

TurboWarpの有能な拡張機能を独断と偏見に基づいて紹介する記事

・はじめに

- ①記事を書くのは初めてなので、まあ色々大目に見てください
 - ②TurboWarpの拡張機能で使ったことが無いものも多いです
 - ③判断基準はドブに捨てました
 - ④語彙力、ないですかね...
- これらのクソガバ要素を受け入れられる方はぜひ読んでいってください

・そもそもTurboWarpってなに？

[TurboWarp](#)とは、Scratchというプログラミング言語をJavaScriptにコンパイルし、プロジェクトをより高速に、そして新たな拡張機能を導入したり個人で詳細な設定をいじれるようにしたScratchのMODで、GarboMuffin氏によって開発、維持されています。
(ここにおけるコンパイルというのは、おそらくScratchで書かれたプログラムをJavaScriptのプログラムにすることだと思う)



←TurboWarpのアイコン

今回はTurboWarpの[拡張機能](#)のうち、私がよく使っているものを4つ取り上げて、使い方と良い点や悪い点、そして総評をして紹介していきます。
これらはScratchとの互換性が無いため、Scratchでは使うことができないことには注意してください。(0敗)
そもそもこの記事を読んでいる人の中でどれだけの人がTurboWarpを使ってるんだ？

・1.アニメーションテキスト



この拡張機能では、スプライトをテキストに変えることができるブロックが追加されます。
また、それらのテキストのフォントを変えたり、アニメーションをつけることができます。

・使い方

ブロックを実行するとスプライトが入力したテキストに置き換わる
その際、スプライトの大きさや座標なども引き継ぐ
クローン事に別で動かすこともできる
(他にもいろいろな機能があるけど割愛します、まあ多少はね?)

・良い点

- ①Scratchにはないフォントも追加できる
- ②文字の大きさや寄せる方向なども設定できる
- ③独自のアニメーションをブロックだけで実行できるので、わざわざ作らなくていい(怠惰)
- ④日本語対応
- ⑤バグは(多分)無い

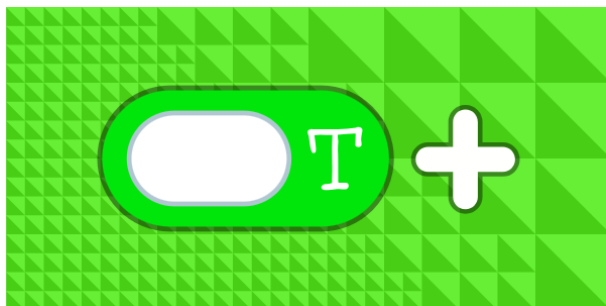
・悪い点

- ①スプライトをテキストに置き換えるため、独立した文章を表示するのならそのために個別でクローンを用意する必要がある
- ②スプライトをテキストに置き換えるため、もとのスプライトは置き換えられてしまう(ただし、これらは当たり前というか仕様なので、実際デメリットは少ない)

・総評

有能で、導入する分に特にデメリットはないし、ブロックが複数必要なものを使い回すのならブロック定義を活用すればいいので、UIとかは全部こいつに頼っていいと思います。

・2.テキスト



(さっきのアニメーションテキストと名前が似てるけど、機能は全く違うよ)

・使い方

このブロックでは、主にテキストを操作して、答えが出力されます。例えば、

- ・lettes[1]to[2]of[3]: テキスト[3]から、[1]文字目から[2]文字目までを出力する
(例: lettes 2 to 4 of apple→ppl)
- ・item[1]of[2]split by[3]: テキスト[2]から、[3]に仕切られた[1]番目を出力する
(例: item 2 of banana split by a→n)
- ・count[1]in[2]: テキスト[2]に[1]がいくら入っているかカウントし、出力する
(例: count p in apple→2)

などなど。(んにゃび...よくわかんなかったです...)

・良い点

- ①Scratchにはこのようにテキストを操作するブロックがないので、このようなことをするならブロック定義やリストを駆使しないと難しいが、この拡張機能ではそれが一つのブロックで解決する
- ②テキストから何かを取り出すだけでなく、unicodeに変換できたり、大文字や小文字などの判定もできる(Scratchだとこのようなことはたぶんできない)
- ③バグは(多分)無い

・悪い点

- ①日本語には対応していないので、分かりづらい部分があるかも...?
(でも、デメリットはほぼ無い)

・総評

(導入してもデメリットはほぼ)ないです。(日本語対応してないくらい)
キングオブ有能。煮るなり焼くなり好きに使いましょ。
個人的にはコマンド(?)を入力して実行するタイプのブロック定義(語彙力)が
作りたい人におすすめです。

・3.一時変数



この拡張機能では、ランタイム変数とスレッド変数という二種類の変数を使い捨てで作成できるようになります。[サンプルプロジェクト](#)も参考にしてみてください。

・使い方

まず、ランタイム変数とスレッド変数の違いについて簡単に説明すると、
ランタイム変数は使用されているスクリプトがすべて終了した際に消去され、
スクリプトに関係なく使用できるのに対し、スレッド変数は一つのスクリプトでしか使用できず、またそのスクリプトが終了した場合消去されます。

要は、ランタイム変数は全体で使えるけど、スレッド変数は一つのスクリプトでしか使えないって
感じで覚えておけば使えると思います。

ほかの仕様は普通の変数と同じです。たぶん。

・良い点

- ①いちいちたくさん変数を作る必要が無くなる
- ②他の仕様は普通の変数とほぼ変わらないので使いやすい
- ③日本語対応
- ④バグは(多分)無い

・悪い点

- ①現在動作しているスレッド変数やランタイム変数の名前はわかるものの、それらの数値を通常の変数と同じように表示することはできない

・総評

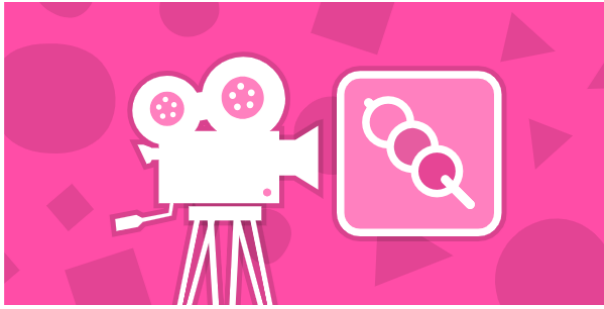
やはりこいつも導入しておいて特に問題ない拡張機能であり、普通の変数と仕様がほぼ変わらないため、

かなり使いやすいですが、難点として、その変数の数値が今どうなっているのか通常の変数と同じように表示することができないため、そこは工夫が必要でしょう。

(例えば、[[ランタイム変数]と言う]を繰り返すとか)

もちろん、そこを差し引いてもほとんど問題なく使える有能な拡張機能なので、おすすめです。

・4.カメラコントロール



この拡張機能では、プロジェクトのカメラを操作することができるブロックが追加されます。ただし、バグが比較的多く、動作が不安定になる可能性があるため、導入する際は判断が必要です。

・使い方

この拡張機能で主に操作できるのは、カメラのx座標、カメラのy座標、カメラの向き(角度)、カメラズーム、そして背景の色です。

追加されるブロックはこの要素に基づいて、スプライトをセンターに持っていったり、角度を変更してカメラを傾けたり、背景の色を変えることができたりします。

・良い点

- ①ほとんどのブロックはScratchの「動き」ブロックと同じように扱うことができるのでわかりやすい
- ②いちいちスプライト一つ一つを動かしてカメラモーションを作成するよりも手間がかからないし軽い
- ③日本語対応

・悪い点

- ①他の保管機能と比べるとバグがとても多く、動作が不安定になる可能性がある

・総評

有能で使いやすいと僕は思っているのですが、どうやらバグが多らしく、動作が不安定になる可能性があるらしいので、導入する際は判断が必要だと思われます。

ただ、僕は今制作している作品にこの拡張機能を導入していますが、バグだと思えるような事象は

今は起きていません。(気づいていないだけかもしれませんが)

おわりに

この記事が最後まで読んでくださり、ありがとうございました。

この記事が少しでもあなたがTurboWarpで拡張機能を導入する際のお役に立てたなら幸いです。

また、納期に間に合わなかったことを深くお詫び申し上げます。許してください、何でもしますからそれでは、良きTurboWarpライフを！