

Visual Studio Code による RPG ツクール MZ のデバッグ

蒼竜 (@soryu_rpmakermv)*

1 はじめに

Visual Studio Code(以下 VSCode) は、Microsoft が開発したソースコードエディタであり、あらゆる言語に対応している。RPG ツクール MV/MZ のゲーム部分は javascript で書かれているため、設定さえ正しく行えば、VSCode を通して RPG ツクールのゲームをデバッグできる(実質、RPG ツクール本体すら不要になるため何とも言いづらいものであるが…)。

基本的には既存の記事がネット上でいくらでもあるので、仔細は省略するが、個人の環境によっては、ところどころで他所のネット記事通りに進めてもうまくいかない部分がある場合がある。

また、新しい PC 環境へ鞍替えする時に VSCode の構築方法をいちいち調べなおすのも面倒なため、自身の備忘録として、実際に成功したローカル環境 (RPG ツクール本体から起動した場合と同じ挙動をする環境) デバッグの構築方法についてまとめておく。

2 VSCode でツクール製ゲームを起動する

2.1 VSCode 入手

VSCode はフリーソフトであるため、公式サイト¹⁾より各々の制作環境に応じて必要なバージョンをダウンロードし、インストールを行う。

2.2 デバッガの入手

RPG ツクール MV/MZ は、NWjs というウェブアプリケーションのためのフレームワークを基幹として作られている。まずインストールしたての VSCode に NWjs のデバッグを行うための (VSCode の) プラグインを導入する。

図 1 のように VSCode 画面左端のタブの最下部に位置するパズルの組み合わせのようなアイコンを選択する。テキストボックスに適当なフレーズを入れて検索すると、候補となるものが列挙されるので、インストールしたいものを選択する。

Debugger for NWjs を探してインストールする。また、VSCode のエディター UI は英語で表記されているため、先にテキストボックスに japan とでも入力し、日本語化プラグインを導入しておくといよい。



図 1: VSCode の拡張機能 (プラグイン) の検索とダウンロード

2.3 ツクール対応版の導入

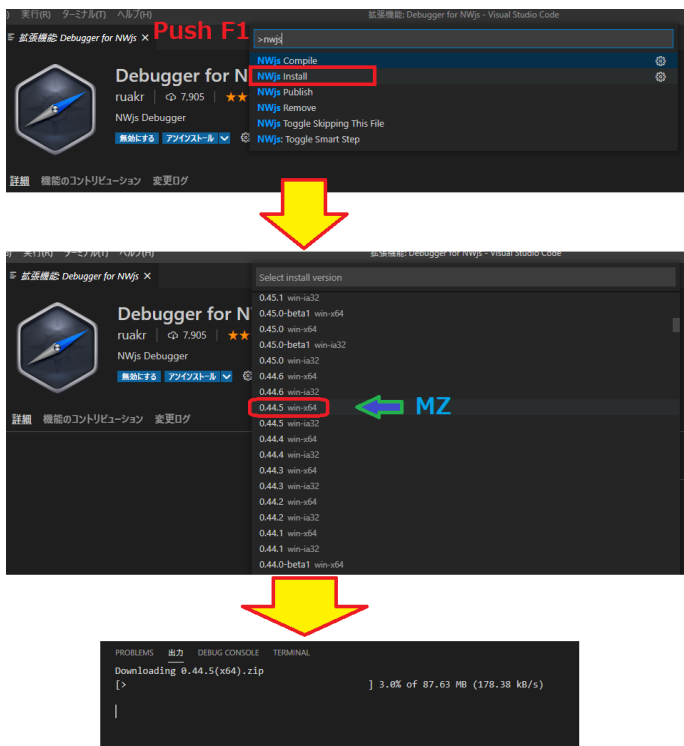
次に先ほど入手した NWjs を、ツクール (MV/MZ) の環境に合わせてインストールする。図 2 のように、F1 キーを押すと上部にコマンドパレットが出現する。

そこへ NWjs と入力すると様々な候補が現れるので、NWjs install を選択する。さらにインストールするバージョンを聞かれるため、RPG ツクール MZ に合わせるように 0.44.5 を選択する。ツクール用の NWjs は存外にもバージョンが低い。

バージョン選択が終わると、ツクールに対応した NWjs デバッガのインストールが始まる。

*<http://dragonflare.blue/dcave/>

¹⁾<https://code.visualstudio.com/download>



ソースコード 1: ローカル起動のための launch.json 記述例

```

1  {
2  "version": "0.2.0",
3  "configurations": [
4  {
5  "type": "nwjs",
6  "request": "launch",
7  "name": "Launch NW.js",
8  "nwjsVersion": "any",
9  "webRoot": "${workspaceFolder}",
10 "reloadAfterAttached": true,
11 "runtimeArgs": ["${workspaceFolder}", "test"]
12 }
13 ]
14 }
    
```

紙面の都合で整形が一部崩れているところはあるが、ゲームプロジェクト最上位 (game.rmmzproject があるフォルダ) を指定してあれば、これを使うことで、RPG ツクールからゲームを起動したときと同じ挙動でデバッグができるだろう。

図 2: VSCode の拡張機能 (プラグイン) の検索とインストール

2.4 ツクールのゲームプロジェクトの対応

ここまで終わると、あとはデバッガーへのゲームプロジェクトの割り当てとなる。これは、デバッグを行いたいゲームプロジェクトごとに行う必要がある。

まず、デバッグを行いたいゲームのプロジェクト (が存在するフォルダ) を VSCode 上でオープンする。もちろん本当にツクールのプロジェクトを開くわけではなく、game.rmmzproject があるフォルダを指定するだけである。

デバッグは F5 キーを押すことで開始できるが、ここにゲームプロジェクトごとの設定が発生する。launch.json というファイルの設定を求められるため、次のソースコード 1 のように記載する。

3 VSCode デバッグの注意点

確かに VSCode を使ったデバッグによって、Visual Studio のように F10/F11 キーを用いたステップ実行を用いたコード 1 行 1 行を追跡する有益な機能が利用できる。その一方で、問題の追及のために頼りすぎてしまうと、RPG ツクールの通常のスクリプティングでは表面には現れない PIXI.js の細部のような非常に抽象的な次元のコードまで追跡を行うことになってしまう。

こういったソフトウェアに日常的に慣れない者が無暗に使うと、本質的なデバッグ作業がかえってやりづらくなる場合もあるので、console.log() 等の javascript の基本的なもので済む場合は、それで済ませることに越したことはない。