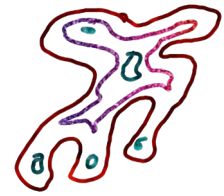




CSC 310 Human Computer Interaction

Heuristic Evaluation Report Template



Dated
11/22/2020

Prepared By:

GROUP NAME: Group 1

CONTRIBUTOR: Cason Konzer

Evaluation Of:

Name of system being evaluated: CODE : : BLOCKS v 2003

Group that created the system being evaluated: Group 5

Feedback & Critical Incidence

<< Record your observations in the table on the following page, based on your observations and notes taken during the evaluation.

Description of columns in the table are as follows:

Prototype Screen/Page:

Which screen of the user interface was being tested at the point of feedback/critical incidence/problem.

Name of Heuristic:

Which of the 9 heuristics is being referenced. Enter the full name of the heuristic. Refer to the heuristic evaluation lecture slides for the 9 design principles.

Reason for negative feedback / breakdown:

Explain the reasons why the interface violates this heuristic. Be sure to be clear about *where* in the screen you are referencing.

Scope:

Describe the scope of the feedback or the problem; include whether the scope of the issue is throughout the product or within a specific screen or screens. If the problems are specific to a page, include the appropriate page numbers.

Severity (H/M/L) :

Your assessment as to whether the implication of the feedback is *low*, *medium*, or *high* severity, and a justification for why you are giving it that rating.

Way(s) to rectify and Tradeoffs (i.e., why the fix might not work):

Suggestion for the modifications that might be made to the user interface to address the issue or issues in this row. You MUST include trade-offs to be credible. If you can't think of some bad trade-off, say so.

Action taken and Justification:

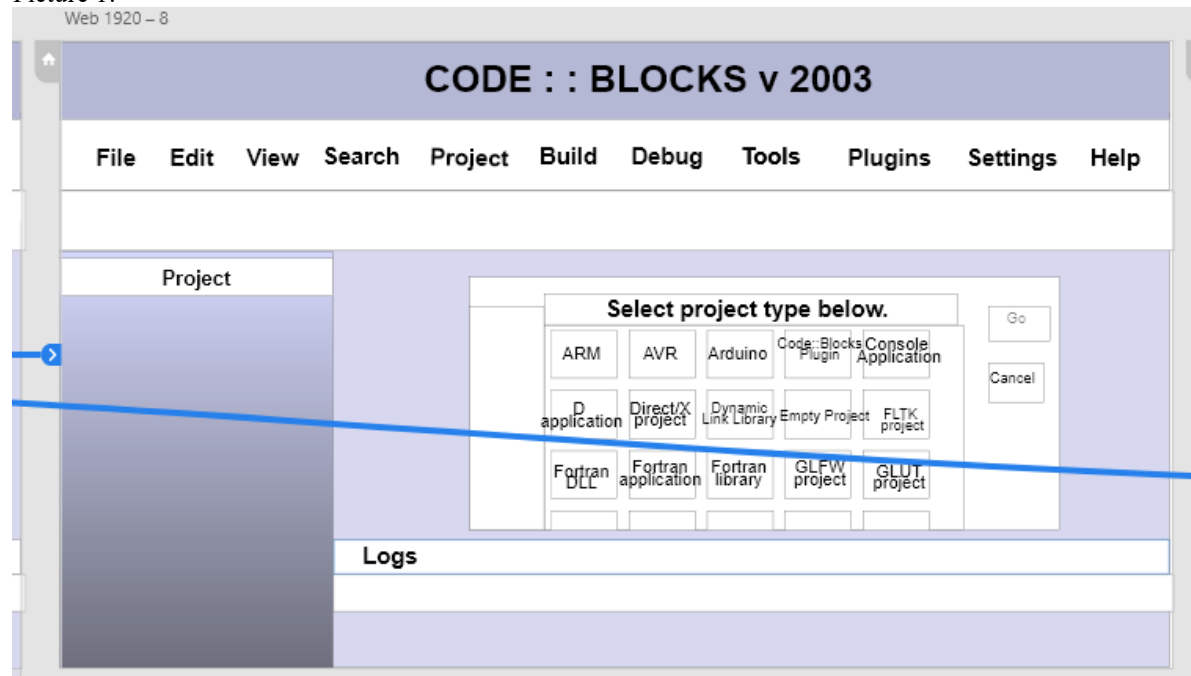
DO NOT FILL IN THIS COLUMN. This will be filled in by the person who created the system being evaluated, to say what he did with respect to the recommendations you make.

#	Prototype Screen	Name of Heuristic	Reason for negative feedback / breakdown	Scope	Severity (High/ Medium/ Low) and Justification for giving it that rating	Way(s) to rectify and any Tradeoffs (i.e., why the fix might not work)	Action taken and Justification (to be filled in later by the person who created the system being evaluated)
1	See Picture 1	Be Consistent, Provide clearly marked exits	This is a “dead” screen. The Read Me provides instructions that are not consistent with the model. There is no way for the user to leave this screen while they are directed here by the instructions.	Error occurs in this screen as well as some others	High. At this point the user has made 3 total clicks and is stuck on a screen. The cancel button does not function nor does any button on the screen.	Implement the cancel button to return to home screen. Implement the Console Application button so the user can complete the given directions.	
2	See Picture 2	Be Consistent, Provide clearly marked exits	Read me directs user to press an edit mode button that is not available. Prototype has an edit profile button that does not work. Exit button is very hard to see.	Various screens do not have clearly marked exits, whole project is quite inconsistent with project directions and read me.	High. User has no way of advancing through the given script at this point.	Implement the edit button, either change the read me or project, mode/profile, so they are consistent. Make the exit button more pronounced.	
3	See Picture 3	Simple and Natural Dialog, Speak the User’s Language, Be Consistent	Am unsure what the given script actually is, read me assume user has a background with technical language, wordage does not logically flow with the English language. (Do not think people would talk like this).	Project Wide. The read me is crucial for providing instructions for completion of tasks.	High. Read me is not consistent with the prototype. Read me is hard to follow. Some wordage is not in layman terms and users may get stuck.	Match the read me with the project so that they are consistent. Provide clear script labeling. Speak in layman terms so the prototype can be tested with new users.	

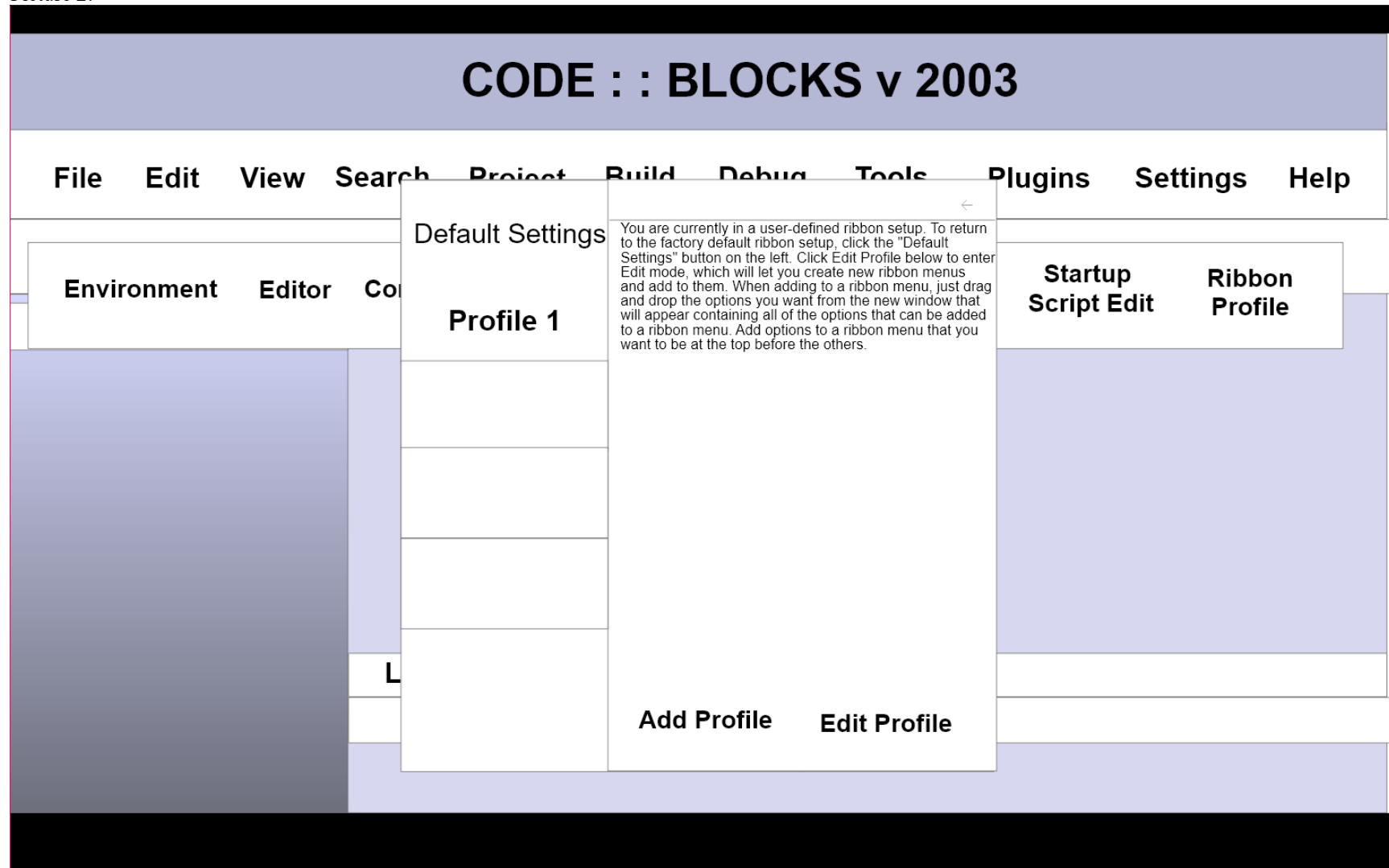
#	Prototype Screen	Name of Heuristic	Reason for negative feedback / breakdown	Scope	Severity (High/ Medium/ Low) and Justification for giving it that rating	Way(s) to rectify and any Tradeoffs (i.e., why the fix might not work)	Action taken and Justification (to be filled in <u>later</u> by the person who created the system being evaluated)
4	See Picture 4	Provide clearly marked exits	No exit available	Present within multiple project screens	Low. User can continue to navigate without the exit but this is a necessary feature.	Add in an exit	
5	See Picture 5	Be Consistent, Provide clearly marked exits	This suffers similar errors as the only functioning click through is clicking Main.cpp which simply returns the user to the previous screen. The Read Me says to click Toggle Breakpoint, which is not consistent with what the prototype offers.	Inconsistency and lack of implementation is project wide.	High. Again, at this point the user cannot complete directions as the read me indicates. Additionally, there is no clear exit and the user will easily be stuck here.	Implementation of the toggle breakpoint button as the read me states. Provide an exit button.	
6	See Picture 6	Provide Help	While the help button opens up various help options there is no help provided	Project wide	Low. Help can be found in the read me.	Implement help dialog.	

#	Prototype Screen	Name of Heuristic	Reason for negative feedback / breakdown	Scope	Severity (High/ Medium/ Low) and Justification for giving it that rating	Way(s) to rectify and any Tradeoffs (i.e., why the fix might not work)	Action taken and Justification (to be filled in <u>later</u> by the person who created the system being evaluated)
7	Project Wide	Provide Feedback, Be Consistent	No feedback to the users for any unimplemented buttons, buttons are only available on certain screens	Project Wide	Medium. Unimplemented buttons do not provide any feedback for the user and while buttons are available on one screen they may not be on the next.	Provide popups/ screens with a message of the “coming soon...” type. Be consistent and implement buttons on all screens.	

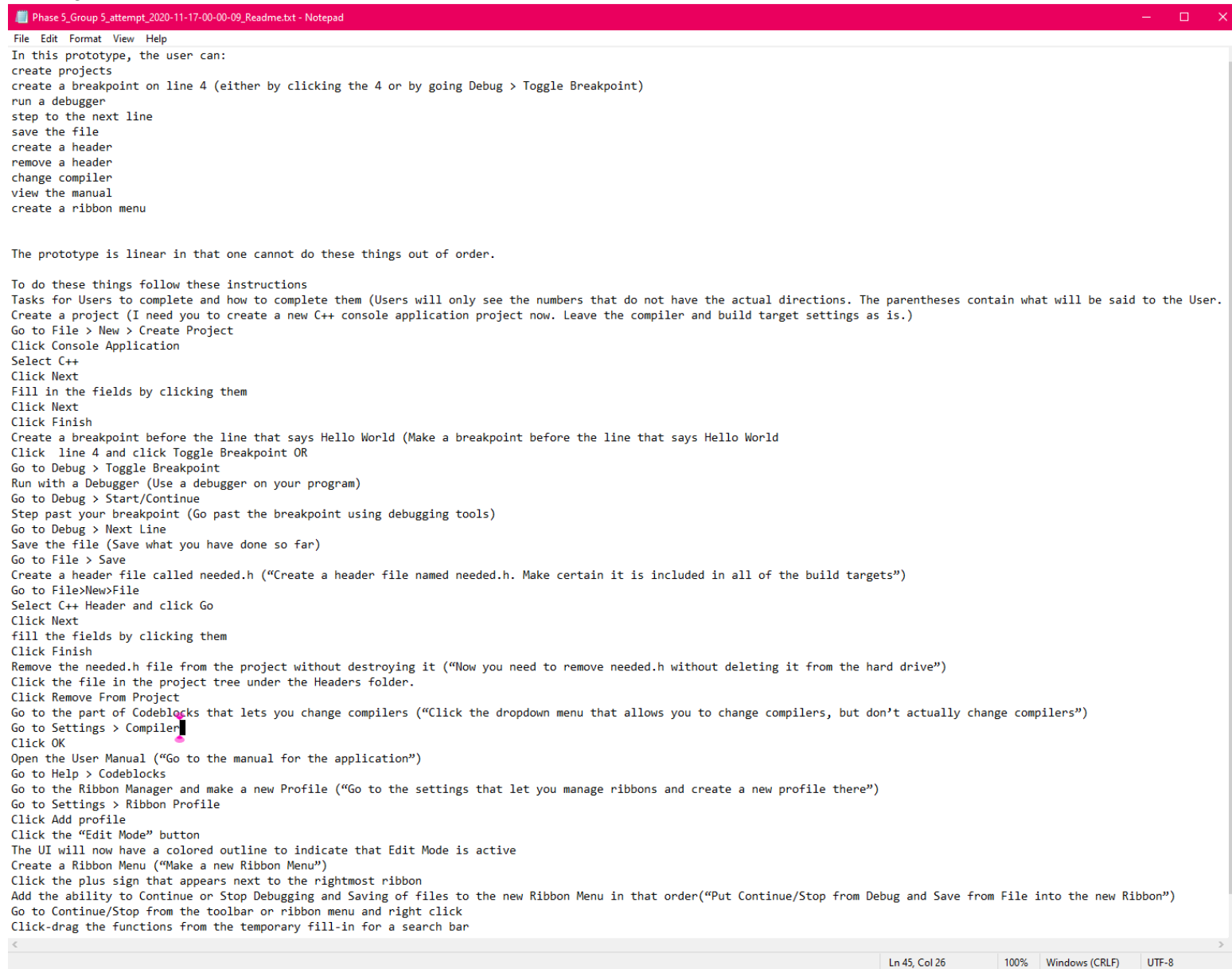
Picture 1:



Picture 2:



Picture 3:



Phase 5_Group 5_attempt_2020-11-17-00-00-09_Readme.txt - Notepad

File Edit Format View Help

In this prototype, the user can:

- create projects
- create a breakpoint on line 4 (either by clicking the 4 or by going Debug > Toggle Breakpoint)
- run a debugger
- step to the next line
- save the file
- create a header
- remove a header
- change compiler
- view the manual
- create a ribbon menu

The prototype is linear in that one cannot do these things out of order.

To do these things follow these instructions

Tasks for Users to complete and how to complete them (Users will only see the numbers that do not have the actual directions. The parentheses contain what will be said to the User.)

Create a project (I need you to create a new C++ console application project now. Leave the compiler and build target settings as is.)

Go to File > New > Create Project

Click Console Application

Select C++

Click Next

Fill in the fields by clicking them

Click Next

Click Finish

Create a breakpoint before the line that says Hello World (Make a breakpoint before the line that says Hello World)

Click line 4 and click Toggle Breakpoint OR

Go to Debug > Toggle Breakpoint

Run with a Debugger (Use a debugger on your program)

Go to Debug > Start/Continue

Step past your breakpoint (Go past the breakpoint using debugging tools)

Go to Debug > Next Line

Save the file (Save what you have done so far)

Go to File > Save

Create a header file called needed.h ("Create a header file named needed.h. Make certain it is included in all of the build targets")

Go to File>New>File

Select C++ Header and click Go

Click Next

fill the fields by clicking them

Click Finish

Remove the needed.h file from the project without destroying it ("Now you need to remove needed.h without deleting it from the hard drive")

Click the file in the project tree under the Headers folder.

Click Remove From Project

Go to the part of Codeblocks that lets you change compilers ("Click the dropdown menu that allows you to change compilers, but don't actually change compilers")

Go to Settings > Compiler

Click OK

Open the User Manual ("Go to the manual for the application")

Go to Help > Codeblocks

Go to the Ribbon Manager and make a new Profile ("Go to the settings that let you manage ribbons and create a new profile there")

Go to Settings > Ribbon Profile

Click Add profile

Click the "Edit Mode" button

The UI will now have a colored outline to indicate that Edit Mode is active

Create a Ribbon Menu ("Make a new Ribbon Menu")

Click the plus sign that appears next to the rightmost ribbon

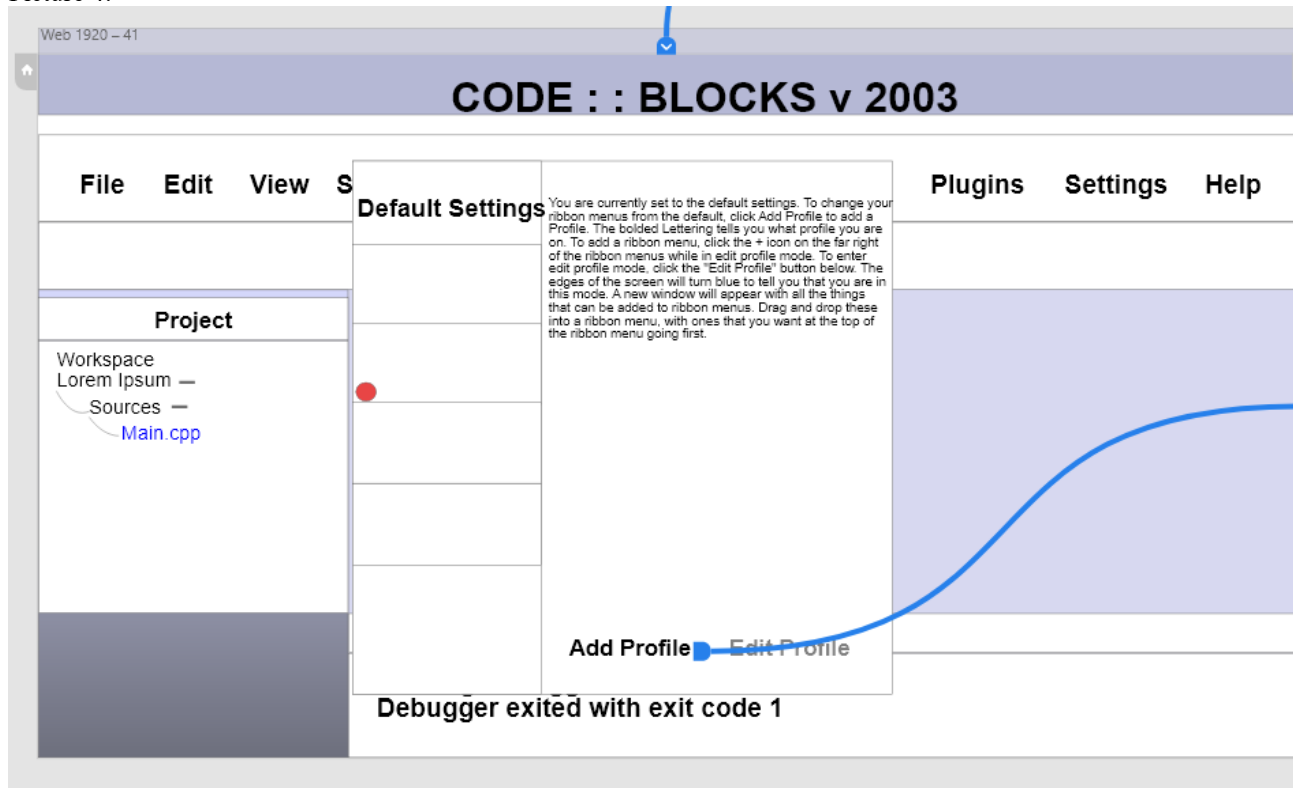
Add the ability to Continue or Stop Debugging and Saving of files to the new Ribbon Menu in that order("Put Continue/Stop from Debug and Save from File into the new Ribbon")

Go to Continue/Stop from the toolbar or ribbon menu and right click

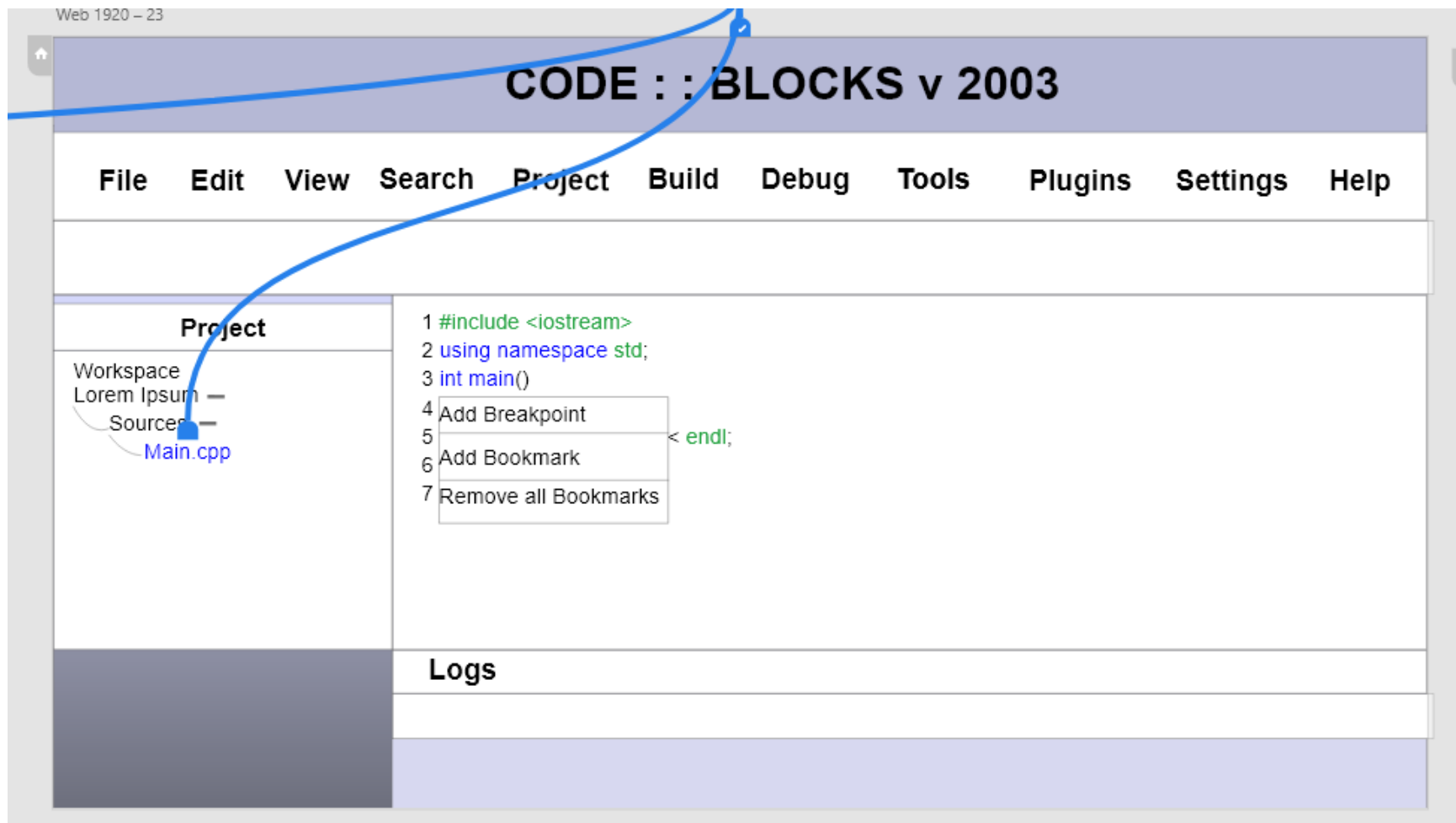
Click-drag the functions from the temporary fill-in for a search bar

Ln 45, Col 26 100% Windows (CRLF) UTF-8

Picture 4:



Picture 5:



Picture 6:

