

例題 2 プログラムを拡張しながら基本を復習

爆弾ゲームで Ruby くんから弾を撃ち返そう！

今回の目的とプログラミングの内容

- (1) 昨年つくった「爆弾ゲーム(9-3)」をもとにして、Ruby くんから弾を撃ち返せるようにする
 - ・爆弾ゲームの動作確認
 - ・弾クラスを作成する
 - ・弾クラスを爆弾ゲームに組み込んでみる
 - ・弾クラスを使ってたくさんの弾を撃ってみる
 - ・弾と爆弾の当たり判定をする
- (2) 技術的な要素として「クラス」と「配列」をつかうので、その使い方もできるだけ把握する
- (3) 上記の(1)(2)についての関連事項についても適宜復習する

[完成予想図]



例題 2 プログラムを拡張しながら基本を復習

プログラミングの手順

(1) 爆弾ゲーム(9-3)

まずは、昨年つくった「爆弾ゲーム」を動かしてください。

まだ「9-3 (第 9 章 9.3 の状態)」までできあがっていない場合は、「9-3」のプログラムと画像一式をサーバからコピーして動かしてみてください。

すでに「9-3」より先に進んでいる場合や、独自の拡張や変更などを行っている場合は、そのオリジナル版をもとに今回の機能追加をするので、プログラムが正しく動くかどうか確認してください。

(2) 弾クラスを作成する (tama.rb)

ライブコーディング

- tama.rb ファイルの作成(空ファイルとして名前を付けて保存)
- class Tama ~ end
- if \$0 == _FILE_ ~ end
- 動作確認(引数の値を変更したり)

(3) 弾クラスを爆弾ゲームに組み込んでみる (main_tama1.rb)

ライブコーディング

- main.rb ファイルを main_tama1.rb としてコピー
- require_relative
- Tama.new, tama.move, tama.draw

(4) 弾クラスを使ってたくさんの弾を撃ってみる (main_tama2.rb)

ライブコーディング

- main_tama1.rb ファイルを main_tama2.rb としてコピー
-
- tamas <<
- tamas.each ~ end

(5) 弾と爆弾の当たり判定をする (main_tama3.rb)

ライブコーディング

- main_tama2.rb ファイルを main_tama3.rb としてコピー
- hit?
- 後処理