

Python3用

移植版

PAC-WORLD

ABURI6800



◆はじめに

マイコンBASICマガジン(電波新聞社刊)の1986年3月号に掲載された、PC-8001(N-BASIC)用の「PAC-WORLD」をPythonに移植してみました。

◆遊び方

タイトルが出たら、「z」キーでスタート。
カーソルキーでスピードを調整しつつ、「z」キーで穴に落ちないようにジャンプして、ゴールを目指してください。
ジャンプ台に乗ってジャンプするとロングジャンプ!「\$」を取るとスコアアップします。
全部で4面あります。

◆実行環境について

Python3.8.2で書きました。
tkinter、Pillowを主に使っています。
Windows、Mac、Android(Pydroid3)での動作確認をしています。(多分、LinuxのX Windowでも大丈夫だと思います)

◆N-BASICからの移植について

まず、表示系の処理ですが、PC-8001の文字フォントを全て画像で用意しました。
これを文字コードに対応した配列に入れ、表示する処理では文字列または文字コードの配列を元に、指定された座標にpasteしています。

また、ImageTk.PhotoImageはロード後に加工できない、知らないうちにガベージコレクションで破棄される、など使い方にちょっとクセがあるので、画面描画処理では毎回画面のimageを作り、PhotoImageに渡しています。

なので、画面のimageを作ってから、その後にフィルタ加工などをして表示することもできます。(今回はやっていません)

プログラム本体については、BASICのプログラムをベタで移植するのではなく、ゲーム処理と画面描画処理に分けて記述しました。

ゲームの状態をgameStatusで、その画面の経過時間をgameTimeで管理して、各状態に合わせてゲームの処理や画面を描く関数を呼び出しています。

そのため、Pygame等に移植するときも、画面周りの処理を変えるだけで済むようになっています。

なお、このゲームは結構大きめのキャラクタが動きますが、さすがに毎回描いていると無駄な気がしたので、予め画像ファイルとして作ったものをロードしています。ゲーム画面でも、

固定で表示されるものは画像ファイルに作っておいたほうが良かったですね。

音は原作はBEEP音のみなのですが、OSに依存しない使いやすいようなライブラリが見つからなかったので、残念ながら再現できていません。

◆苦労したところ

画面の表示がうまく行ったので楽勝かな？と思ったのですが、なるべくそのままのゲーム性となるように、原作と同じタイミングで判定などの処理をするのに苦労しました。

また、マップデータはリストに持っていますが、Pythonで特定の1つの要素を取り出してもリスト型になっていることに気づかず、少しハマりました。(考えてみたら分かるんですけどね…)

◆最後に

Pythonでもtkinterだとスムーズな動きは難しいですが、昔の8ビットマイコンのBASICゲームは逆に良い感じで再現できる気がしますね。

OSを選ばずに動かせるのも良いですね。

せっかくPC-8001のキャラクタセットを作ったし、移植のコツも見えてきたので、あと何本か作ってみようかと思います。(MZもそのうち…)

あ、そうそう、実はコンティニューもできます。
タイトル画面で、あるキーを押すと…(って、プログラム見たらすぐわかりますね)

最後に、素晴らしいゲームを作られた原作の作者である松島 徹氏に、この場を借りてお礼申し上げます。