International
Institute of Information
Technology Bangalore
ज्ञानमुत्तमम्

Creating a Visual Programming Language like Scratch

9th May 2022

Group 24:

IMT2019088 Tarun Reddy

IMT2019066 Pratyush Upadhyay

IMT2019084 Pratik Ahirrao

IMT2019084 Shrey Tripathi

Problem Statement

- **Aim**: To create a Visual Programming Language like Scratch
- **End Goals:**
  - Understood how a visual programming language like Scratch works internally
  - Got a high-level overview on how to implement basic features like *if-else* conditions, *for/while loops,* variable *CRUD* operations, etc.
  - Implemented these basic features using the Python GUI framework **PyQt5**

Features Implemented

The features we implemented are

- Looping
- Conditional looping
- If conditions
- Scoping
- Arithmetic operations
- Variable CRUD
- Basic movement and rotation

Looping

- We implemented loops using stacks: both the *for* and *while* loops
- On encountering the loop start, we push the *true/false* condition to the stack, and pop it once we encounter the corresponding *end* statement
- We use pointers to jump to a location in the code
- *while* loop ends with *end while*
- Normal *Loop* ends with *End Loop*
- A special stack is maintained for conditional loops
- In normal loops, we add the loop blocks to the stack the number of times we specify in the loop

If conditions

- We implemented if-else conditions using stacks
- Implementation similar to loops
- *If* ends with *end if*
- Instead of repeating code multiple times, we just jump to the relevant block using pointers

Scoping

- We implemented *scoping* using stacks
- We implemented *dynamic scoping* where a global identifier refers to the identifier associated with the most recent environment

Arithmetic Operators

- Add/Subtract/Multiply/Division
- Operations are performed on *Variables* and output is shown on the console

Variable CRUD

- We implemented basic CRUD operations on Variables
- A separate class denotes a variable, which enables access/modification of a global dictionary that stores all variables

Basic movement and rotation

- Implemented the *move* and *rotate* functionalities that perform the corresponding actions on an output image in the output window
- Movement can be in both the *x*- and *y*- directions, based on whether the input provided is *+ve or -ve*

Design Choices

- We used Python because our group was comfortable with Python
- We used PyQt5 because it is a robust GUI framework with increased support for complex GUI applications
- A stack based approach is needed for *if-else*, *loops* and *scoping* because it is a suitable data structure; we pop from the stack on encounter of end of loop/if/end-of-scope

Challenges faced

- Lack of proper documentation for our implementation-specific features PyQt5
- Difficulties were faced while implementing stack-based features like scoping, conditionals and nested conditionals
- GUI implementation wasn’t straightforward in Python, hence consuming a lot of our time

THANK YOU

ज्ञानमुत्तमम्

International Institute of Information Technology Bangalore

26/C, Electronics City, Hosur Road,
Bengaluru – 560 100, Karnataka, India

www.iiitb.ac.in

https://www.facebook.com/IIITBofficial/

https://www.linkedin.com/school/iiit-bangalore/

https://www.instagram.com/iiitb_official/

https://twitter.com/IIITB_official

https://www.youtube.com/user/iiitbmedia