



Typing-Teaching System for Beginning Programmers

Andres Zamora
Josh Chavez
D'Angelo Felix
Juan Orozco



Introduction

- What is it?
 - Typing game for beginner programmers
- What's the purpose?
 - Gives a just start for those wanting to learn a certain computer language

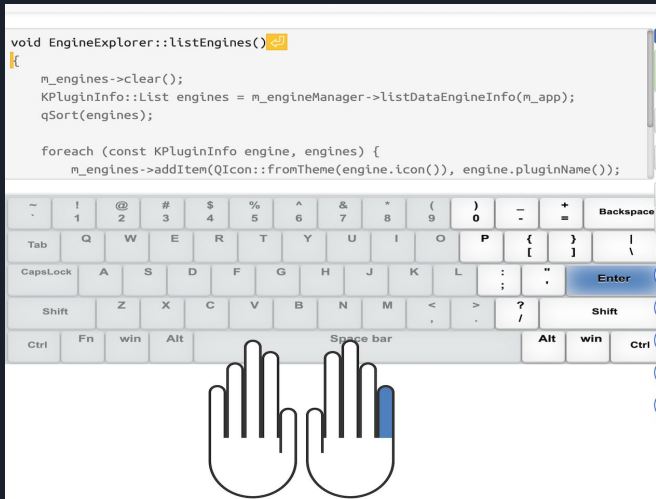


Background

- Origins of the idea
 - Reflected on our own experience
- Why is it needed?
 - There are many that don't know how to type, let alone know how to code

Reason Our Game Is So Unique

Through research we have found countless typing and coding games, but there is no game that combines the two. The issue is that the typing games are extremely interactive and seem exciting for the user, but the coding games just have you copy the code that is given with very little explanation of what is actually happening.





Methods

- Methods of teaching
 - Timed Objectives
 - Memorization
 - Techniques
 - Quizzes



Methods(cont'd)

- Unity
 - Offers models, sprites, scenes and materials
 - Team collaboration - built in version control
 - Works well with 2D Games
 - Uses C#

Game Features

Basic C++ Commands

1. #define

This C++ Command can be used to substitute a particular value throughout the file in which it is located. This helps the compiler to go through the entire file and replaces the name of that macro which was created with the value which was defined. The replacement of this string stops at the end of the line

```
#define PI 3.14159
#define NEWLINE '\n'
```

2. #error

This Command helps in displaying the error message which is defined and indicates the compiler to stop when this command is encountered. When the compiler comes across this command then the compiler displays the line number and message which is mentioned. This helps the programmer in debugging easily.

```
#error message
```

3. #include

This includes Command helps in including the mentioned files in the code and making use of those. It can be used in two ways:

- #include <filename>
- #include "filename"

When angular brackets are used when a file needs to search or include through the standard library directories.

```
#line line_number "filename"
```

6. #if

This Command helps in specifying conditions that are to be followed. It enables the user to simply apply logic control and if the condition is true then the code will immediately follow the command that will be compiled.

```
#ifdef CPP11MS_DLL
#define DLEXPORTE __declspec(dllexport)
#else
#define DLEXPORTE __declspec(dllimport)
#endif
extern DLEXPORTE void test2()
{
    printf("HI");
}
```

7. #ifndef

This C++ Command is useful to check if a macro has been defined by #define statement or not. If it is already defined then it executes the next statement that follows.

```
#define NON
int main()
{
    #ifndef NON
        printf("hello");
        int i;
        address >> i;
        return 0;
    }
```

8. Cout

When a user wants to print some text or value of any variable that was defined previously then cout is used. It displays all values on the screen.

```
int x = 5;
int y = 23;
cout <<"x"; // prints 5 on screen
cout <<"y"; // prints 23 on screen
```

9. Cin

In order to accept the value from the user and assign it to a particular variable, the developer must make use of cin variable. Any value can be accepted from the user and assigned to the variable of any data type that is created.

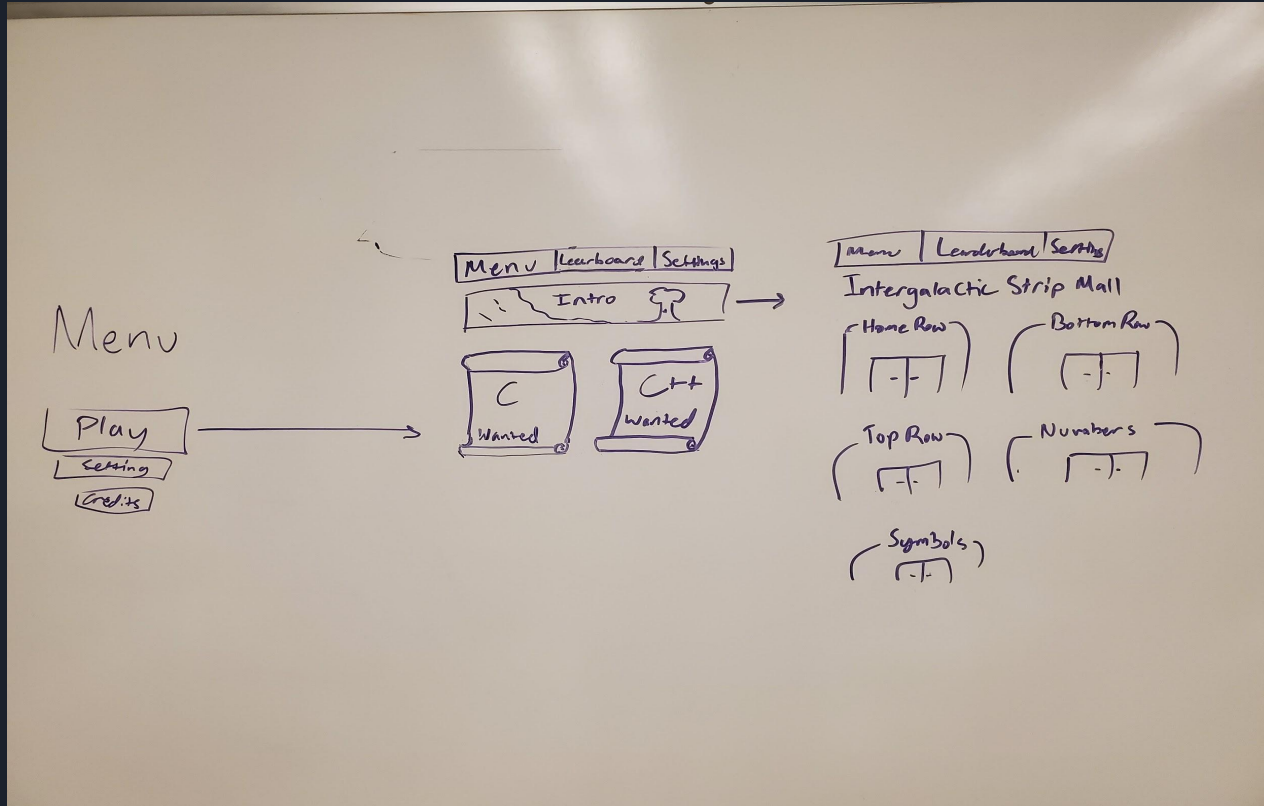
```
int x;
cin >> k; //input assigns a value to the variable x
```

10. __FILE__

This variable helps in representing the current file that is being processed. It can be defined and use in the entire

Before each game we plan to let the user know exactly what commands and objects they are using beforehand. This gives the user better knowledge of what they are using before they actually begin to play.

Game Features(cont'd)





Game Features(Cont)

- To Teach General Keyboard Skills
- User will be able to choose the programming language of their choosing
- Mini Games
- User will learn to code big chunks of code
- The game will autosave depending on level
- Mini Games will increase difficulty
- Leaderboard for each game to show progress



Goals:

- Teach typing and coding, while keeping it fun for all.
 - Should lead to improvements in both coding & typing when they finish the program.
- By the end of the semester:
 - Aim to have functionality in our program.
 - Have a working prototype for testing.



Potential Goals:

- Being able to have customization for the avatar character.
 - By implementing a points system, in which points are gained by means of completing minigames.
 - Trade those points towards cosmetics.
- Try to incorporate original music pieces rather than using copyright free music.
 - Some audio software tools include: Audacity, Adobe Soundbooth CS4, Logic Studio , and FLStudio
 - <https://answers.unity.com/questions/10684/preferred-audio-software-tools-for-unity-game-deve.html>



Questions?



Thank you for your time