に富み、より多くの知識が要求されるタイルが属している。

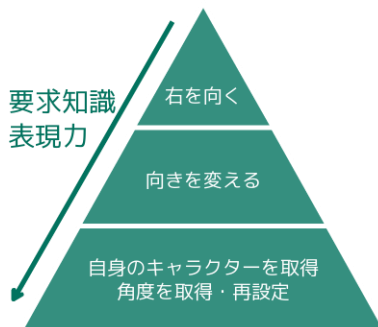


図3 命令タイルの階層構造

4.2 処理フローの制御

処理フローの制御にはコールスタックを利用する。あるタイルが実行される毎に対応したスタックフレームが発行され、コールスタックに積まれる事で実行待ち状態となる。スタックフレーム内には実行対象となるタイルや変数セット等、タイルの実行に必要な様々な情報が記録される。実行時に他のタイルの評価値が必要な場合は、タイル間でスタックフレームをやりとりする事によって受け渡す。

4.3 実行モード

関数呼び出しにおける引数の評価や計算式の実行等、あるタイルを実行する際、必要に応じて先に他のタイルを実行し、必要な情報を参照した上で実処理を行わなければならないケースが考えられる。本実装では各タイルが必要とされる全ての処理を予め定義した上でそれらに対応した実行モードを定義し、スタックフレーム側に実行モードを保持する事で、タイル自体の状態を遷移させる事なくその振る舞いを操作できる様にしている。これは同じ条件下で実行されている限りタイルの実行結果が変わらない事を意味し、ループ等により1つのタイルを複数回実行する場合に処理が煩雑化する事を防ぐ。

4.4 記述例

具体例として、キャラクターが「壁の手前まで前進する」動作を自動化する場合について考える。この動作は、キャラクター前方の地形を取得する「前の地形」タイル、壁属性を持つ地形を表す「壁」タイル、自身のキャラクターが現在向いている方向に前進させる「歩く」タイル、キャラクターを停止させる「止まる」タイル、および、それらを制御する構文タイルを用いて図4に示す形での記述が可能である。

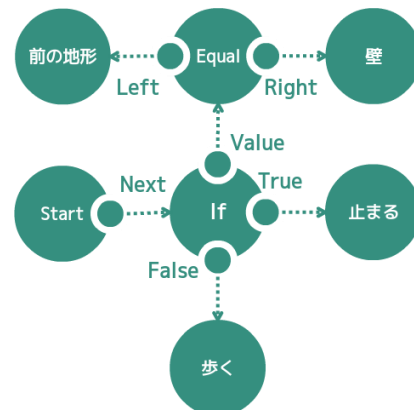


図4 タイルスクリプティングによる記述例

5. おわりに

本稿では、タイルスクリプティングを利用したプログラミング学習環境における言語処理系の実現について述べた。今後は、上述の言語で記述されたスクリプトを視覚化しつつ編集可能なエディタ、およびその実行環境である仮想ゲーム空間を実現する必要がある。加えて言語処理系において仮想ゲーム空間に干渉する手段である命令タイルの実現を行い、各モジュール間の連携が可能になった段階で評価実験を行う予定である。

参考文献

- (1) 先導的 IT スペシャリスト育成推進プログラム拠点間教材等洗練事業 PBL 教材洗練 WG: “PBL (Project Based Learning) 型授業実施におけるノウハウ集”, <http://grace-center.jp/wp-content/uploads/2012/05/pblknowhow20110726.pdf> (2012)