

並列処理イベントによるプレイヤー監視とその応用

蒼竜 (@soryu_rpmakermv)*

1 はじめに

RPG ツクールシリーズのイベントのトリガー（開始条件）には、「並列処理」というものがある。ツクールで言う「並列処理」とは、一般のコンピュータ科学で言うそれではなく、「マップの更新やプレイヤーの入力受付などと共に常に繰り返し実行される」という処理のこと（図1）であり、早い話、記述したイベントの内容が「そのマップ上にいる限り、常に繰り返し実行される」と考えればよい。ツクールでの「並列処理」という言葉は比喩的なものであり、実際にパソコン上で並列処理が行われるわけではないため、これをツクールで利用するとゲーム中のCPU使用率が爆発的に上がるという事は起こらない。

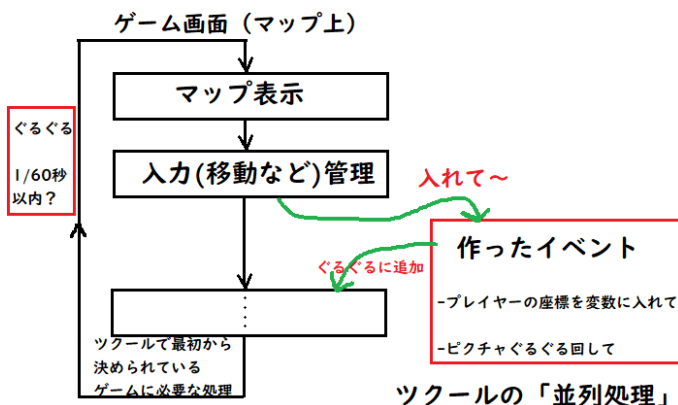


図1: 「並列処理」の概念イメージ

これを用いれば、特にゲーム中のプレイヤーの位置情報を取得することがツクールのイベントコマンドのみで実現可能であり、取得した位置情報を元にイベントを起動したり、処理を分岐させたりということが自在に行える。ただし、前述のようにコンピュータ科学的な並列処理が行われるわけではないため、「並列処理」をトリガーとするイベントがマップ上で大量に同時起動されてしまうと、ゲーム自体の処理が重くなり、プレイヤーにとっては快適性を損なう結果に繋がるため、利用には十分な配慮が必要である。（図で言えば、入れてくれ~と言って

くるツクールユーザーが作った並列処理イベントが大量に入ってくると、ゲームはそれも含めて全体の処理を常に管理しないといけないので、毎回到処理すべき仕事の量が増える）

本稿では、RPG ツクールでの並列処理トリガーを持つイベントの基本的な使用法および応用例を示す。

2 並列処理イベントによるプレイヤー監視

まず、基本事項としてRPG ツクールでの並列処理イベントを用いたプレイヤーの座標情報を取得する方法を述べる。並列処理イベントは、基本的に空のグラフィックのまま好きな（ゲームの邪魔にならない）場所に置いて構わない。著者はプレイヤーがゲーム内で触れることのないよう、壁や川などの侵入不可エリアにしばしば置いている。

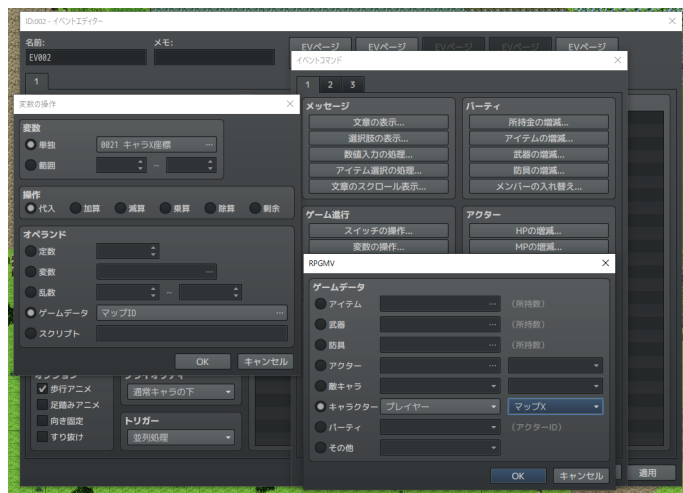


図2: RPG ツクールの機能によるプレイヤー位置取得までのコマンド操作

プレイヤーの座標はX座標,Y座標で管理されているため、そのツクールの内部データを取得するために、予め変数を用意しておく。その変数にツクールの内部データとなっている座標情報を格納する方法は、図2のよ

*<http://dragonflare.dip.jp/dcave/>

うにイベントコマンド「変数の操作」より「ゲームデータ」、「キャラクター」と選択をして、現れたプルダウンメニューの中から X 座標あるいは Y 座標を選択することである。例に挙げた図では、プレイヤーの X 座標を格納する変数を準備して、そこへゲームから取得した X 座標を格納している。

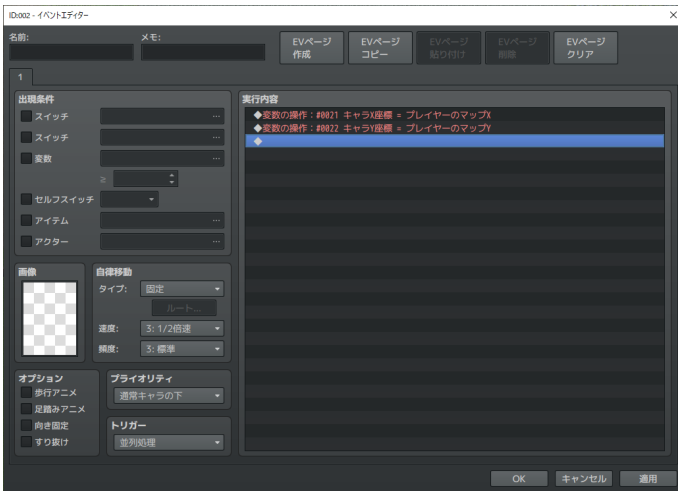


図 3: プレイヤー座標の取得

これを、Y 座標についても同様に行った結果が図 3 のようなイベントになる。



図 4: プレイヤー位置取得結果

このイベントを適当なマップに配置して、ゲームを実行した結果が図 4 になる。ここでは説明のため、KMS_DebugUtil.js¹⁾を使用して左上にプレイヤー座標 (=格納した変数の値) を常に表示している。実際には、プレイヤーキャラクターの移動に対応して値が常に変化する。

¹⁾http://ytomy.sakura.ne.jp/tkool/rpgtech/tech_mv/develop/debug_util.html

この並列処理イベントに処理 (条件分岐) を書き加えて、ゲームのイベント発生に組み込めるようにしていく。次節から、その処理の例を挙げていく。

2.1 指定位置に到達

先の節のように、並列イベントを用いてプレイヤー座標を常時取得している状態であれば、図 5 のようにして、プレイヤー座標に基づく処理を行うことができる。

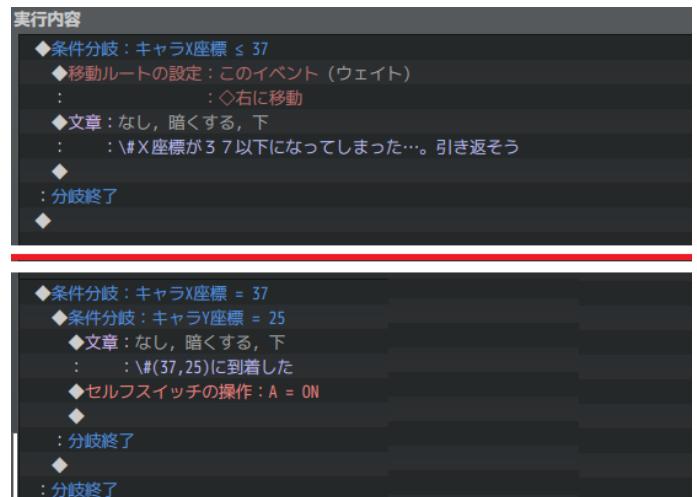


図 5: プレイヤー位置による処理分岐イベント

図の上半分では、プレイヤーの位置が 37 以下になったら指定の処理が行われるイベントになっていて、図の下半分では、プレイヤーが (37,25) という指定の位置に立っていたら処理が進むイベントになっている。これらは、プレイヤー位置座標取得とは別に、並列処理イベントで監視することができる。

下図 6 のようにまだ進んではいけないエリアに立ち入らせないようにする (1 歩引き戻す等して先へ進ませない) ために同じ処理のイベント (トリガーを「プレイヤーから接触」) を大量配置するよりも並列処理で位置情報を監視した方がエディタで保存・読み込みするデータ量を減らすことができる。(この例では、マップ構造がその地点に来るまで x_j38, つまりマップ右下の方から上がってくるようになっていることが前提)

類似な例として、以下の図 7 ようなイベントの書き方 (条件の判定) を行うと「(25,20) から半径 6 セル以内にプレイヤーが進入したら」という円形範囲の位置判定を行うことができる。

「変数の操作」より「ゲームデータ」、「キャラクター」でプレイヤー以外の任意のキャラクターの X 座標、Y 座標も取得できるため、変数をうまく用いれば「任意のイ

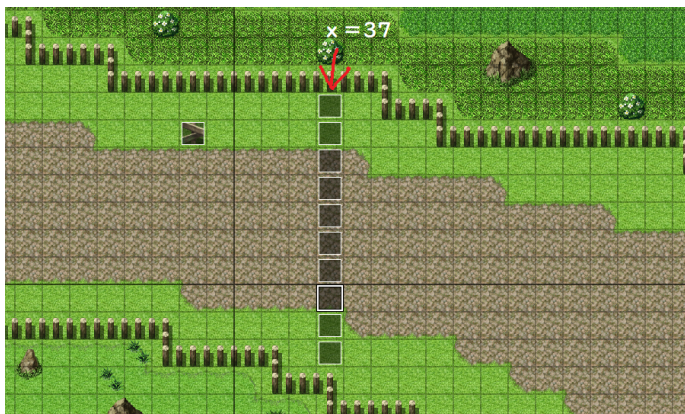


図 6: プレイヤーの指定位置到達判定を行うイベントを愚直に配置した例

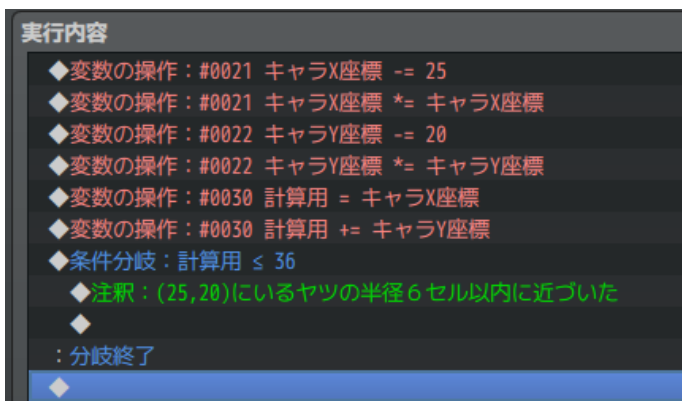


図 7: 円形範囲内のプレイヤー侵入検査イベント

「イベントの半径 N セル以内」に入ったらと一般化できる。イベントとプレイヤーの距離を

$$r = \sqrt{(x_p - x_e)^2 + (y_p - y_e)^2} \quad (1)$$

とすると、 $r < N$ となればイベント開始条件を満たす。式中の添字順不同である。

またツクールで平方根は面倒（少なくともイベントコマンドにはない）なので、2乗値で条件分岐をしている。図中の計算用 ≤ 36 で6セル以内、ということになる。

2.2 任意のボタンを押す

条件分岐に「指定したボタンが押されている」という条件が存在する。もちろんこれはプレイヤー座標取得と関係なく使用できるが、スクリプトの概念がツクールに導入されていなかった～2000/2003時代では、マップ上にプレイヤーやイベントの位置座標取得と合わせて使用し、今ではプラグインで行われているような自作システムの設計でしばしば活躍していた。

2.3 当該マップ内へ進入した瞬間に1度だけ実行

ツクールのエディタのイベントコマンドには「イベントの一時消去」というものがある。これを並列処理のイベントに挿入すれば、「マップにいる間、条件を満たした1回だけ機能する」イベントが作れる。

「自動実行」という同様なトリガーもあるが、こちらはイベントを強制的に割り込ませる処理になるため処理が終了するまでプレイヤーの入力を受け付けなく（イベントの処理に操作権を持って行かれる）なる。

どちらがいいかはゲームの状況によりけりであるが、「自動実行」で不自然に一瞬だけ操作を受け付けないという違和感が生まれることが問題で、プレイヤーに悟られずにスムーズに処理させたい場合は有用と考えられる。

3 応用例

3.1 場所移動

SFC時代のドラゴンクエストシリーズにあったような、「マップの四隅に到達するとフィールドマップへ移動する」という処理はマップ全体の終端に「場所移動」イベントをばら撒かなくとも実現できる。

対象のマップの大きさが $M \times N$ のとき、1つ並列処理イベントを配置して、プレイヤーの X, Y 座標を確認し、四隅 ($X=0, X=N, Y=0, Y=M$) にプレイヤーがいたならば「場所移動」を実行とすれば、エディタ上のマップの見目はかなりスッキリする。このマップの隅全てに移動のためのイベントをばら撒いていると、単純に移動のためにイベントの個数は $2(M + N - 2)$ 個必要であり、これがただ1個の並列イベントに置き換えられるともなればそのマップの json ファイルの容量としては非常にリーズナブルなサイズにまとまる。

3.2 イベントの発生

以前の記事とも繋がるが、プレイヤーが指定位置に到達した時に、ストーリーを進行（次のイベントを起動）させるという処理にも活用できる。

下図の上半分が、イベント起動条件（プレイヤー位置）を並列処理で監視している。発生条件（ここでは Y 座標を見ているが、指定キャラの周りなど何でもよい）を満たしたら、ただちにセルフスイッチを入れて、次ページのイベント（下図の下半分）に移行している。下半分

では、画面のフェードアウトを含むイベントの開始準備処理となる。

ここをページで独立させているのは、この部分が並列処理で処理されてしまうと画面のフェードアウト中でもまだプレイヤーが自由にマップを移動できてしまうことから、演出面で不自然になってしまうからである。

ストーリーの進行準備ができれば、次のセルフスイッチを使ってさらに次のページのイベントを発生させ、実際のストーリーを展開する処理に移る。

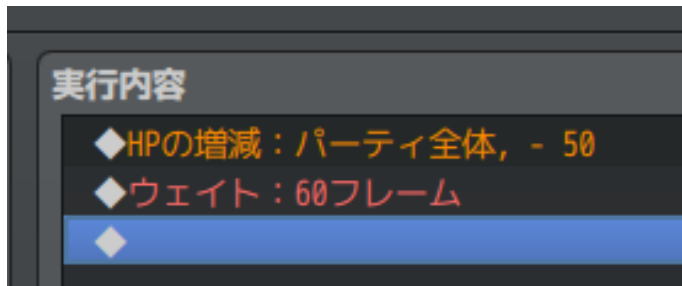


図 9: 並列処理によるスリップダメージ演出



図 8: 指定範囲内にプレイヤーが進入するとストーリー演出シーンが起動するイベント設計

火山のような暑い場所でじわじわと時間に比例して体力を奪われる演出など、ダメージ床とは異なるマップの演出方法を確立できる。ここに、条件分岐で装備やステータスを確認する処理を追加すれば、対策・防止策についても実装が可能である。先ほどの円形範囲の判定等も組み合わせれば、「ダメージを受ける毒ガス地帯」も表現可能である。

3.3 特定マップでの継続ダメージ

最後に、並列処理における「ウェイト」についても触れる。通常のイベントで「ウェイト」の処理を入れると指定したフレーム時間の間、処理を停止させることができる。

並列処理をトリガーにしたイベントにウェイトを入れると、そのフレーム時間の間 その並列処理イベントだけを止めることができる。

(最初に示したツクールにおける並列処理イベントのフローと直感的に一致しないことになるが、「指定のウェイトがかかっている間、その並列処理イベントの処理だけが飛ばされる」と考えればよいだろう)

合わせ技のような使い方になるが、例えば下図のようなことができる。

これは、この並列処理イベントを置いたマップにプレイヤーがいる間、1秒おきに50のダメージがパーティー全員に発生するというものである(当然、パーティーメニューを開いている間は処理が止まっているのでHPは減らない)。