Arantxa Viladomat
Lorena González
Maya Alba

Aladdin's Game

How to start using the game?

To start playing with Aladdin's game you will need to modify the newInput.txt and the world.txt

- The newInput.txt:
  - This file includes the code that Aladdin and his clones (if exist) will execute. The main structure is the following:

```
class program{
        program(){
        }
}
```

  All the Aladdin's clones must have a function where their instructions are declared, and all the instructions for a clone or for the main Aladdin must end with a "turnoff()" function.

  WARNING: The error of missing the turnoff function is not validated by the program and will end in a failure state if skipped.

- The world.txt:
  - This file describes the world where Aladdin and the clones will interact:
    - The first row must contain the "number of rows" of the world
    - The second row must contain the "number of columns" of the world
    - In the next line all the world must be added, to indicate something on a cell do the following:
      - **Empty cell:** 0
      - **Beeper:** Integer number (1 for one beeper, 2 for two and so on) (Represented by a diamond.
      - **Wall:** Use an "x" (Walls are represented by Aladdin enemy)
      - **Aladdin:** Use "r" if Aladdin is facing right, "l" if is facing left, "u" if is facing up, or "d" if is facing down.

  You should only add 1 Aladdin when declaring the world; otherwise, the program will throw an error message

  - This is an example:

```
7
7
0000020
0010000
r000000
xxxx0000
0010000
00000xx
000000x
```

Arantxa Viladomat
Lorena González
Maya Alba

Modules of the program

main.py

The main.py module is the executable module to let the magic begin.
This method is in charge to obtain the initial tokens

lexicographical.py

This module contains the lexicographical functionality. It reads from the inputnew.txt file to gather all the valid tokens, fails if an invalid token is read. Also writes the output (list of tokens) to the output.txt file

rec_desc_karel.py

This module performs the work of the descendant recursive program reviewing the semantic and generating the intercode array that will be executed. It order to achieve it, it uses the meta_rd_karel module

errors_karel.py

This module handles the semantic errors by printing to console the detected errors and stopping the execution, this module is called by the descendant recursive of Karel

meta_rd_karel.py

This module has all the necessary methods to handle Aladdin's semantic

executeInterCode.py

This module receives the intercode and execute the sequence of instructions in the proper order for each Aladdin instance

worldKarel.py

This module contains all the functions related with the world, such as creation of the world, updates, and interaction between all the elements in the world

gui.py

This module is in charge of displaying the current state of the world

vm.py

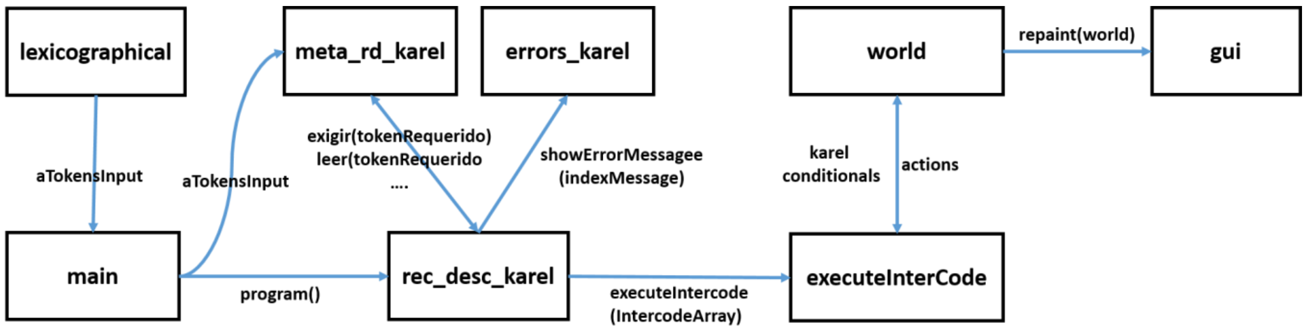
This module acts like a virtual machine, and it's an alternative to the main module, so instead of executing main.py, you can execute vm (also receiving world and newinput files), additionally, to use this module you should have already the intercode of the program to be executed in the intercodeFile.txt file

The intercodeFile should look like this:

```
['JMP', 24, 'TURN_LEFT', 'TURN_LEFT', 'TURN_LEFT', 'RET', 'IF', 'NEXT_TO_A_BEEPER',
'JMP', 13, 'PICK_BEEPER', 'JMP', 6, 'MOVE', 'IF', 'ANY_BEEPERS_IN_BEEPER_BAG', 'JMP', 22,
'PUT_BEEPER', 'MOVE', 'JMP', 14, 'TURN_OFF', 'RET', 'MOVE', 'CLONE', 'CALL', 6, 'MOVE',
'MOVE', 'MOVE', 'CALL', 2, 'MOVE', 'PICK_BEEPER', 'MOVE', 'TURN_LEFT', 'TURN_LEFT',
'TURN_LEFT', 'TURN_LEFT', 'TURN_LEFT', 'TURN_LEFT', 'TURN_LEFT', 'TURN_LEFT',
'TURN_OFF']
```

Arantxa Viladomat
Lorena González
Maya Alba

Design diagram



Opportunities for future improvements

Improvements
A more paused animation, so the user can keep the pace of each action
Having the high level code visible in the GUI, highlighting which instruction is being executed
Having a cleaner design, since some functionalities where built on top of others due to the lack of time
When the user does not type ‘turnoff()’ at the end of the execution of each Karel the program should show an error