

ゲームエンジンを用いたレスキューロボットコンテストのための 画面表示システムの実装と評価

○栗栖 輝（大阪府立大学）， 小島 篤博（大阪府立大学）

Implementation and Evaluation of Screen Display System for Rescue Robot Contest Using Game Engine

○Hikaru KURISU (Osaka Prefecture University), and Atsuhiko KOJIMA (Osaka Prefecture University)

Abstract: The game management system for Rescue Robot Contest consists of several subsystems. Screen Display System is one of the subsystems that presents the competition situation to participants and spectators. In this study, we implemented a Screen Display System using Game Engine Unity, which improved the versatility of the system and enabled rich expression of visual effects. In addition, the implemented system was put into practical use at the 19th Rescue Robot Contest to confirm that the system worked without problems from the viewpoints of stability and interoperability, and to evaluate the visual effects on the audience.

1. はじめに

都市部の大規模災害におけるレスキュー活動をテーマとしたレスキューロボットコンテスト（以下「レスコン」）[1]が毎年開催されている。レスコンでは、競技を円滑に運用するための独自の競技運用支援システム（以下「レスコンシステム」）[2]が運用されている。

レスコンの競技内容は、被災地を模擬したフィールド上で要救助者に見立てたレスキューダミー（以下「ダミヤン」）を遠隔操作型のロボットで救助するというものである。ダミヤンには救助活動の際に受ける痛みを検知するセンサが内蔵されており、そのセンサが検知した数値や救助の達成度などが各チームの得点に換算される。

本研究では、レスコンシステムのサブシステムの1つである画面表示システムを新たに実装し、第19回レスコンで実地運用し、評価を行った。レスコンシステムは、複数のサブシステムから構成されているが、そのサブシステムの1つに、競技チームや観客に対して競技状況等を提示する画面表示システムがある。観客への競技状況の提示は、会場のスクリーンを用いているが、会場のスクリーンは、観客向けに提示されるため、注目してほしい部分を強調するなど視覚的な演出が求められる。観客に向けた視覚的な演出を考慮した画面を構成するには、開発環境の選択も重要である。画面表示システムは、従来 Visual Basic で実装されていた。しかしながら、Visual Basic では、ゲーム開発で利用されるような視覚的な効果のある演出が期待できず、観客に向けた視覚的な演出を考慮した画面を作成するには限界があった。また、Visual Basic で実装

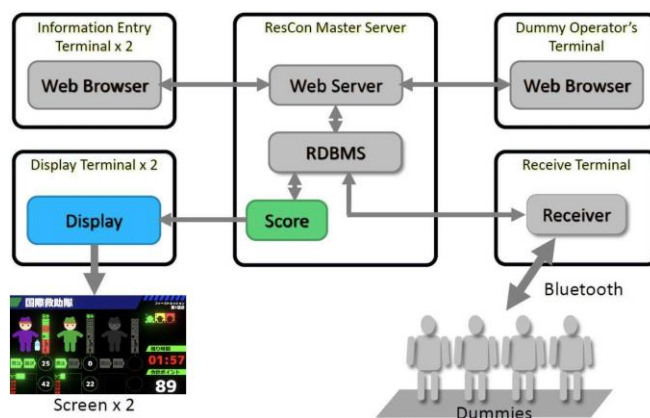


Fig.1 Game management system for Rescue Robot Contest

したシステムは、Windows でしか動作しないため、システムを運用する際の端末が限られるという問題もあった。

一方、近年スマートフォンなどのゲーム開発のために、ゲームエンジンと呼ばれる開発環境が提供され、ゲーム開発に必要な機能を容易に利用できるようになってきている。そこで、本研究ではゲームエンジン Unity を用いることにより、様々なプラットフォームに対応するとともに、観客に向けた視覚的な演出を考慮した画面表示システムを実装した。本報告では、実装したシステムが、安定性、相互運用性の観点から問題なく動作することを確認するとともに、観客に対する視覚的な演出の効果についての評価を行なった。

2. レスコンシステム

レスコンシステムは、Fig.1 のようにデータベースを中心として、複数のサブシステムから構成されている。

以下では、各サブシステムの機能について概説する。

- Master Server
後述する RDBMS, Web Server, Score が動作する主サーバ。
- RDBMS
競技の状況やダミヤンのセンサ値など、すべての情報を一括して管理する。
- Web Server
競技状況の入力を行うための Web ページなどを提供する。
- Receiver (ダミーデータ受信システム)
ダミーの痛みを受信し、それを数値化したものを RDBMS に記録する。
- Score (競技進行管理システム)
ダミーの痛み計算や競技進行の管理を行う。
- Display (画面表示システム)
競技状況などを競技チームや観客に対して表示する。

このうち、本研究で新たに実装したのは、Display (画面表示システム) である。これは、従来 Visual Basic で開発されていたものである。

また、第 19 回レスコンでは、レスキュー活動の際に新しい規定「支援物資提供」が導入された。「支援物資提供」は、支援物資に見立てたブロックをダミヤンの近くにある台に輸送することで、ダミヤンのダメージを回復させるという規定である。この規定が新しく導入されたため、システムもこの規定に対応させる必要が生じた。

3. 画面表示システムの実装

3.1 開発環境

画面表示システムの開発環境として、以下の 3 つの理由からゲームエンジン Unity を採用した。

- GUI ベースの直感的な操作で画面を作成できる。
- ゲーム開発に用いられる様々な表示効果により、観客に向けた視覚的な演出が実現できる。
- Windows, Mac, iOS, Android など、様々なプラットフォームに対応している。

次節以降、実際に実装する画面表示システムの概要を述べる。

3.2 画面表示システムの機能

レスコンは、レスキュー活動だけでなく、チーム紹介やプレゼンテーションなど、いくつかの手順で構成されており、以下の流れで行われる。



Fig.2 Score Window

1. チーム紹介
各チームの紹介が行われる。
2. プレゼンテーション
ロボットのコンセプトや救助方針などをプレゼンテーションする。
3. 作戦会議
フィールドの状況を確認しながら、救助方法やロボットの移動経路などの作戦を立てる。
4. レスキュー活動
各チームが遠隔操作型のロボットで要救助者に見立てたダミヤンを救助する。
5. 活動報告及び得点表示
レスキュー活動の結果報告、得点の表示を行う。

画面構成部では、この競技の進行に沿って、以下の 6 種類の画面の切り替えを行う。

- Main Window
画面表示システムを起動した際に、最初に表示される画面である。スタートボタンを押すと、パケットの受信を開始する。
- Stay Window
競技開始前や競技進行の待機状態などに表示される画面である。
- Intro Window
チーム紹介の際に表示される画面である。チームの集合写真が表示される。
- Presen Window
プレゼンテーションと作戦会議の際に表示される画面である。残り時間の表示を行う。
- DummyIndicator Window
レスキュー活動の際に表示される画面である。得点、時間など競技状況の表示を行う。詳細は 3.3 節で述べる。
- Score Window
レスキュー活動終了後に表示される画面である。Fig.2 に示すように、最終的なポイントと各ダ

ミーにおけるポイント、減点など、ポイントの詳細が表示される。

このうち、DummyIndicator Window について、次節で詳細を述べる。

3.3 DummyIndicator Window

レスキュー活動の際に表示される画面である。Fig.3 に示すように、以下のような競技に関するすべての情報が表示され、競技状況が確認できる。

- ダミーの状態（ダメージを受けると、ダメージの種別によって画像が変わる。）
- ミッションの達成状況、残り時間
- 容体判定結果（緑：正解、赤：不正解、グレー：無回答）
- ロボットの台数および各反則フラグの状態（緑：通常、黄：イエローフラグ、赤：レッドフラグ）
- 各種ポイント（フィジカルポイント、ミッションポイント）
- 各種センサ（ライフバー、ダメージバー）

レスキュー活動は、レスコンのメインとなる競技であり、情報量も多いため、最もスクリーンに注目が集まると考えられる。そのため、観客への演出を考慮するためには、この画面における演出に最もこだわるべきである。本研究では、Unity の機能 Bloom を視覚的な演出のために取り入れた。次節では、その詳細を述べる。

3.4 新たに実装した要素

Unity の機能 Bloom による演出を新たに実装した。Bloom は、光源から光が溢れるような演出ができる機能であり、明るさの調整も可能である。DummyIndicator Window では、容体判定結果、ミッションの達成状況やロボットの反則フラグの状態、各種センサを Bloom の機能により発光させ、インパクトのある表現を実現させた。

また、2 章で述べたように、第 19 回レスコンでは新しい規定「支援物資提供」が導入された。そのため、これに対応し、各ダミヤンにおける支援物資の提供状況を画面に追加した。

3.5 画面表示システムの構成

ここでは、画面表示システムの内部構成について述べる。画面表示システムは、競技状況などを競技チームや観客に対して提示するシステムである。Fig.4 が示



Fig.3 DummyIndicator Window

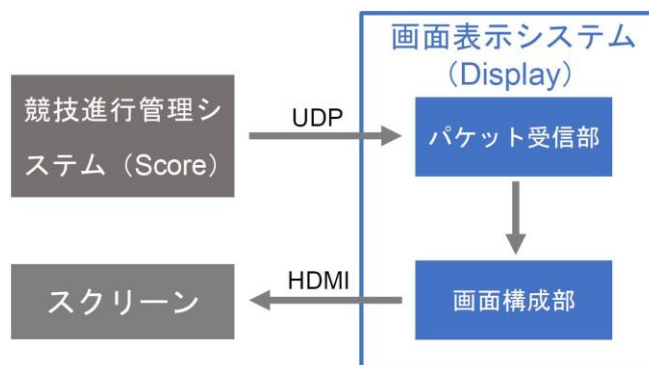


Fig.4 Configuration of Screen Display System

すように、画面表示システムは、以下の 2 つの部分から構成され、パケットの受信と画面の構成を切り離し、非同期で処理を行なっている。

- パケット受信部
2 章で述べた競技進行管理システムから画面表示を指示する UDP パケットを受信する。
- 画面構成部
受信した UDP パケットの情報を基に画面各部の構成や切り替えを行う。画面の切り替えについては 3.2 節で述べた。

競技進行管理システムから受信する UDP パケットには、チーム名や得点、時間などの情報が含まれている。また、画面構成部で構成した画面は、HDMI 経由でスクリーンに投影される。

前述したように、パケットの受信と画面の構成は非同期で処理を行っているが、パケットの受信は、メインスレッド外の受信スレッドで行われている。Unity で非同期処理を行う際に問題となるのは、メインスレッド外で Unity API を利用する場合である。その理由は、Unity には、Unity API をメインスレッド外では操作できないという制限があるためである。本システムでは、その問題を C# の async/await を用いて解決している。

Fig.5 は、その動作を表しており、受信スレッドでは、シングニチャに `async` 修飾子を付けた「非同期メソッド」内において、Unity API を利用する処理を `await` で指定してメインスレッドに渡し、その処理が完了するまで待機する。

4. 運用と評価

3 章で実装したシステムを、第 19 回レスコンの岡山予選、東京予選、および神戸本選において実地運用した。運用の際には、システムに問題が発生した場合に備え、従来の画面表示システムを予備機として動作させ、切り替えが可能な状態にしていた。本章では、予選・本選での運用状況、問題点、改善などを述べる。

4.1 岡山予選・東京予選

岡山予選、東京予選のいずれにおいても、システムが安定性、相互運用性の観点から問題なく動作することが確認できたが、予選全体を通して顕在化した問題が 2 つある。

1 つ目の問題は、3.4 節で述べた Bloom による光の効果についてである。岡山予選では、Bloom の光を明るくしすぎていたため、発光部分の文字が見えにくいという問題があった。また、東京予選では、光の効果がスクリーンに反映されなかった。光の明るさを大きくしても反映されなかったため、プロジェクタの特性によるものと思われる。予選での問題を踏まえて、神戸本選の際には、テスト時にスクリーンを通して光の明るさを調節することとした。

2 つ目の問題は、競技状況の入力者による入力ミスに対する対策がなかったことである。2 章で述べたように、レスコンシステムでは、Fig.6 のように、Web Server を通して競技状況の入力を行っているが、岡山予選の際、3.2 節で述べた得点表示において、入力ミスから正確な得点が Score Window に表示されなかった。それは、入力者が、Score Window に画面から、不正に前の画面に戻る操作をしたためであった。この問題を解決するため、神戸本選に向け、Score Window に切り替わった際にスクリーンショットをする機能を追加した。

4.2 神戸本選

神戸本選の際にも、システムが問題なく動作することが確認できた。また、4.1 節で述べたスクリーンショットもうまく機能していた。しかしながら、ここでも Bloom による光の効果について課題が残った。神戸予選でも意図した発光がスクリーンに反映されず、最小限の発光をさせるに止まった。そのため、従来のシステムとの視覚面における差別化はほとんどできなかつ

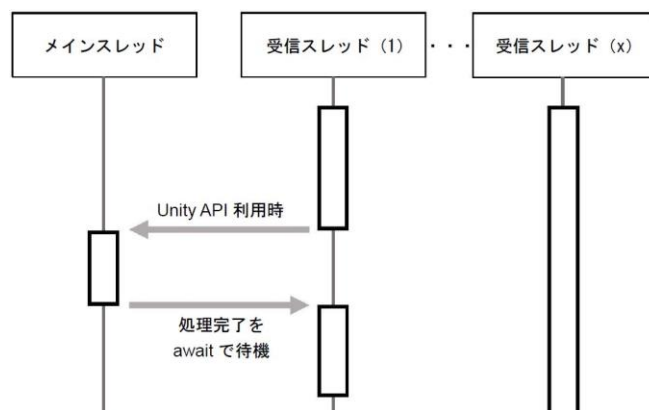


Fig.5 Asynchronous processing between threads



Fig.6 Data entry in the competition

た。

Bloom による光の効果は、実行時に光の明るさを調整できるようにするなど、新たな改善策を考える必要がある。また、Unity で利用できる機能は様々であるため、より効果的な視覚的演出の導入を検討する。

5. まとめ

本研究では、ゲームエンジン Unity を用いて画面表示システムを実装し、第 19 回レスコンで実地運用して評価を行なった。実装したシステムは予選、本選ともに、問題なく動作することが確認できたが、Bloom による光の演出については課題が残った。今後の課題としては、Bloom による光の演出を改善するとともに、Unity で利用できる機能を活かし、より効果的な演出を実現していくことが挙げられる。

参考文献

- [1] レスキューロボットコンテスト公式サイト：
<https://www.rescue-robot-contest.org>
- [2] 小島篤博，小枝正直，山内仁：“レスキューロボットコンテストのための競技運営支援システムの開発と評価”，電子情報通信学会論文誌 D, Vol. J93-D, No. 10, pp. 2317-2325 (2010)