Spec av P-uppgift: Dalek

Skriven av: Samppa Raittila 970206-9536

Beskrivning:
Programmet är ett spel inspiretat av Doctor Who. Användaren styr Doktorn och Daleks försöker komma närmare.

Skärm

Skärm 1: introduktion och filhantering

```
....................................
....................................
*****
*     *   **   *      ****** *    *
*     *  *  *  *      *      *   *
*     * *    * *      *****  ****
*     * ****** *      *      *  *
*     * *    * *      *      *   *
*****   *    * ****** ****** *    *
....................................
....................................


Don't let Daleks get too near professor.
You can destroy Daleks by chrashing them together.
Move professor using 'w','a','s' and 'd'.
You can use Sonic Screwdriver by command 'ss'.

Give the name of the game map file which you want to play
>>>
```

Skärm 2: spel [D = Dotktorn, A = Dalek, *=vägg, .=golv]

```
************************************************************
*........A.................................................*
*................................***********...............*
*................................*...............A.........*
*................................*.........................*
*.....**********.................***********...............*
*..............*........D..................................*
*..........*****..........................A................*
*......................***.................................*
*.........A............***.........................A.......*
*....................................*****.................*
*....................................*.....................*
*.........................A..........*.....................*
*..........................................................*
************************************************************
>>>
```

Algoritm

1. Användaren ger filnamnet och programmet läser labyrinten eller hanterar fel
Loop:
2. Programmet printar ut labyrinten med Doktorn och Daleks.

3. Användaren tar ett steg enligt styr-kommandon från användaren eller teleportar sig till en slupmässig ruta.
4. Daleks tar ett steg mot Doktorn så rakt som möjligt. Om Dalek försöker gå rakt in i vägg, står det stilla men om Dalek ska snett in i väggen kommer det att följa väggen.
5. Kraschar två eller flera Daleks med varandra, ligger dom kvar och får symbol #
6. Krockar någon Dalek med Doktorn eller har alla Daleks kraschat så är spelet över
    → break
7. Programmet printar ut resultat av spelet

Minne

Programmet hämtar olika labyrinter från filer men programmet kommer inte att ändra dem. Doktorn och Daleks kommer att vara GameCharacter objekt som vet sina koordinater och om dom lever eller inte. Klasset DalekGame har ett attribut som innehåller labyrinten och GameCharacter objekt som matris. Det updateras enligt information från GameCharacter objekt.

Programskelett

```
'''DalekGame contains all the logic of Dalek game. It is responsible of
updating the gameboard after commands from user interface and can tell if
game is over and who won the game. It can take commands from user
interface and it is responsible to check that commands can be done
obeying the rules of the game. It can return updated game board to other
classes '''

class DalekGame:
    def __init__(self):
        gameboard: matrix containing labyrinth and characters
        doctor: GameCharacter object
        daleks: array containing GameCharacter objects
        gameIsOngoing: boolean variable to tell if game has ended
        winner: variable to tell who won the game

    def setGameboardProfessorAndDaleksFromFile(self, filename):
        '''Function reads labyrinth and initial state of the game from
            file. It gives this information to gameboard variable
            and set coordinates for daleks and professor'''

    def getGameboard:
        '''Returns gameboard variable'''

    def makeStepInGameToCoordinates(self, dx, dy):
        '''This is the main function that user interface should use to
            move doctor to coordinates giving relative distances as
            attributes. For example one step upwards would be
            makeStepInGameToCoordinates(0,1). Function updates the
            coordinates for GameCharacter objects according to game
            rules and updates the gameboard variable. '''

    def sonicScrewdriver(self):
        '''Function places doctor to random place in the board
            and updates the gameboard variable'''

    def getGameIsOngoing(self):
        '''Returns True if doctor and at least one dalek is alive,
            False otherwise'''

    def getWinner(self)
        '''returns 1 if player won the game and -1 if player lost'''
```

```
      def __moveCharacter(self, dx, dy, character):
            '''Moves character to desired direction if there is no wall'''

      def __updateGameboard(self):
            '''Updates the gameboard variable according to the coordinates
            of the GameCharacter objects in the board. It also checks if
            some of the characters are in same coordinates and kills
            these'''

      def __moveDaleks(self):
            '''Function goes through array of daleks and tries to move
            them one step closer to doctor.

'''GameCharacter class repsesents character that knows its own
coordinates and if its living or not. Objects can be classified by
string'''

class GameCharacter:
      def __init__(self, coordinates, char):
            living: boolean variable to tell if character lives
            coordinates: coordinates of character
            char: Type of the character for example "A"

      def getCoordinates(self):

      def setCoordinates(self):
            '''Sets new coordinates if living==True'''
      def getChar(self):

      def setChar(self):

      def isLiving(self):
            '''Returns True if character is alive, False if not'''

      def die(self):
            '''Chances 'living' to False'''


'''DalekCLI class contains all the methods needed to play Dalek in cli.
It has DalekGame object as attribute'''

class DalekCLI:
      def __init__(self, dalekGame):
            dalekGame: DalekGame object

      def startGame(self):
            '''Method introduces the game and asks user to
            type the file name for labyrinth. Calls dalekGame to
            initialize the board. Starts game loop after that'''

      def __gameloop(self):
            '''Uses while-loop and asks user to type commands how to move
                  doctor. Breaks out of the loop when dalekGame tells that
                  is over'''

      def __printGameboard
            '''Clears screen and prints updated game board to terminal'''
```

Arbetsplan

Uppgift	**Deadline**
Labyrint från fil utan daleks och koordinater för doktorn	30.10
Styr kommandon så att doktorn går inte genom väggar	2.11
Funktion för att räkna Daleks riktningar	4.11
updateGameboard funktion/prototyp redovisning	6.11
Krav för godkänd P-uppgift	10.11
Extrauppgift C	15.11
Extrauppgift B, noggrannare plan för Extrauppgift A om det finns tid	25.11