

デバッグ制御システム

－ コマンドベースの効率的なデバッグインターフェース作成 －

2014 年 2 月 10 日 初版

板垣 衛

■ 改訂履歴

版	リリース	担当	改訂内容
初版	2014 年 2 月 10 日	板垣 衛	(初版)

■ 目次

■ 概略	1
■ 目的	1
■ 要件定義	1
▼ 基本要件	1
▼ 要求仕様／要件定義	2
■ 仕様概要	4
▼ システム構成図	4
■ データ仕様／処理仕様	4

■ 概略

デバッグ機能の追加実装や、デバッグ情報の扱いを簡単にするためのシステムを設計する。

■ 目的

本書は、本書が示すシステムにより、デバッグ機能を簡単に追加実装できるようにすることを目的とする。

また、PC などと通信して、デバッグ情報をより手軽に扱えるシステムにすることも目的の一つとする。

デバッグ機能とインターフェースの追加を簡単に行えることにより、コンテンツ製作者や QA スタッフにとって便利な機能も迅速に拡充できるようになる。それにより、全体的な開発効率と品質が向上し、プロジェクトの進行をスムーズにすることを最大の目的とする。

■ 要件定義

▼ 基本要件

本書が扱うシステムの基本要件は下記のとおり。

- ・ デバッグメニューからデバッグ機能を実行できるものとする。
- ・ デバッグメニューの項目は、外部データ化して扱えるものとし、分かり易い項目の配置や日本語の表示が柔軟に行えるものとする。
- ・ PC 上のデバッグツールからゲームにネットワーク接続して、専用の GUI でデバッグ操作やデバッグ情報を扱うこともできるものとする。
- ・ 制作スタッフや QA スタッフからの要請に迅速に応じることができるよう、デバッグ機能の追加実装が、きわめて簡単に行えるシステムとする。

▼ 要求仕様／要件定義

以下、本書が扱うシステムの要件を定義する。なお、要件として不確定の要求仕様も併記する。

- ・ デバッグ機能の実行は、ゲーム上のコマンドプロンプトから実行するコマンドとして構成することを基本とする。
 - 例えば、「CPU ON」というコマンドを実行すると、CPU 使用率メーターが画面に表示される。
 - この場合、「CPU」がコマンドで、「ON」がコマンドパラメータである。
- ・ デバッグメニューシステムを用意し、メニュー項目に応じたデバッグコマンドが実行されるシステムとする。
 - メニューにはコマンドのリストが並び、選択して実行することができる。
 - メニュー項目は、コマンドがそのまま表示されるのではなく、見て分かり易い見出し（名前）をつけることができる。
 - 例えば、「CPU 使用率」と書かれたデバッグメニューを選択して決定ボタンを押すと、内部で「CPU ON」コマンドが実行され、CPU 使用率メーターが画面に表示される。
 - メニューの実装が済んでいないコマンドも、コマンドプロンプトから直接実行することができる。
- ・ デバッグメニューシステムのメニュー構成は、外部データ化して定義可能とする。
 - メニューの表示項目（日本語&英語）、メニューに対応した実行コマンド、コマンドパラメータ、サブメニューを扱えるものとする。
 - 同じメニュー項目でも、状況によってコマンドおよびコマンドパラメータが変わるようなものも扱えるものとする。
 - 例えば、CPU 使用率メーターが画面に表示されている状態で「CPU 使用率」メニューを実行すると、「CPU OFF」コマンドが実行されて CPU 使用率メーターが非表示になる。
 - デバッグメニューの定義は JSON 形式（テキスト）で定義する。
 - 別紙の「ゲームデータ仕様」のデータ仕様に準拠し、バイナリデータに変換して扱う。
- ・ PC 上のデバッグツールから、ゲームにネットワーク接続して、リモートでデバッグコマンドを実行することに対応するものとする。
- ・ デバッグコマンドの実行結果は、その実行方法に応じた出力を行うものとする。
 - コマンドによっては特に何も出力されない。
 - 例えば、「CPU」コマンドは、画面上の CPU メーターの表示 ON/OFF が行われるだけで、特に何も出力しない。

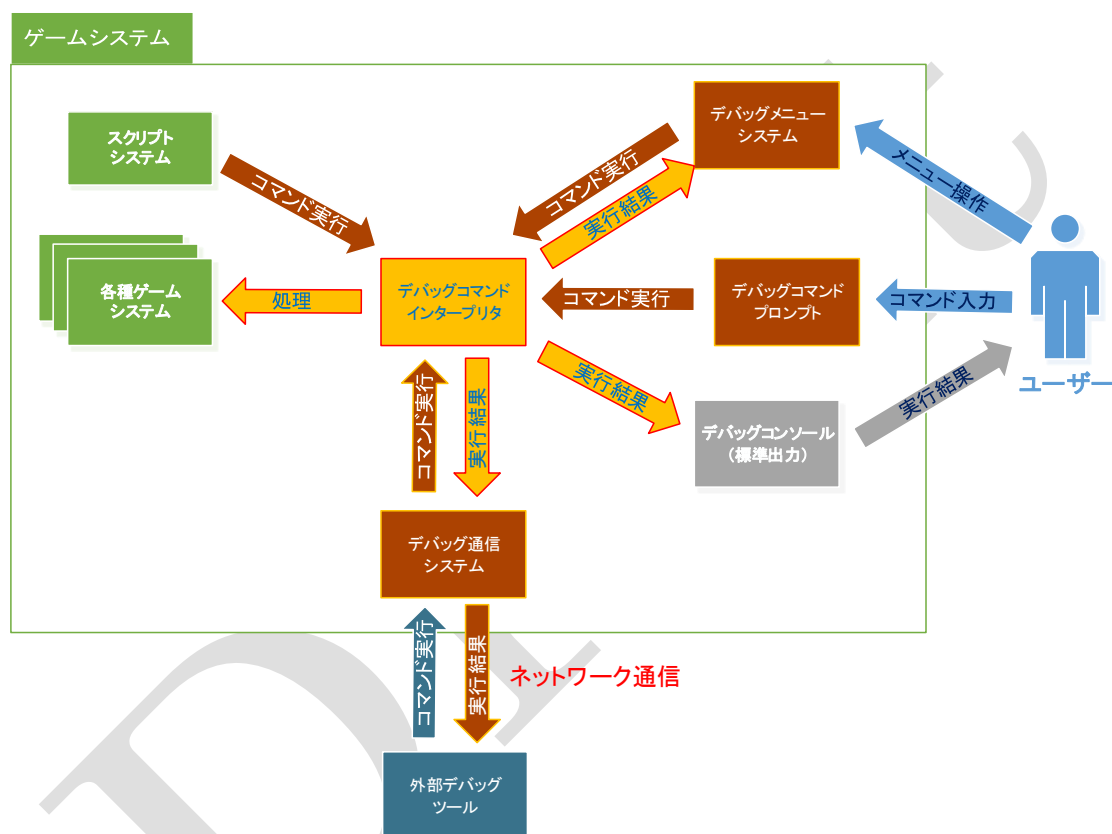
- コマンドプロンプトから実行したコマンドは、標準出力（デバッグコンソール）に出力される。
 - 例えば、「FRATE」コマンドの結果は、標準出力に「30 fps」といった結果が出力されて、内容を確認することができる。
- デバッグメニューから実行したコマンドは、標準出力、もしくは、デバッグメニュー自体に表示される。
 - 例えば、メニュー項目として「フレームレート [%s]」、「メニュー表示用コマンド」として「FRATE」が定義されたメニューには、「フレームレート [30 fp]」という項目が表示される。
- リモートから実行したコマンドは、接続元に実行結果が送信される。
- ・ デバッグ機能（コマンド）を簡単に実装するための中心機能として、デバッグコマンドインタープリタを用意するものとする。
 - デバッグ機能を追加する際は、デバッグコマンドとそれに応じたコールバック関数を登録する。
 - コールバック関数は、main() 関数のように任意の数のパラメータを文字列で受け取れる形式。
 - デバッグコマンドは内部で CRC 値化され、ハッシュテーブル、もしくは、バイナリサーチ可能な構造で記録される。
- ・ ゲーム内の各種スクリプトからもデバッグコマンドを実行可能とする。
 - スクリプト開発を補助するデバッグ機能はデバッグコマンドとして登録して扱う事を基本とする。
- ・ デバッグコマンドはリリースビルド時に実行されても無視される。

■ 仕様概要

▼ システム構成図

要件に基づくシステム構成図を示す。

デバッグシステム関係のシステム構成図：



なお、Release ビルドではデバッグ関係のシステムは全て消滅し、スクリプトシステム音デバッグコマンド実行処理（関数）は、呼び出されても何もしない処理になる。

■ データ仕様／処理仕様

(現狀未定義)

■■■■以上■■■■

■ 索引

索引項目が見つかりません。

デバッグ制御システム

以 上