

nanamd / ProjExDPublic

<> CodeIssuesPull requestsActionsProjectsWikiSecurityInsights

main

ProjExD / day7 / README.md

nanamd 今回の提出物完成形History

1 contributor

53 lines (45 sloc)3.38 KB

第6回

迷路 (ex6/maze.py)

作成者 三島 東 遠藤 久保 丸山

ゲームの概要

- ex13/maze.pyを実行すると、1500x900に迷路が描写され、迷路に沿ってを スタートからゴールまで移動させるゲーム
- 実行するたびに迷路の構造は変化する
- 穴掘り法という手法で迷路を作成(穴掘り法の説明はメモ)

操作方法

- 矢印キーで上下左右に移動する
- 通路を辿ってゴールを目指す

追加機能

- スタートとゴールの設定
- 答えの表示
- ゴールすると"ゲームクリア"と表示する
- 自分の通った道に色が付く #三島
- タイトルの説明 #久保
- 時間の表示 #遠藤
- ボタンの作成と配置 #丸山

https://github.com/nanamd/ProjExD/blob/main/day7/README.md

1/3

ToDo(実装しようと思ったけど時間が無かった)

- 画像の挿入 #丸山
- コンティニューしたらもう一度迷路生成してゲームを始める #丸山 久保
- コンティニュークリアしてコンティニューしたら迷路がさらに細くなる /迷路のサイズ設定で数値を大きくする #三島
- 壁の貫通 #久保

メモ

穴掘り法

- 最初全てのマスに壁を設定し、そこからランダムに穴を掘ることで「通路」を作成していく手法

ゲーム表示の流れ

- ゲームの起動
- ゲームのプレイ
- 迷路の解答表示

迷路ゲームの起動

- 壁と穴の設定
- スタートとゴールの設定
- 迷路を表示

実装の手順

- コードの説明（実装手順）
- まずそれぞれの色や壁、ウィンドウのサイズを設定をしました。
- `classMaze()`でゲームの起動をしたときに渡されるものを設定しました。その時に値をNoneに設定することで後に数値が入り、動くように設定しました。
- `def createMaze()`で迷路のもとになる穴掘り法による道や壁、スタートやゴールを表示するための設定をしました。
- `def setStart`や、`def setGoal`でスタートとゴールの位置の設定をします。ここでランダム関数を使うことで、迷路が生成された時ランダムな位置にスタートとゴールを表示できるようになります。
- `def dig()`で穴掘り法の設定を行います。穴掘り法の原理は後に紹介します。`def change_color`で、`i`、`j`という変数に位置が返され何処が壁で道か、現在地かスタートかゴールかを判断し、対応するところに一番最初に設定した色が返されるようになっています。
- `def createWidgets`で今まで設定したものを画面に表示できるようにし、`def resolve_maze`で自分が移動できるように設定します。ここで壁には進めないように、そして自分が何処を進ん

だかを記憶します。

- def playでゲームを開始した時、現在地をスタートにするとこと、上下左右の動きを設定し、def updateで自分の状態を迷路に反映させます。ここで、自分がどこに動いたか、何処に動けるのかを設定しました。
- def game_clearで自分がゴールに到着した後の設定を行いました。ここに、追加した時間を計測するコードを入れることで、自分がゴールにたどり着くまでにどれだけかかったの時間を表示させます。