

IMAGI- Child Friendly Programming Language

Project Requirements and Main Features

Prepared by:

Héctor García (hector.garcia@upr.edu) Edgardo Rivera (edgardo.rivera@upr.edu) Jariel Laureano (jariel.laureano@upr.edu)

Introduction

Project Motivation

With the technology industry developing at a rapid pace, the need for more programmers increases everyday. Little options exist to introduce programming to kids successfully, this is why our team decided to attack this need. By creating a new programming language focused and designed for children we will be able to encourage and motivate students at a young age to pursue STEM related careers but specifically, computer and software engineering majors. After experiencing IMAGI, kids will be able to understand the basics of programming, learn to have fun with it and be prepared to move to a more standard programming language like python or java in the future.

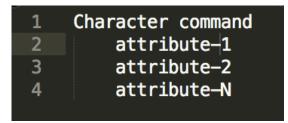
Features

Project Concept

IMAGI is a child friendly programming language developed with the goal of introducing programming to kids in a new and interesting way from a very young age. Lines of code alone will surely have a hard time arousing any children at such a young age, this is the reason why IMAGI's programming experience will be aided by a simple graphical user interface which will allow the programmer to interact with different characters and scenes, and create simple storylines by writing code. One of the challenges of kids as our scope is that this language must try to develop the programming instincts in the simplest way possible, avoiding complicated syntax, overcrowding of commands and ambiguity. Using commonly known words and making the programming language straightforward, as easy as talking, will be the focus. By doing this we will make sure the children understand the commands without putting too much thought into it, keeping them comfortable and confident while programming.

Example Program

In general, the code will follow the convention showed bellow, taking into account indentation.



This convention will be used to interact with both characters and scenes. Apart from the simple commands for interaction, there is also a repeat command which will act as a simple loop and only two data types word(string) and number(integer). All this will be demonstrated below.

1	fish say	Character Name
2	"Hello"	
3	fish move 🗲	Move Command
4	left 🛶 🛶 🛶	Move Command Attribute
	fish say	
6	"I'm going to sing."	
	fish sing	
	word name	Word type variable and
_	fish ask	variable name
10	"What's your name?"	
11	name	
	fish say	
13	"hello" + name	Den est service d'uith verset
14	repeat 5	Repeat command with repeat
15	fish jump	times attribute
16	fish say	
17	"I'm happy"	Commands to repeat
	number a_number=5	
	fish domath	
20	+	
21	5,6,a_number	

When running the program all these commands will be reflected on your graphical scene which should include a simple character and a background. The character will react based on the commands issued, it will jump, move, prompt messages and much more.

Commands/Tokens	Target	Number of Attributes	Attributes Description	
Jump	Character	N/A No attributes needed.		
Walk	Character	1	Direction: left, right, forward or backward	
Say	Character	1	Text to say: raw string or word type	
Domath	Character	2	Operator: add, subtract, multiply or divide Number/s to add: can be a raw number or a number type described below.	
Turn	Character	1	Direction: right or left	
Domultiple	Character	1	Character name	
Dance	Character	N/A	No attributes needed.	
Sing	Character	N/A	No attributes needed.	
Grow	Character	N/A	No attributes needed.	
Shrink	Character	N/A	No attributes needed.	
Ask	Character	2	Text to prompt: raw string or word type Variable to store input: word or number type	
Repeat	Character	1	Times to repeat	
Flip	Character	N/A	No attributes needed.	
Run	Character	1	Direction: right or left	
Time	Scene	1	Scene time: day or night	
Word	N/A	N/A	Word type that will store strings.	
Number	N/A	N/A	Number type that will store integers.	

Implementation

Requirements and Tools

The tools and requirements needed to be able to use our programming language are kept to the minimum possible so that it is easy to use in any of the most popular operating systems. It is required that the user has Python version 2.7.11 and PyQt4 installed to make sure the language can be run from the command prompt. The use of an IDE or command prompt directly will be avoided by providing an executable file.

Project Plan

Task	Description	Start Date	End Date
Programming Language Concept and Description	The team will get together to develop a proposal for a programming language to develop.	Feb. 1	Feb. 10
GUI Design and Development	The team will decide on a design for the GUI to aid our programming language and work on the code development.	Feb. 10	Feb. 15
Graphic Design and Animations	The team will design the different scenes and characters for the programming language and will work on the code development of the animations.	Feb. 15	Feb. 22
Scanner Development	The team will decide on a syntax for the programming language and use one of the provided online tools to generate a scanner.	Feb. 22	Feb. 26
Parser Development	The team will work on the development and testing of the parses for our programming language.	Feb. 26	Mar. 7
GUI, Scanner and Parser Integration	The team will integrate all the previous phases on a complete application that will make the GUI work along with the scanner, parser and graphic animations.	Mar. 7	Mar. 24
Final Report and Video Tutorial	The team will work on a video tutorial on how the programming language works and write the final report and documentation needed.	Mar. 24	Mar. 31
Web Page Development	The team will work on the development of a simple webpage to showcase the programming language, give support and make the source code available.	Mar. 31	Abr. 7

Timeline

