Mini Project “1” Report

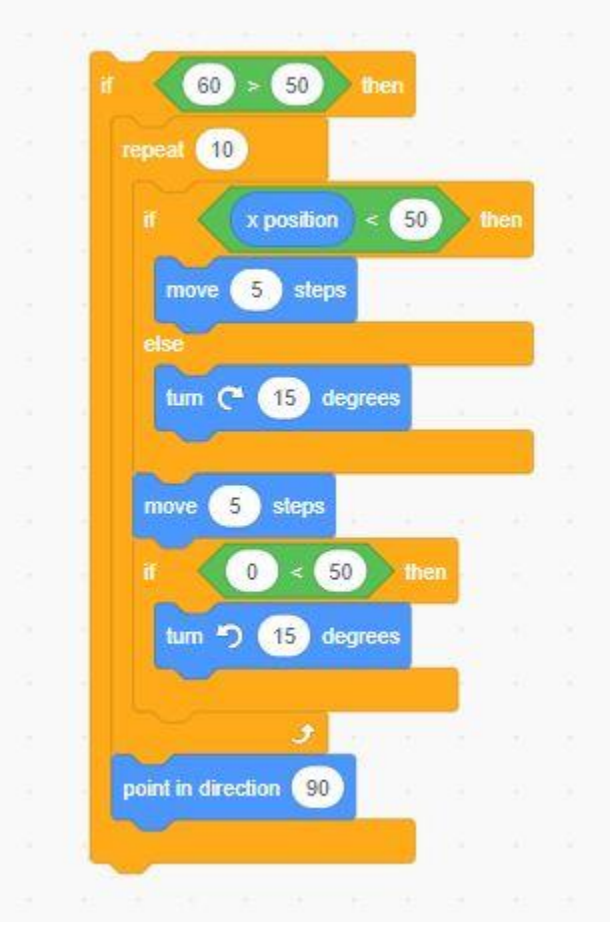
Name:	**ID No :**
• Reem Essam	(37-8875)
• Hadeer Mohamed	(37-5898)
• Nourhan Ahmed	(37-9784)
• Nermin Ayman	(37-18156)
• Nada Saeed	(37-11486)

How to run?

First, create a new project in Unity. Then, Import the unity package in it. And finally run the project. A path for the file will be required, either to write the path of one of the text files (which contains the command that follows the grammar) or write the path of the Json file (the Json file has to be with the Python file in the same directory).

Input/output Example (1):

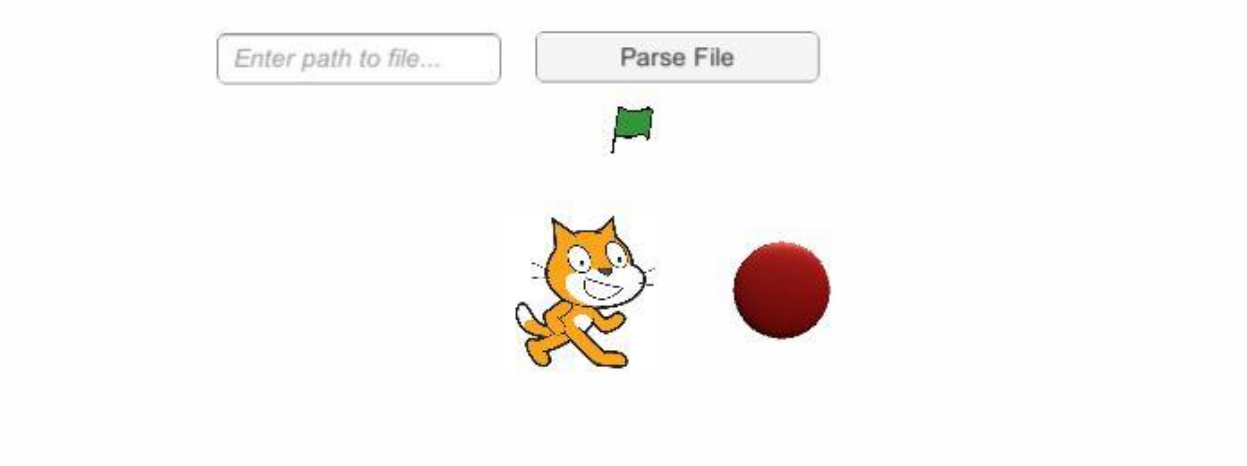
The scratch program is as follows:



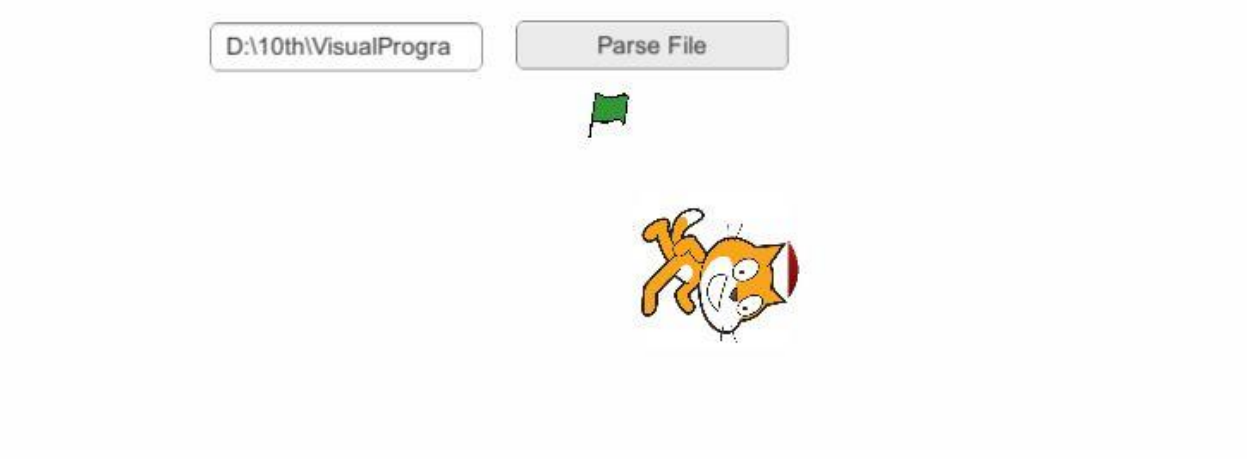
The corresponding text file command:

```
OUT = if 60>50 then Repeat 10 if xposition<50 then Move 5 endThen else Turn Right 15 endElse Move 5 if 0<50 then Turn Left 15 endThen endRepeat Point 180 endThen
```

The output Sprite **“Before”** running the action:

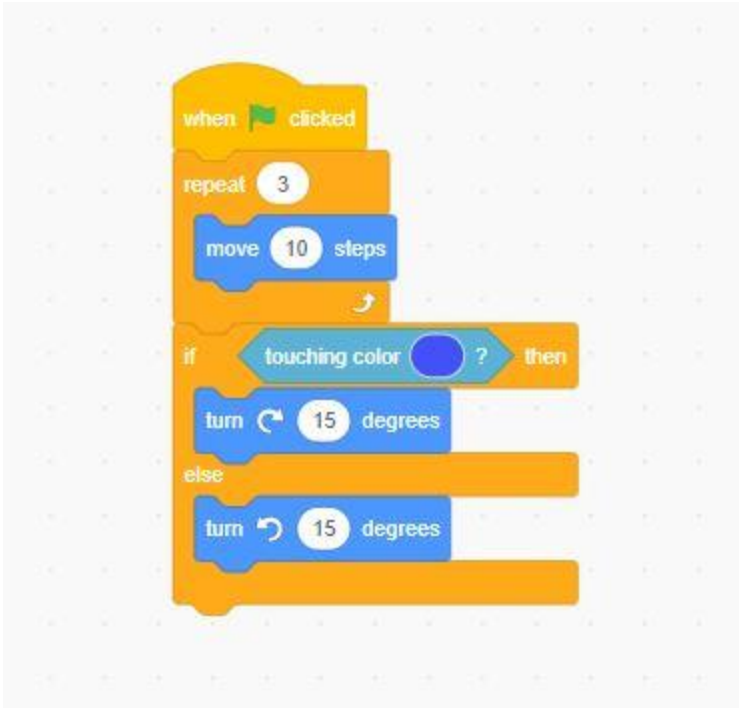


The corresponding Sprite **“After”** running the action:



Input/output Example (2):

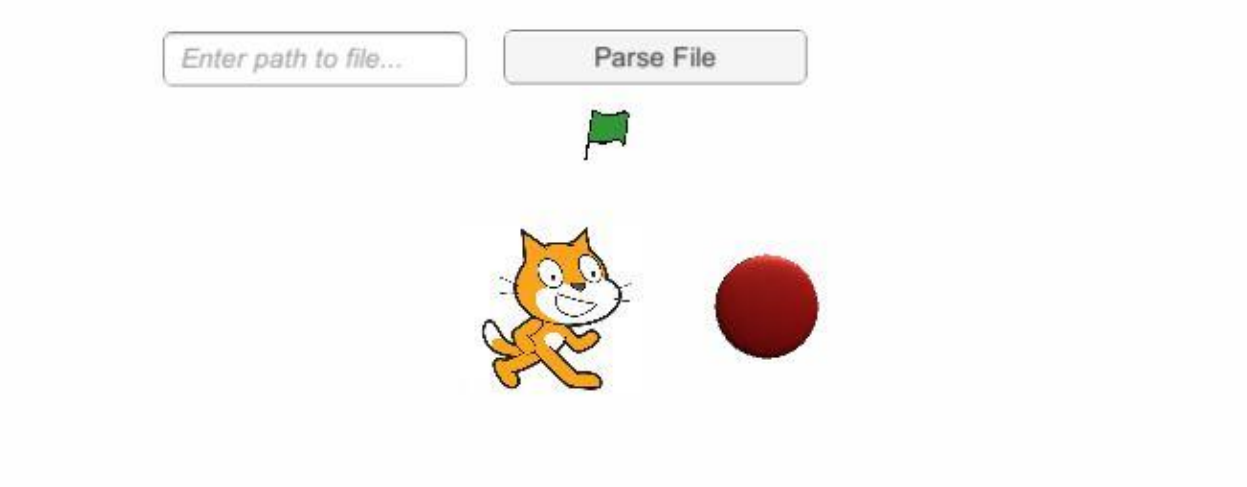
The scratch program is as follows:



The corresponding text file command:

```
OUT = if GREENFLAG then Repeat 3 Move 10 endRepeat if Color"4351f6"istouched then Turn Right 15 endThen else Turn Left 15 endElse
```

The output Sprite **“Before”** running the action:



The corresponding Sprite **“After”** running the action:

Brief Description:

We used two languages (Python and C#). The python code is used to parse the Json file, then outputs the text file that includes the command that follows the grammar. The C# file is used to take the command in the text file as an input and reflects it on the sprite in unity.

- **C# Script:**

  All of the code starts by pressing the “Parse file” button. This part is handled in the “getfile” method which in turn calls “parseJson” method in case the input file was a json file. “parseJson” executes the python program while giving it an input. We provide the process by the directory, the python file name and the Json file that is considered as the main input to the python file. The update method is responsible for a handling a special case, which is if the

program starts by waiting for a key press. After preparing the input our main method, which is “ParseTextFileHelper” is called. This method parses the command action by action calling for the corresponding methods according to the action. There is a method that visualizes each action on the sprite (e.g. move 10 corresponds to Move (10) method). However, the repeat action has an extra method that handles the nested repeat called “getRepeat”. In addition, there is a method called “condHandler” that returns Boolean (true or false) according to the condition itself. “whenGreenFlagPressed” method is used to handle the case of green flag pressing. Moreover, “OnTriggerStay” method is used to detect the collision of the sprite with the color. “getNumber” method is used to get the value of mouse-x, mouse-x, position-x and position-y. “ParseTextFile” method is used to call the “ParseTextFileHelper” method in most of the cases except in the event cases, as they are handled externally in another separate methods.

- **Python code:**

  This file is used to translate the whole JSON file into text file compatible with our grammar. This file is divided into several methods. First, “getKey” method is used to translate the key value written in the JSON format to our grammar format. “getParent” method is used to

determine the end of the following blocks: (if,else,repeat). “opcodeParsing” method is used to get the action op code value and translate it to the corresponding action names in our grammar. “getInputs” method is used to get the exact input value for the specific action from the JSON. And finally the main method which is called “parseJson” which takes the JSON file itself and apply the “opcodeParsing” method on it with the help of the other methods in order to output the correct corresponding text file.