

ゲーム専用ブラウザのためのフレームワークの開発

井上 寛生, 速水 治夫

神奈川工科大学 大学院 工学研究科, Project L.C.

lyude.sc@gmail.com

近年, SNS 上で提供され, ウェブブラウザ上で動作するソーシャルゲームが増え, そのゲームをより快適に遊ぶために専用ブラウザが開発されている. 専用ブラウザはゲーム内にはない新たな機能を追加できるようになっている. 一つのソーシャルゲームに対し複数の専用ブラウザが存在している. 数ある専用ブラウザは機能やユーザインタフェースが大きく異なるにもかかわらず, そのほとんどはレイアウトが固定されているため, ユーザの環境に適していないことが多い. また, 開発においても機能を追加するたびにユーザインタフェースを見直さなければならない. こうした問題に対し, 機能をプラグイン化することで機能の追加や削除を容易にし, プラグインごとにウィンドウを自動で生成する専用ブラウザと, プラグインを開発するためのライブラリを開発することで, 開発者とユーザの双方にとって有用なフレームワークの開発を試みた. 開発したフレームワークを実際に使用し, 専用ブラウザの開発における有用性やユーザに柔軟な操作ができるユーザインタフェースが提供できるか考察した結果を報告する.

キーワード ソーシャルゲーム, ブラウザゲーム, 専用ブラウザ, フレームワーク, プラグイン

1 はじめに

近年, SNS (ソーシャル・ネットワーキング・サービス) 上で提供されるソーシャルゲームが増えている. ソーシャルゲームはユーザ同士がゲーム内で繋がりを持つことができるオンラインゲームであり, その多くはウェブブラウザ上で動作する. ソーシャルゲームの増加に伴い, ゲームをより快適に遊ぶために専用ブラウザが開発されている.

専用ブラウザは, ソーシャルゲームを遊ぶことに特化した専用のアプリケーションを指し, 様々な機能を提供する. ソーシャルゲームの多くは, 公式が運営する WEB ページにゲーム画面が存在し, 広告や他のサービスといった直接ゲームに関わることのない情報が表示されている. 専用ブラウザではこれらの広告や余白を排除し, ゲーム画面のみを映し出すことができる. ゲームサーバとゲームクライアントの通信内容を取得し, 専用ブラウザで整形することで, ゲーム内とは異なる形で情報の表示が可能となっている. また, Twitter や Facebook にゲーム画面を投稿できる機能など, その他の SNS との連携や, 新たな機能を付与することができる.

これらの機能を持つ専用ブラウザは, 一つのゲームに対し複数の専用ブラウザが存在することがあり, ユーザは数ある専用ブラウザから一つを選択し使用しなければならない.

本稿では, 異なる開発者が様々な機能を追加できる専用ブラウザのためのフレームワークを試作し, 実際にフレームワークを用いて専用ブラウザとプラグインを開発することで実験とし結果を報告する.

2 問題点と解決策

2.1 問題点

現状で提供されている専用ブラウザは, 実装している機能やユーザインタフェースが大きく異なるが, そのほとんどはレイアウトが固定されている. ユーザによって実行環境やプレイスタイルは異なるため, 環境に合ったレイアウトに変更できると有益である. また, ユーザによっては異なる専用ブラウザの両方の機能を求める場合もある. 開発者はこうしたユーザのニーズに応えるために新たな機能を追加しなければならないが, 機能を追加するたびにユーザインタフェースを見直さなければならない.

2.2 解決策

これらの専用ブラウザの混在と, 機能の追加に対しかかるコストを軽減するために, 異なる開発者が様々な機能を追加できる専用ブラウザのためのフレームワークを提案する. フレームワークでは機能をプラグイン化し, 専用ブラウザに読み込ませることで機能の追加や削除を可能にする. また, 機能を追加する際に生じる開発コストを減らすために, プラグインを追加すると自動でウィンドウを生成する仕組みを実装する. プラグインに応じた専用ブラウザの仕組みの概要を図 1 に示す.

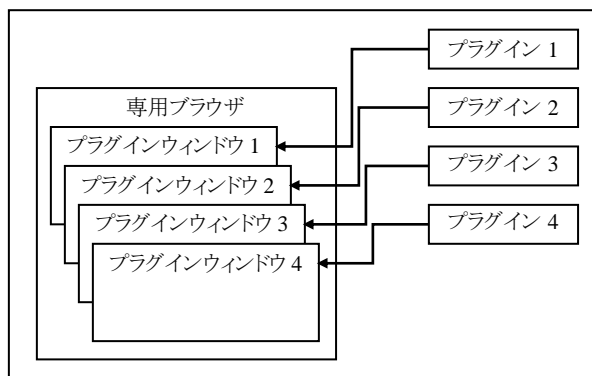


図 1 専用ブラウザとプラグインの仕組み

ユーザごとに実行環境やプレイスタイルが異なる問題に対し、専用ブラウザで読み込み自動的に構築されるウィンドウは、ドッキングにより柔軟な操作を行えるようにすることで、ウィンドウの配置や大きさをユーザが自ら変更をできるようにする。ドッキングウィンドウは分離可能なタブを持つウィンドウであり、ウィンドウの分割や上下左右の端で固定することができる。固定したウィンドウは自動で隠れ、マウスカーソルが固定したウィンドウのタブと重なると再び表示される。

3 フレームワーク

3.1 概要

フレームワークは専用ブラウザとプラグイン開発用ライブラリで構成される。専用ブラウザはプラグインを読み込む機能を備え、プラグイン開発用ライブラリは他の開発者が専用ブラウザに手を加えずにプラグインを開発できるように独立したものとする。

専用ブラウザは数々のユーザコントロールを扱うためコントロールを容易に扱うことができる Microsoft Visual Studio 2012, C#で開発し、.NET Framework 3.5 がインストールされている Windows で動作するアプリケーションとし、プラグイン開発用ライブラリは Microsoft Visual Studio で扱えるものとする。

3.2 専用ブラウザ

専用ブラウザはプラグインを読み込む機能と、読み込んだプラグインをドッキング機能のついたウィンドウとして配置し、自動でユーザインタフェースを構築する機能を実装する。また、アプリケーションの終了時にウィンドウの位置や状態を保存し、起動時に復元する機能を実装する。ドッキングウィンドウには MIT License によりライセンスされている DockPanel Suite を利用した。

ドッキングウィンドウは DockPanel Suite から提供される DockPanel コントロールに、DockContent を配置するこ

とで実装でき、DockContent は親となる DockPanel に対してのみタブ化や固定ができる仕組みになっている。しかし、専用ブラウザ側に DockPanel を実装し、プラグイン側に DockContent を実装する方法では、専用ブラウザとプラグインの両方に DockPanel Suite ライブラリを参照する必要があり、プラグイン開発用ライブラリのみでの開発ができなくなる。そこで、DockContent は専用ブラウザで実装し、読み込んだプラグインの数の DockContent を動的に初期化し、プラグインが持つユーザインタフェースを DockContent に配置することでプラグイン開発用ライブラリとの分離を図る。

DockPanel Suite ではウィンドウの状態は XML 形式で保存される。保存されているウィンドウの状態と、対象となるウィンドウは名前空間とクラス名により関連付けられている。通常、DockPanel Suite を利用する際は DockContent を継承した Windows フォームを扱うため、名前空間とクラス名を重複させずに管理することができる。しかし、専用ブラウザ側で動的に DockContent を初期化すると、DockContent の名前空間とクラス名はすべて同一になるため、保存されたウィンドウの状態と対象となるウィンドウを区別できず復元できなくなる。そこで、専用ブラウザ側で DockContent を継承したクラスを新たに作り、名前空間とクラス名を取得するメソッドをオーバーライドし、ウィンドウごとに固有の名前を付与できるようにすることで復元できるようにした。

3.3 プラグイン

プラグインは専用ブラウザで利用できる DLL (ダイナミックリンクライブラリ) を出力する。出力されたプラグインは専用ブラウザのプラグインディレクトリに配置することで機能を追加することができる。また、プラグインごとに内容を保存する必要がある場合は、保存用ディレクトリヘッダを保存でき、復元することができる。各プラグインは IPlugin インタフェースにより実装するプロパティが保証されている。IPlugin インタフェースで実装されているプロパティを表 1 に示す。

表 1 IPlugin インタフェースのプロパティ

名前	内容
Name	名前
Copyright	作者
Description	説明
AccessKey	アクセスキー
Version	バージョン
Icon	アイコン
Image	画像
ToolStripMenuItem	メニューバーアイテム
Control	コントロール

AccessKey プロパティは専用ブラウザ本体のメニューバーにプラグインの一覧が表示される際、キーボードに

よる入力を受け付けるためのキーを指定している。Image プロパティにはメニューバーやドッキングウィンドウで表示される画像が設定されており、Control プロパティにはプラグインのユーザインタフェースが設定されている。Version プロパティはプラグインのアセンブリ情報からバージョンを取得し、Icon プロパティは Image プロパティから生成されるようになっている。ToolStripMenuItem プロパティは Name プロパティや Image プロパティから自動的に設定され、同時にクリックイベントなども実装される。必要な場合は開発者がプロパティに変更を加えることも可能となっている。

4 実験

4.1 概要

開発したフレームワークを使用して実際にソーシャルゲームの専用ブラウザを開発することで実験とし、その結果を評価する。対象となるソーシャルゲームは角川ゲームス開発のブラウザゲーム『艦隊これくしょん -艦これ-』とした。艦隊これくしょんはプレイヤーが提督となり、戦時中に活躍した日本海軍の艦を擬人化した『艦娘』を集め、編成した艦隊で敵海域を攻略するシミュレーションゲームである。2014年4月19日時点で登録者数は190万人を超えており、ネットで大きな話題を呼んでいる^[1]。艦隊これくしょんの運営は専用ブラウザを取り締まっていないため、ゲームサーバへの過負荷や通信内容の改変によるゲームの不正な進行をしないことを条件に専用ブラウザの開発をする^[2]。また、他のブラウザゲームの専用ブラウザの開発シミュレーションとして、URL からWEB ページを開くことのできる簡易ウェブブラウザも開発することで、二つのプラグインの開発とする。

4.2 実装するプラグイン

専用ブラウザには『艦隊これくしょん -艦これ-』と『簡易ウェブブラウザ』の二つをプラグインとして開発する。また、艦隊これくしょんプラグインには、専用ブラウザで実装しているプラグインを読み込む機能と、プラグインをドッキングウィンドウとして生成し、配置できる機能を実装することで艦隊これくしょんプラグイン用の拡張プラグインも開発する。

4.3 実装手順

プラグインの実装は、Microsoft Visual Studio のソリューションにプラグイン用のプロジェクトを追加し、参照設定にプラグイン開発用ライブラリを設定する。Program クラスに IPlugin インタフェースを継承させ、プラグインに必要なプロパティを設定する。必要であればプラグインにユーザコントロールを作成し、Control プロパティに設定することでプラグインは専用ブラウザ上にウィンドウを持つことができる。プラグインのプロジェクトはテンプレ

ートとしてエクスポートすることで次回からはより簡単にプラグインを開発することができる。

4.4 簡易ウェブブラウザプラグイン

プラグイン内でユーザコントロールを作成し、ユーザコントロール上にボタンやコンボボックスを配置した。コンボボックスに URL を入力すると、入力した URL へ遷移する。簡易ブラウザの戻る・進む・更新といった動作が適切であることを確認した。例としてブラウザカードゲーム『ゴッドフィールド』のゲーム画面を図 2 に示す^[3]。



図 2 簡易ウェブブラウザで実行しているゲーム

4.5 艦隊これくしょんプラグイン

艦隊これくしょんプラグインには、ゲーム内で扱うデータのインタフェースやAPIを実装している。ゲームへのログイン完了と、パケットを監視しゲームサーバからの通信を検出したときに拡張プラグインへ通知するイベントを追加し、各拡張プラグインはログイン完了やゲームサーバとの通信を検出するとプラグイン内に記述された処理を実行する。パケットの監視には FiddlerCore を使用した。すべての拡張プラグインは専用ブラウザと同様にウィンドウを生成し、配置や大きさを変更することができる。初期レイアウトでは公式コミュニティ・ブラウザ・所持艦プラグインを画面中央にタブで分け、編成・遠征プラグインを画面右にドッキングしている。公式 Twitter やログインプラグインは画面左に隠れており、タブにマウスカーソルが重なると自動で表示される。艦隊これくしょんの画面を図 3 に示す。



図 3 艦隊これくしょんプラグイン

4.6 ログインプラグイン

艦隊これくしょんへのログイン機能を提供し、ログイン ID とパスワードを与えることでログインする。ログインが完了するとプラグインホストインタフェースを通して各拡張プラグインへログインが完了したことを通知する。

4.7 公式コミュニティプラグイン

公式コミュニティプラグインにはウェブブラウザコントロールが配置されており、艦隊これくしょん公式コミュニティページを表示することができる。このプラグインのコントロールは、プラグインのウィンドウが起動してから初めて表示されたときに初期化されるため、起動直後から無駄なメモリを消費しない設計となっている。

4.8 公式 Twitter プラグイン

公式 Twitter のタイムラインを表示する。公式コミュニティプラグインと同様に、コントロールが初期化されるタイミングはプラグインが初めて表示されたときになっている。

4.9 ブラウザプラグイン

艦隊これくしょんのゲーム画面を表示する。ログインプラグインにログインが完了したことが通知されると、ゲーム画面を取得しブラウザプラグインのウェブブラウザコントロール上に表示する。実際に艦隊これくしょんで遊ぶのはこのプラグインから行う。

4.10 資材・アイテムプラグイン

ゲーム内で所持している資材やアイテムの推移をグラフ化して表示することができる。ゲームサーバとの資材に関する通信を検出するとグラフの内容が自動で更新される。初期状態で資材・アイテムプラグインは画面上部にドッキングしており、マウスカーソルがプラグイン上から離れると自動的にウィンドウが隠れるようになっている。資材・アイテムプラグインの画面を図 4 に示す。



図 4 ドッキングしている資材・アイテムプラグイン

4.11 編成プラグイン

編成している艦隊の詳細を常に表示することができる。ゲームサーバとの編成に関する通信を検出すると画面が更新される。編成プラグインの画面を図 5 に示す。

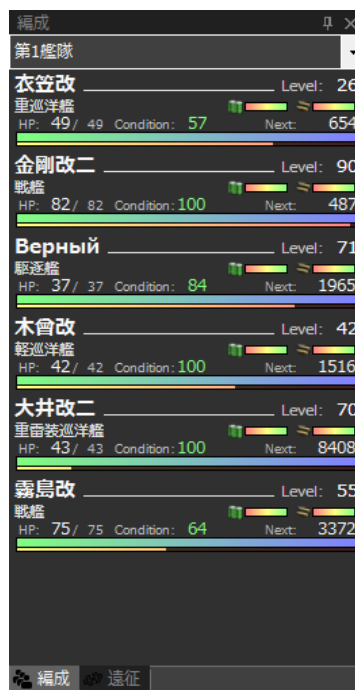


図 5 編成プラグイン

4.12 所持艦プラグイン

所持しているすべての艦のデータをリストにし、値とプログレスバーで表示する。ゲームサーバと所持している艦に関する通信を検出すると自動で内容が更新される。所持艦プラグインの画面を図 6 に示す。



図 6 所持艦プラグイン

4.13 遠征プラグイン

遠征へ出撃している艦隊の遠征完了時間を表示する。遠征に関する通信をゲームサーバとした際に、自動的に内容が更新される。遠征プラグインの画面を図 7 に示す。



図 7 遠征プラグイン

4.14 評価

フレームワークによる専用ブラウザとプラグイン開発用ライブラリを使用することで、プラグインによって様々な機能を追加することができ、柔軟なウィンドウ操作のできる専用ブラウザを開発することができた。また、プラグインは専用ブラウザとインタフェースを通した最低限の繋がりしか持っていないため、単体でのテストが可能でありシステムの拡張性や保守性に優れ、短時間での開発が可能であると考えられる。

5 考察

実験の結果からフレームワークを使用することで専用ブラウザに対し機能を追加できることがわかり、柔軟なウィンドウ操作によりユーザの実行環境やプレイスタイルに合わせた変更が可能であることもわかった。異なる開発者が一つのプラットフォームに対し、各々開発した機能を公開することで、ユーザは欲しい機能を選択することができるようになる。これらのことから本フレームワークは数ある専用ブラウザを統合するための方法として、開発者とユーザの双方にとって有効であると考察する。

また、ゲーム専用ブラウザのためのフレームワークとしてだけでなく、機能をプラグイン化し拡張性を高めることが可能な点やドッキング可能なウィンドウシステムを提供できる本フレームワークは、ゲームのみならず統合開発環境やその他オーサリングツールへの転用も可能と考えられる。

6 おわりに

現状では実際に専用ブラウザを使用してもらい評価実験をするに至っていないため、今後本フレームワークを外部に公開し、開発者やユーザからより多くの改善点や意見をもらい、より高度なフレームワークへと発展させていきたいと考えている。

考慮すべき改善点として、プラグインはインタフェースによる実装の保証はできるが、バグを含む処理や悪意のあるコードを防ぐための機構を備えていないため、セキュリティ面に脆弱性を持っていることがあげられる。今後はセキュリティ面を考慮したより高度なフレームワークとして改善していく予定である。

参考文献

- [1] KanColle_STAFF, Twitter / KanColle_STAFF: 「艦これ」登録者数は遂に 190 万人を突破しました…まさかの大量 ... , Twitter , <https://twitter.com/KanColle_STAFF/status/457680440430579712>, (参照 2014-04-19)
- [2] KanColle_STAFF , Twitter / KanColle_STAFF: @k2lunatic 厳密には規約には抵触してしまうかもです ... , Twitter , <https://twitter.com/KanColle_STAFF/statuses/373964944653312001>, (参照 2014-04-19)
- [3] ゴッドフィールド, <<http://www.godfield.net/>>, (参照 2014-04-19)