<C>

Cerious

CS team - 05

Software Engineering

Team Members

Sagrika Rastogi (Leader) - 201451071

Vaishnav Vishal - 201451051

K. Naveen Kumar - 201451074

Mukesh Sahu - 201451003

Neetu Shah - 201451015

G. V. Sai Rahul - 201451054

Tarachand Gurjar - 201451067

N. Santosh - 201451063

Ayush Pandey - 201451057

Problem Statement

Conversion of Natural Language to C code.

A novice who has little or no idea about programming and does not wish to go through verbose documentation does not have many interactive options available to him/her.

Solution

To provide a facility that lets user use Natural Language to produce working programs.

An interactive platform makes the process of learning more natural.

Cerious : Project Overview

Cerious is intended to work as an assistant to a user by providing the facility of converting speech or text in Natural Language to C code.

This is a Desktop application that would be useful to a beginner who may use it as a learning tool.

Cerious also provides predefined snippets of common functions.

Who is our Target Audience

Beginners in C

This class of target audience include novice programmers and people who occasionally require coding.

Such people may have an algorithm ready in natural language to solve the problem but do not want to go through the tedium of documentation to write a simple program

Good Programmers

This class of users are good at coding. They may use Cerious at intervals to break repetitive strain of typing. It may also help in case a particular programmer has health issues (hand realted).

Moreover, they may utilize code snippets available for direct inclusion into code.

How we came up with this idea

We all remembered the difficult experience we had trying to learn programming. So our initial idea was to develop an assistant where the user could speak in natural language.

After taking interviews of both faculty and students, we pivoted towards the direction of tutorials and also incorporated the text feature where the user can type in natural language commands, so that it can work even when there is no internet available.

Cerious : Why is our software better than others?

This software is first of its kind.

We took the best pre-existing softwares and collaborated them to make a new class of learning platform or an assistant according to the need of the user.

The user sees his natural logic being converted to code, thus making the whole process of learning programming a lot more intuitive.

We do not let wrong things happen.

Interviews

In the interviews, we presented the idea of our software to both faculty and students and got their responses regarding the idea.

Interviews made us aware about the most common problems they face.

This helped us in clearly framing our problem statement and also creation of survey form to obtain the opinion of a large number of people.

Surveys

We designed a survey form for the students of our institute which had 178 responses.

In that survey form, out of 178 people, 132 had agreed that it will be helpful for a novice programmer to see the construction of code from his natural speech

This survey revealed that pointers was the area of most difficulty for people.

What We Do: Functionalities

Functionality to be able to declare variables.

Functionality to be able to speak in the microphone.

Functionality to be able to call in built Libraries into the code.

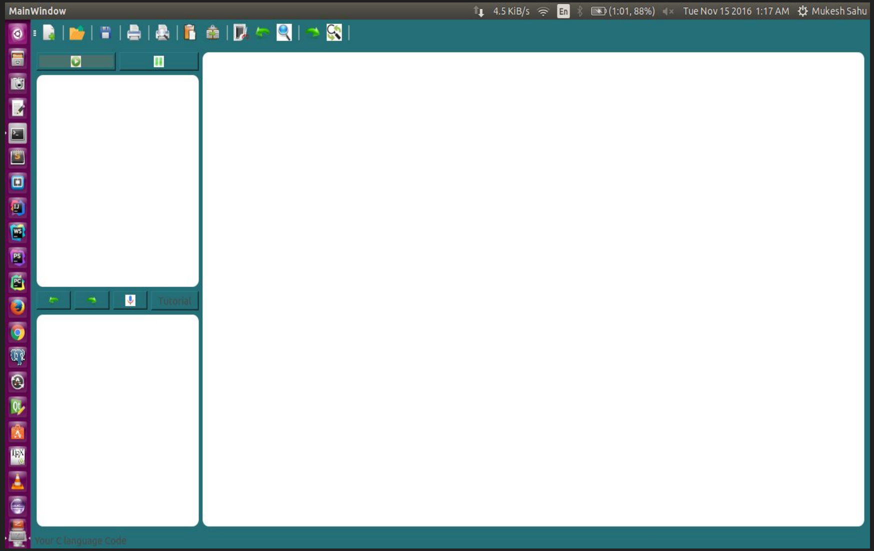
Functionality to be able to run loops.

Functionality to be able to run conditional statements.

Functionality to be able to display the code on a GUI.

Functionality to be able to save the codes onto the machine.

User Interaction 1/2

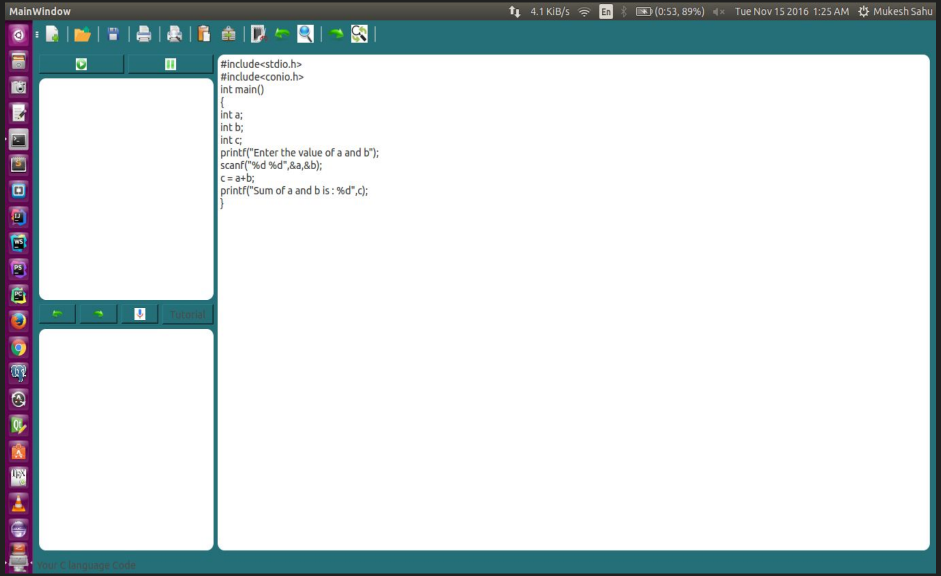


At the start of software, there will be a home screen which presents all the features of the software.

Speech input, editing options and the tutorial option are there in the homepage of the software.

User can simply edit and perform functionalities of an editor on the code in the software.

User Interaction 2/2



```
#include<stdio.h>
#include<conio.h>
int main()
{
int a;
int b;
int c;
printf("Enter the value of a and b");
scanf("%d %d",&a,&b);
c = a+b;
printf("Sum of a and b is : %d",c);
}
```

User can type the code and speak simultaneously.

In the tutorial part, user will get a detailed description about the language or the syntax of the language which can be used over a particular algorithm.

There will be a display block where all of the actions done by the user are displayed to him.

SDLC Model - Incremental Model

Firstly, we have to make a working model of our software, which gives us the idea about taking speech input from the user, conversion of speech input into text input and finally, generation of code.

After checking the feasibility of this model, we incremented various functionalities of C.

Design

Design of this software is of a simple editor with an additive feature of giving input in the form of speech.

We have used the Redux architecture for the development of the software.

This helps us for implementing easy testability and undo/redo functionality.

For tutorials, we can give the address of presentation or documentation of that language.

Software Specification 1/2

All the back end has been done in Python and front end code has been done in PyQt5.

For the conversion of speech input we have used **Google Speech to Text API**, whose source code is free.

For the generation of code, we use **NLTK** toolkit in Python, which helps us to understand the meaning of the input. Toolkit and dataset both the source code are available for free.

Software Specification 2/2

Google Speech to Text API can convert over 80 languages from speech to text.

NLTK has a context free grammar sorted data of dictionary, stored in it.

It uses that stored data for comparing the input and finally understanding what the user wants to do and generates the code accordingly.

Testing

Firstly, after the model is made, each functionality such as running of the software on an environment, taking speech input, conversion of speech input and generation of code is tested.

After that, as the increment is added in the software, the testing of those functionalities had also been done along with integration testing.

Features that could not be implemented

We had made this software only for C language.

Speech input can be given in those languages which can be converted by **Google Speech to Text API** and whose data is stored in **NLTK** library.

Future Scope

We will include other programming languages like C++, Python etc.

This can serve as a platform where a simple programmer can start a beginning of becoming a professional developer.

Proper, structured tutorials can be constructed to learn a particular language.

What we learnt

To function as a team and learn how to resolve conflicts.

Team management, time management and coordination for an efficient working environment.

Prioritizing suggestions and ideas.

How Software Development Life Cycle works

Anything can happen at the last moment. Be prepared.

THANK YOU