

LablGtk2 による universe teachpack の実装とユーザーテスト

上原 千裕 (指導教員: 浅井 健一)

1 はじめに

Racket の universe teachpack [1] には、簡単にゲームを作るための関数が一通り定義されている。本研究の目的は、Racket が押し進めるインタラクティブなゲームのプログラミングが、他の環境でも有効かどうかを調べることである。

そのため本研究では、LablGtk2 を用いて、universe teachpack と同じ働きをするライブラリを、静的型付けの言語である OCaml で実装する。これを universe library と呼ぶ。

また、学部 1～3 年生の 15 人に、universe library を使って通信ゲームを作ってもらおうというユーザーテストを行った。その結果、ライブラリにはある程度の実用性があることが確認できた。

なお本稿は、universe library の概要と実用性の説明を目的とするため、具体的な実装は述べない。Racket の universe teachpack の説明や、universe library の実装の過程については論文 [3] を参照されたい。

本稿で述べる universe library は <http://p1lab.is.ocha.ac.jp/~asai/Universe/> にて公開されている。

2 universe library の実装の概要

universe teachpack と同じ働きをするライブラリを実装するために 5 つのモジュールを実装した。

1. World モジュール (クライアントの作成に使う)
2. Color モジュール (色に関する処理を行う)
3. Image モジュール (画像に関する処理を行う)
4. Universe モジュール (サーバの作成に使う)
5. Socket モジュール (通信に関する処理を行う)

universe teachpack では、1 人で遊ぶゲームも複数人で通信をして遊ぶゲームも作ることができるため、universe library もそのように実装した。1～3 は、クライアントのプログラムを作るためのもので、通信ゲームか否かに関わらず、ゲーム作成時には必ず必要になるモジュールである。4 は、サーバのプログラムを作るためのもので、通信ゲームをする時のみ必要になる。5 は 1 と 4 で呼び出され、通信ゲームを作るための関数を 1 と 4 に提供する。1～5 の関係図は図 1 である。

3 ユーザーテストの説明

約 2ヶ月間、大学で週 1 回の授業を行い、学部 1, 2, 3 年生の計 15 人に universe library を使って通信ゲームを作ってもらった。1 年生 2 人は C 言語でのプログラミング経験はあるが OCaml のプログラミング経験はなかった。2 年生 10 人、3 年生 3 人は、OCaml でのプログラミング経験があった。チームを組んで行ってもらい、計 6 チームができた。

ここで、このユーザーテストの 3 つの目的を説明する。

1. universe library の実用性の検証

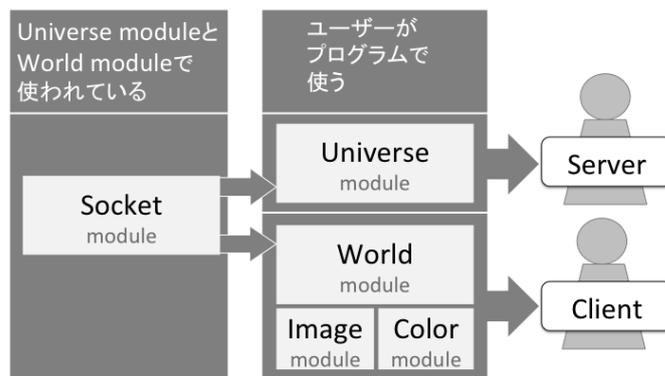


図 1: 各モジュールの関係

2. バグの発見と修正
3. universe library への簡単な機能の追加

universe library の作成後、それを使った簡単なゲームの作成は行ったため、簡単なゲームへの対応はできることが分かっていた。そこで今回は複数人に使ってもらい、実用性があるかを確認したかった。また、自分の環境で使うだけだと、環境に左右されるバグが発見されていないだけで隠れているかもしれない。以上より、1 と 2 を目的にした。

また、universe library には、ライブラリのごく一部分を理解できれば単純作業で追加できる関数が沢山ある。これらを全てライブラリ作製者が実装するのは時間がかかるので、被験者でライブラリ内部まで目を向けた人に実装してもらおうと考えた。そのため、3 を目的にした。

4 被験者が作った通信ゲーム

3 年生 3 人がチームで作ったものを紹介する。残りの 5 つのゲームは、第 4.2 節で簡単に紹介する。

4.1 ゲームの概要

4 人対戦で、自分の棒を矢印キーの操作によって動かし、ボールの跳ね返しと加速を行うホッケーのようなゲームである。ボールは複数個現れる。

ゲームの様子を図 2 に示す。4 人対戦の内の 2 人のクライアント (プレイヤー) の画面である。緑のプレイヤーが 1 点、青のプレイヤーが 4 点取っている。

4.2 他のユーザーが作った 5 つのゲーム

まずは 2 年生のチームのゲームについて紹介する。2 年生は 4 チームあるが、それぞれババ抜きゲーム、ガイスターゲーム、シューティングゲーム、インベーダーゲーム®️ をモチーフにしたゲームを作った。これらは全てホッケーゲーム同様、プレイヤーがゲームの見た目も内容も十分楽しめるものだった。

プログラミング初心者の 1 年生 2 人が作ったのは何人でも参加できる鬼ごっこゲームである。見た目、内容共に他のチームよりも完成度は低いものとなったが、プログラミング経験が少なく、OCaml のプログラミン

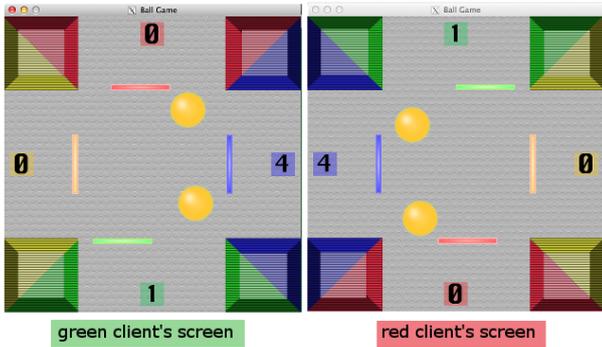


図 2: ゲームの様子

が経験がなかったことを考えると、遊べる通信ゲームが作れたことはこのチームの十分な成果である。

5 ユーザーテストの結果

第3節で挙げた目的に対する結果を述べる。ユーザーテスト後に行ったアンケートも交えて説明する。

5.1 universe library の実用性

全チームのゲームが正しく動いたことから、ある程度実用的であることが確認できた。ここで、プログラミング経験を考慮した実用性の考察をする。

2, 3 年生は、プレイヤーが十分楽しめるゲームを作ることができた。このことより、ある程度プログラミングの経験があれば、universe library を使ってしっかりしたゲームを作ることができる。

1 年生のチームは、プログラミングの初心者であったが、通信ゲームを作ることができた。このチームから分かることは、プログラミング初心者や OCaml を知らない人でも、2ヶ月で通信ゲームを作れるほど、universe library が簡単に使えるものだという点である。

学年別の考察からも、universe library はある程度実用的であること、また、簡単に分かりやすいものになっていることが分かる。

ユーザーテスト後にアンケートを行い、ライブラリについての意見を得た。利点、欠点に分けて紹介する。

5.1.1 universe library の利点だと感じたところ

1. 各モジュールの中身は知らなくても使える
2. 画像データを読み込み変数として使える
3. Racket の universe teachpack と同じ感覚で使える

1 が今回のアンケートで一番多く書かれた意見だった。特に、Socket モジュールの中身、つまり通信に関する知識が無くても使えるのが良いという意見が多数あった。また、2 については、LabiGtk2 の機能を使って実装しているため、LabiGtk2 を使うという選択が良かったと考えられる。3 については、OCaml で Racket の universe teachpack と同じ働きをするライブラリを作るといった目的が達成できたと考えられる。

5.1.2 universe library の欠点だと感じたところ

1. 音が出せない
2. エラーメッセージが分かりにくい
3. サーバが負荷により止まることがある

4. ゲームのウィンドウを縮小できるようにしたい

2 について説明する。第2節で作成したモジュールの中身を見ないままプログラミングをするため、中身を知らないモジュールのエラーメッセージが出てきても意味を理解しにくいとのことだった。

5.2 バグの発見と修正

ゲーム終了時など通信を切る時に、毎回ではないがエラーが発生した。本研究では、TCP のソケット通信を使っている [2, 4]。TCP において、コネクションを正しく切断する方法はない [2]。そのため、何らかの例外処理をしてコネクションを切らなくてはならない。このコネクション切断時の例外処理の実装が間違っていたことがエラーの原因だった。この間違いが通信を切るタイミングに依存してエラーを発生させていたため、エラーの発生が不定期だった。また、ライブラリの開発者の手元でゲームを動かしていた時は偶然タイミングが良かったため、バグが気付かれなかった。この例外処理の修正により、発生していたエラーは出なくなった。

5.3 universe library への簡単な機能の追加

被験者の約半数がこれをする予想していたのだが、実行したのは2人のみだった。呼びかけが足りなかったこと、また、ほとんどの人が画像データだけを使ってゲーム作成をしたため、楕円を描く関数などの追加が必要だと思われなかったことが原因だと思う。

6 まとめと今後について

本研究によって、Racket の universe teachpack とまったく同じではないが、本質的な部分はほとんど同じであるものが OCaml で実装できた。違いの例を述べると、universe teachpack では画像をプログラムに直接貼れるのに、OCaml を使っているためにそれが本研究ではできない。また、第5.1節でも述べたように、universe library はある程度実用的であることが確認できた。

よって、本研究は、universe teachpack によるプログラミングが OCaml においても有効であることを示す1つのケーススタディになった。

universe library はまだ開発途中である。多くの人に使ってもらうことで改良したいと思い公開をした。特に、第5.1.2節で挙げた欠点は、全て直したい。欠点で挙げられた4は、恐らく自力で解決できる。しかし、1, 2, 3 は、良い解決策が得られていない。そのため、色々な人に universe library を見てもらい、解決したい。

参考文献

- [1] Felleisen, M., Findler, R. B., Flatt, M., and Krishnamurthi, S.: A Functional I/O System or, Fun for Freshman Kids, *ICFP 2009*, (2009), pp. 47–58.
- [2] 小口正人: コンピュータネットワーク入門, サイエンス社, 2007.
- [3] 上原千裕, 浅井健一: LabiGtk2 を用いた universe teachpack の実装, 日本ソフトウェア科学会第 31 回大会 (2014 年度) 講演論文集, (2014).
- [4] 渡辺知恵美: システムプログラミング入門, サイエンス社, 2008, pp. 174–190.