Code Workstation

Requirements Analysis Document

Lukas Rickard - Project Leader - l_rickard@u.pacific.edu
Jake MacMillan - Integrations Leader - j_macmillan1@u.pacific.edu

Project Scope

What will be in Code Workstation (On initial release)

- A text editor
    - Standard open and save dialogs
- Buttons which run a user entered shell command
    - ~~Customizable button pictures and colors~~
    - Customizable Button Name
    - Right click Edit and delete options
- Text boxes which display command output (Command Boxes)
    - Boxes can be moved to any side of the screen
    - Boxes can be deleted and edited with right click
- Shortcuts
    - Ctrl plus any key can be used as a shortcut to any program action
        - Activate Buttons
        - Create a new Button or Command Box
        - Save or Open a file
        - Perform cursor movements in text editor
- Persistent settings
    - Closing the window will not lose settings
    - Settings can be saved and loaded deliberately

What will not be in Code Workstation (On initial release)

- A built in debugger
- Groundbreaking concurrent efficiency
- Beautiful interface
- Built in compilers
- Syntax highlighting (Planned in future releases)

Requirements Summaries

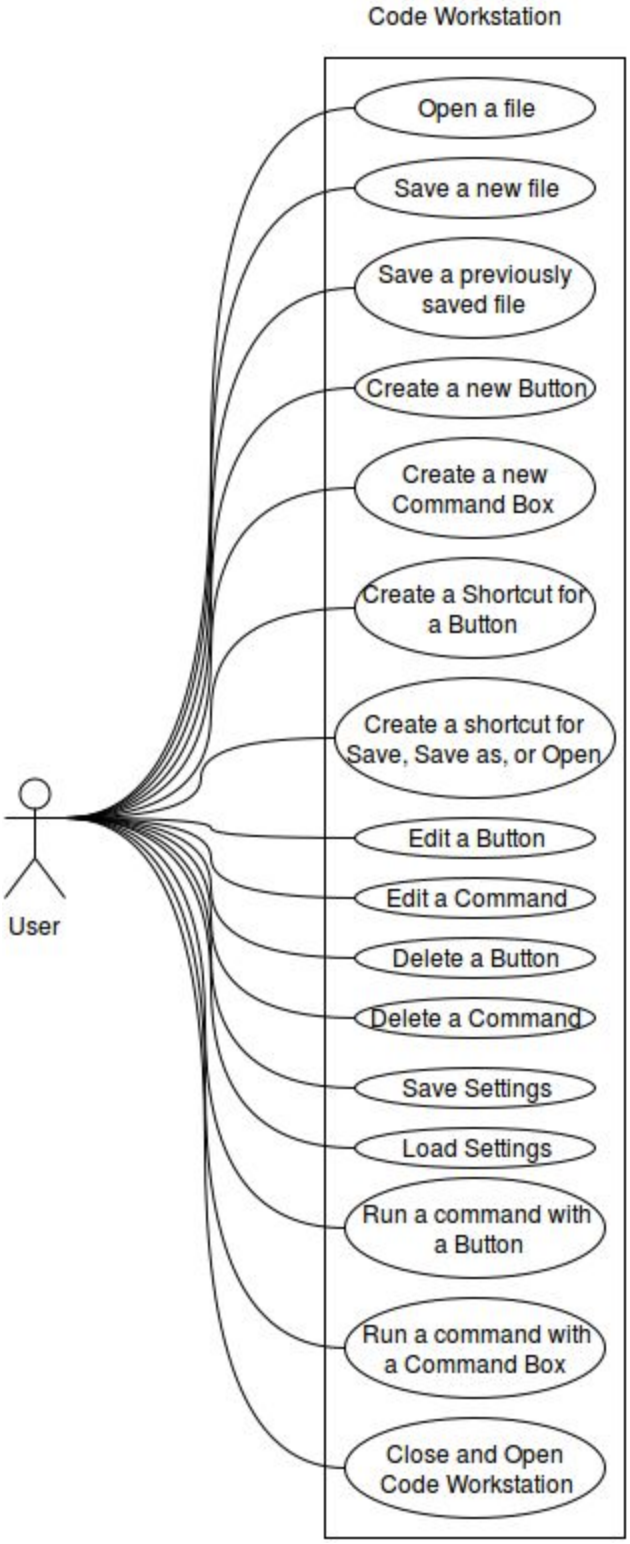
Non-functional Requirements

- Usability
- Intuitiveness
- Extensibility (users can expand on functionality)
- Responsiveness

Functional Requirements

- User interface can be created and customized
- Right click commands allow convenient deleting and editing
- Settings can be saved and loaded
- Files can be opened, edited, and saved

Use Case Diagram



Casual Use Case Descriptions

Open a file:
User clicks File->Open. A file selection dialog appears. User selects a file. User clicks okay. The file is displayed.

Save a new file:
User clicks File->Save. A file selection dialog appears. User selects or enters a file name. User clicks okay. File is saved.

Save a previously saved file:
User clicks File->Save. File is saved to the previously entered file name.

Create a new Button:
User clicks New->Button OR the “New Button” button. A new button dialog appears. User enters a button name, a command, and an output location. Users presses Okay. The dialog disappears and a new button with the selected name appears.

Create a new Command Box:
User clicks New->Command Box. A new Command Box dialog appears. User enters a name and any desired information. User presses okay. The dialog disappears and a now command box appears a the selected location.

Create a shortcut for a Button:
User clicks Edit->Shortcuts. A shortcut dialog appears. Users enters a key and selects a button name from a drop down. User clicks save. Shortcut description appears. Users clicks Okay. The dialog disappears and the shortcut now activate the selected button.

Create a shortcut for Save, Save as, or Open:
User clicks Edit->Shortcuts. A shortcut dialog appears. Users enters a key and Open, Save, or Save as. User clicks save. Shortcut description appears. Users clicks Okay. The dialog disappears and the shortcut now activate the selected command.

Edit a Button:
User right clicks a Button. User clicks “Edit”. A dialog similar to the new button dialog appears. User changes settings. User clicks okay. The button is updated.

Edit a Command Box:
User right clicks a Command Box. User clicks “Edit”. A dialog similar to the new Command Box dialog appears. User changes command box settings. The command box updates with the new settings, possibly moving to a new location in the process.

Delete a Button:
User right clicks a Button. User clicks “Delete”. The button disappears and all other buttons stretch to fill the lost space.

Delete a Command Box:
User right clicks a Command Box. User clicks “Delete”. The command box disappears and other command boxes in the same area stretch to fill the lost space.

Save Settings:
User clicks File->Save Settings. A file selection dialog appears. The user selects a file name. User clicks Okay. The settings file is saved with the selected name.

Load Settings:
User clicks File->Load Settings. A file selection dialog appears. The user selects a file. User clicks Okay. The window configuration completely changes to the selected settings file’s description.

Run a command with a Button:
User clicks a button. The entered command’s output or an error appears in the selected Command Box.

Run a command with a Command Box:
User Edits a command box. User ticks the Auto Run check box. User enters a command and an interval. User clicks okay. The command box immediately populates with the output or error from the selected command. The command box updates every few seconds with fresh output from the command.

Close and Open Code Workstation:
User clicks the X at the top right part of Code Workstation. The window closes and the settings are saved automatically. User launches Code Workstation. All the settings present before closing are available in the relaunched Code Workstation.

Fully Dressed Use Case Descriptions - User Stories

Title: Open a file
Preconditions:
- User has started Code Workstation

Main Flow:
1. User clicks File->Open.
2. A file selection dialog appears.
3. User selects a file.
4. User clicks the Okay button.

Exceptional Flows:
3b. User closes the selection dialog with Cancel or otherwise
4b. The selection dialog disappears and nothing is updated

Postconditions:
1. The contents of the selected file is displayed and available for editing in text box.

Title: Save a new file
Preconditions:
- User has started Code Workstation

Main Flow:
1. User clicks File->Save.
2. A file selection dialog appears.
3. User selects a file or enters a new file name.
4. User clicks the Okay button.
5. The Dialog disappears

Exceptional Flow:
3b. User closes the selection dialog.
4b. The selection dialog disappears.

Postconditions:
1. The file is correctly saved in the file system under the entered name.

Title: Save a previously saved file
Preconditions:
- US: “Save a new file”

Main Flow:
1. User clicks File-Save.

Exceptional Flow:
NOTHING

Postconditions:

1. The file is saved correctly with the same name that it was saved with previously

Title: Create a new Button

Preconditions:

- User has launched Code Workstation

Main Flow:

1. User clicks New->Button or the “New Button” button.
2. A new button dialog appears.
3. User enters a button name, a command, and an output location.
4. Users presses Okay.
5. The dialog disappears

Exceptional Flow:

3b. User does not enter information in all the necessary fields.
4b. User clicks Okay
5b. An error dialog appears asking them to enter all the necessary information

Postconditions:

1. A new button with the entered name appears

Title: Create a new Command Box

Preconditions:

- User has launched Code Workstation

Main Flow:

1. User clicks New->Command Box.
2. A new Command Box dialog appears.
3. User enters a name and any desired information.
4. User presses Okay.
5. The dialog disappears.

Exceptional Flow:

3b. User does not select a location for the command box.
4b. User presses Okay.
5b. The dialog disappears
6b. The new Command box appears on the right side of the screen (default)
3c. User does not enter a name.
4c. User presses Okay.
5b. An error dialog appears asking the user to enter a name.
3d. User checks Auto-Update without entering a command.
4d. User clicks Okay.
5d. An error dialog appears asking the user to enter a command

Postconditions:

1. A new command box appears at the selected location.

Title: Create a shortcut for a Button

Preconditions:

- US: “Create a new Button”

Main Flow:

1. User clicks Edit->Shortcuts.
2. A shortcut dialog appears.
3. Users enters a key and selects a button name from a drop down.
4. User clicks Save Shortcut.
5. Shortcut description appears in the table.
6. Users clicks Okay.

Exceptional Flow:

3b. User enters a key already in an existing shortcut.
4b. User clicks Save Shortcut.
5b. The old shortcut will disappear from the list.
3c. User selects A, U, C, X, V, or Z for the shortcut.
4c. User clicks Save Shortcut.
5c. An error dialog appears telling the user this key is reserved for operating system commands

Postconditions:

1. The dialog disappears and the shortcut now activate the selected button.

Title: Create a shortcut for Save, Save as, or Open

Preconditions:

- User has started Code Workstation

Main Flow:

1. User clicks Edit->Shortcuts.
2. A shortcut dialog appears.
3. Users enters a key
4. User selects Open, Save, or Save as from the dropdown.
5. User clicks Save Shortcut.
6. Shortcut description appears.
7. Users clicks Okay.

Exceptional Flow:

See US “Create a shortcut for a Button”

Postconditions:

The dialog disappears and the shortcut now activate the selected command.

Title: Edit a Button

Preconditions:

- US: “Create a new Button”

Main Flow:

1. User right clicks a Button.
2. User clicks “Edit”.
3. A dialog similar to the new button dialog appears.
4. User changes settings.
5. User clicks okay.

Exceptional Flow:

4b. User removes a setting.

5b. User click Okay.

6b. An error dialog appears telling the user they need to enter all the required information.

Postconditions:

The button is updated with the new settings. Including having a new name.

Title: Edit a Command Box

Preconditions:

- US: “Create a new Command Box”

Main Flow:

1. User right clicks a Command Box.
2. User clicks “Edit”.
3. A dialog similar to the new Command Box dialog appears.
4. User changes command box settings.
5. User clicks Okay.
6. The dialog disappears.

Exceptional Flow:

4b. User unticks the Auto Run check box.

5b. User clicks Okay.

6b. The dialog disappears and the Command box is no longer running commands.

4c. User tries to change the name.

5c. The name is read only.

See US: “Create a new Command Box” for additional exceptional flow for missing data

Postconditions:

1. The command box updates with the new settings, possibly moving to a new location in the process.

Title: Delete a Button

Preconditions:

- US: “Create a new Button”

Main Flow:

1. User right clicks a Button.

2. User clicks “Delete”.
3. The button disappears.

Exceptional Flow:

NOTHING

Postconditions:

1. The button is gone and all other buttons stretch to fill the lost space.

Title: Delete a Command Box

Preconditions:

- US: “Create a new Command Box”

Main Flow:

1. User right clicks a Command Box.
2. User clicks “Delete”.
3. The command box disappears.

Exceptional Flow:

NOTHING

Postconditions:

1. The Command Box is gone and other command boxes in the same area stretch to fill the lost space.

Title: Save Settings

Preconditions:

- User has started Code Workstation

Main Flow:

1. User clicks File->Save Settings.
2. A file selection dialog appears.
3. The user enters a file name.
4. User clicks Okay.

Exceptional Flow:

3b. User selects a protected file
4b. User clicks okay
5b. Dialog closes and nothing happens

Postconditions:

1. The settings file is saved with the selected name.

Title: Load Settings

Preconditions:

- US: “Save Settings”
- The User has changed some settings OR The user is on a different computer where they have not run Code workstation before.

Main Flow:

1. User clicks File->Load Settings.
2. A file selection dialog appears.
3. The user selects a previously saved settings file.
4. User clicks Okay.

Exceptional Flow:

3b. User selects a file that is NOT a saved settings file.
4b. The User clicks Okay
5b. The file select dialog will disappear and the program will throw a de-pickling error
6b. None of the window settings will change.

Postconditions:

1. The window configuration completely reverts to that saved state.

Load Settings:

Title: Run a command with a Button

Preconditions:

- US “Create a new Button”

Main Flow:

1. User clicks a button.

Exceptional Flow:

1b. The user clicks a button with an invalid command
2b. The corresponding error message will display in the selected command box and the program will continue as normal.

Postconditions:

1. The command’s output is displayed in the selected command box

Title: Run a command with a Command Box

Preconditions:

- US “Create a new Command Box” (+ US “Edit a Command Box [kinda])

Main Flow:

1. User Edits a command box.
2. User ticks the Auto Run check box.
3. User enters a command and an interval.
4. User clicks Okay.

Exceptional Flow:

3b. The user enters an invalid command.
4b. The user clicks Okay
5b. The command box populates with the corresponding error message every interval seconds

Postconditions:

1. The command box immediately populates with the output from the selected command. The command box updates every few seconds with fresh output from the command.

Title: Close and Open Code Workstation

Preconditions:

- User has launched Code Workstation and edited some settings

Main Flow:

1. User clicks the X at the top right part of Code Workstation.
2. The window closes and the settings are saved automatically.
3. User launches Code Workstation.

Exceptional Flow:

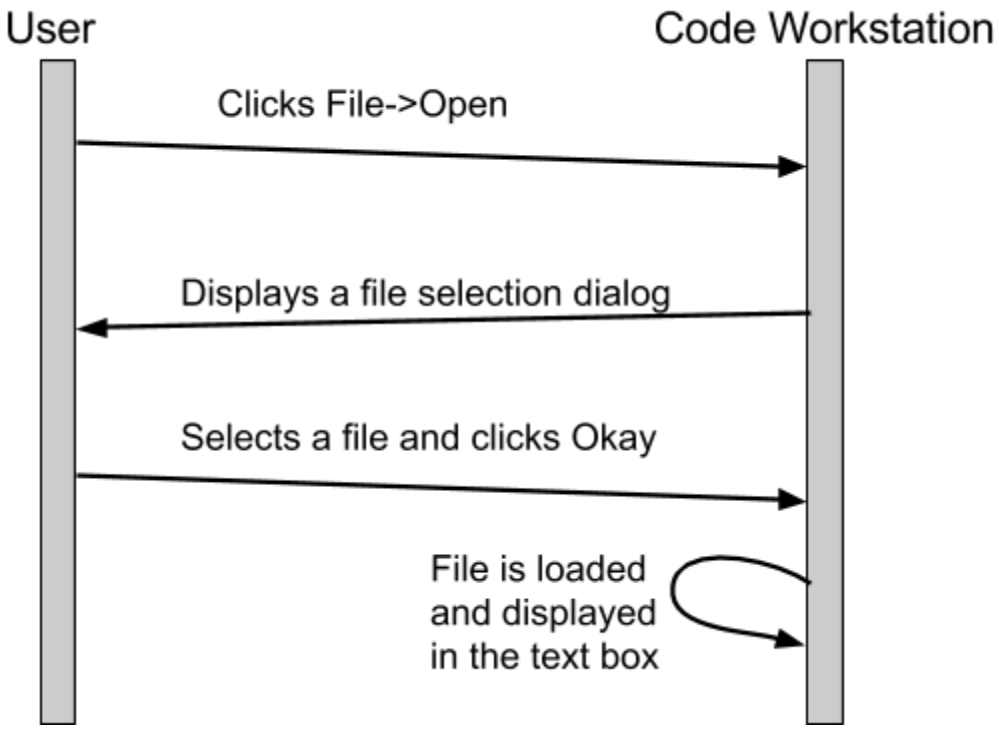
None currently

Postconditions:

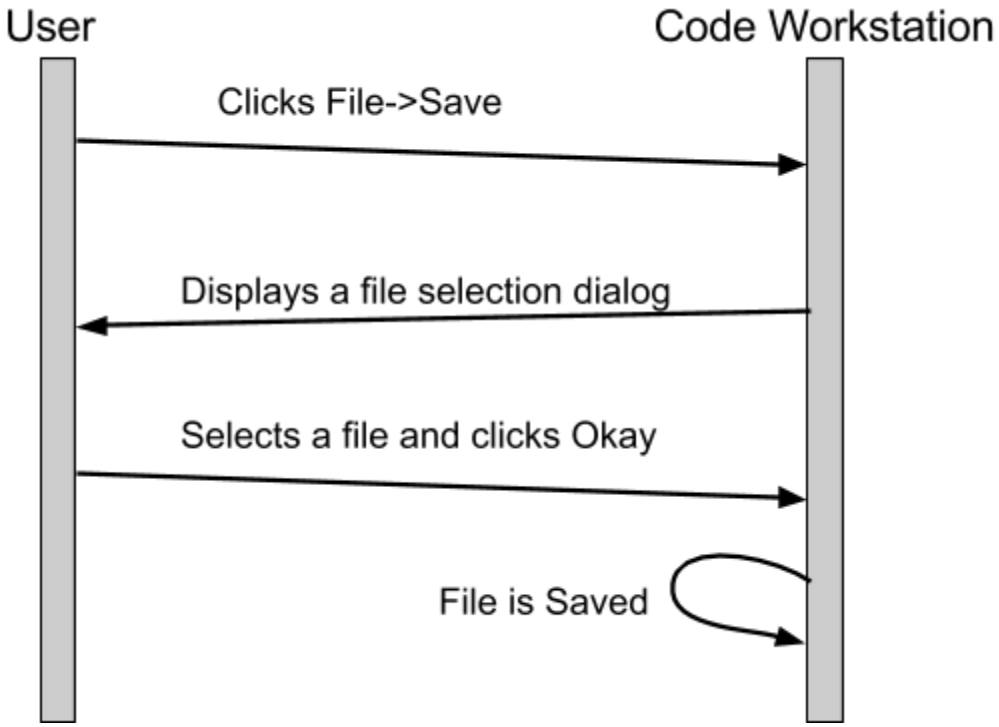
1. All the settings present before closing are available in the relaunched Code Workstation.

Sequence Diagrams

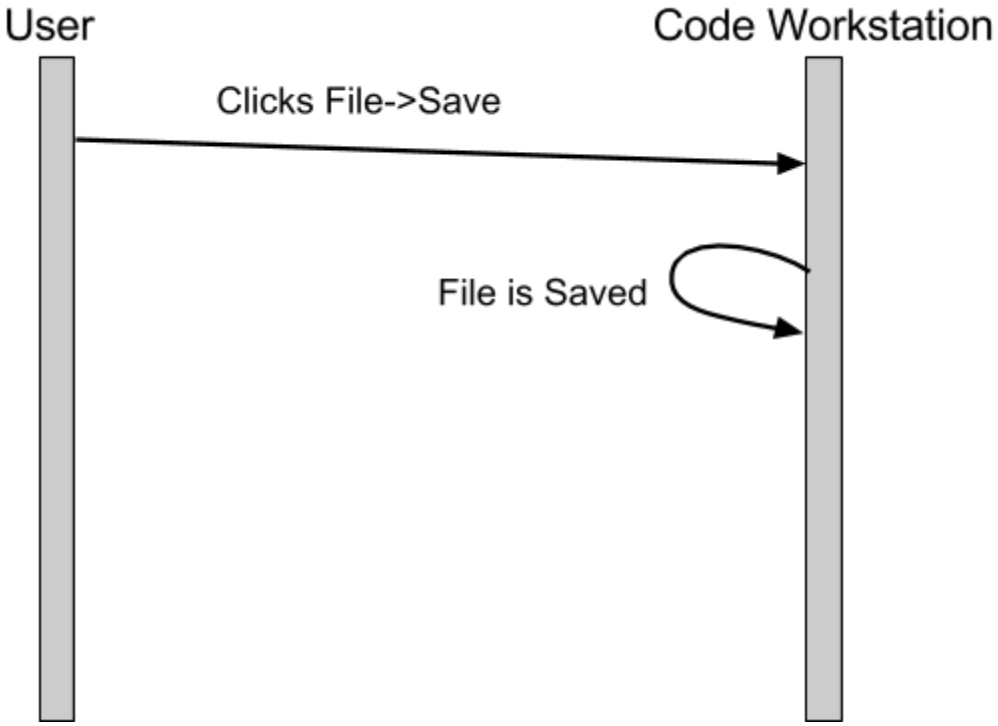
Title: Open a file



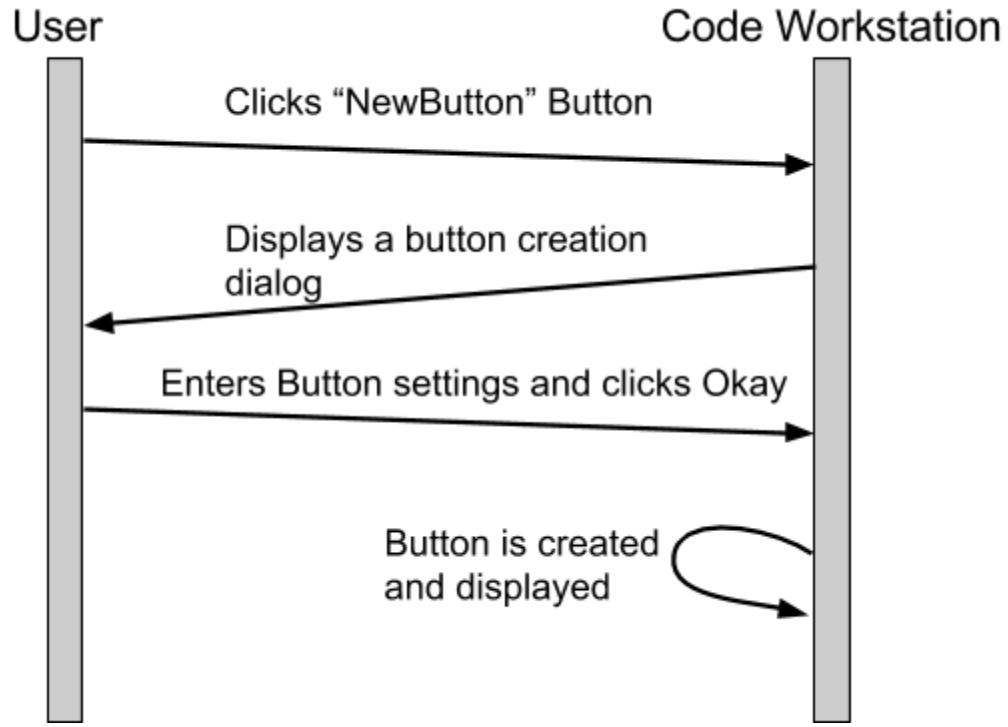
Title: Save a new file



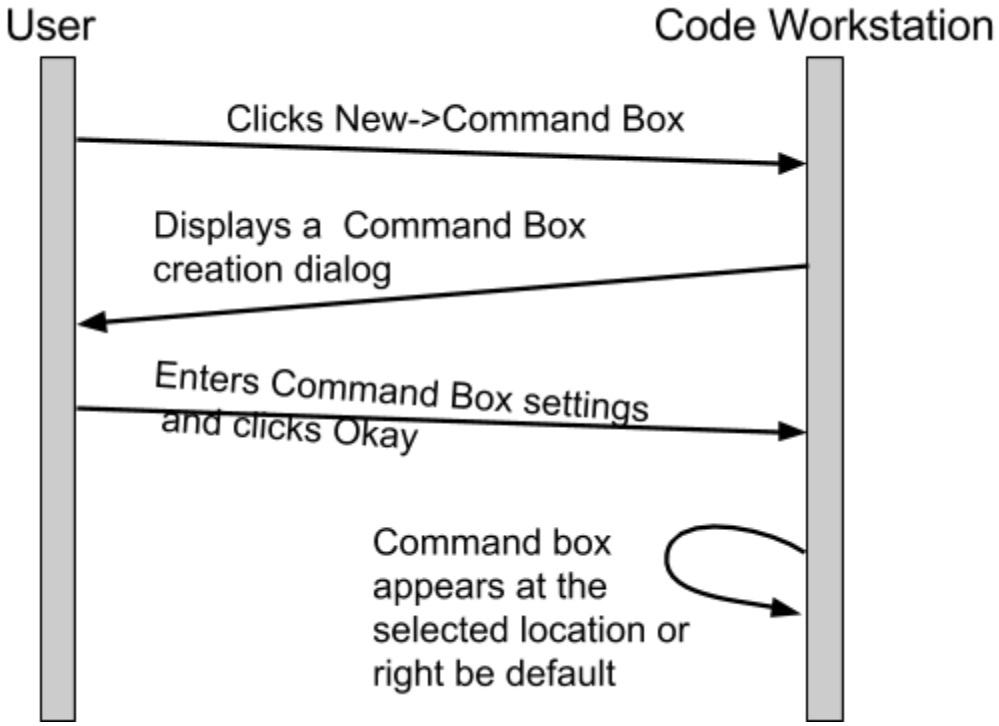
Title: Save a previously saved file



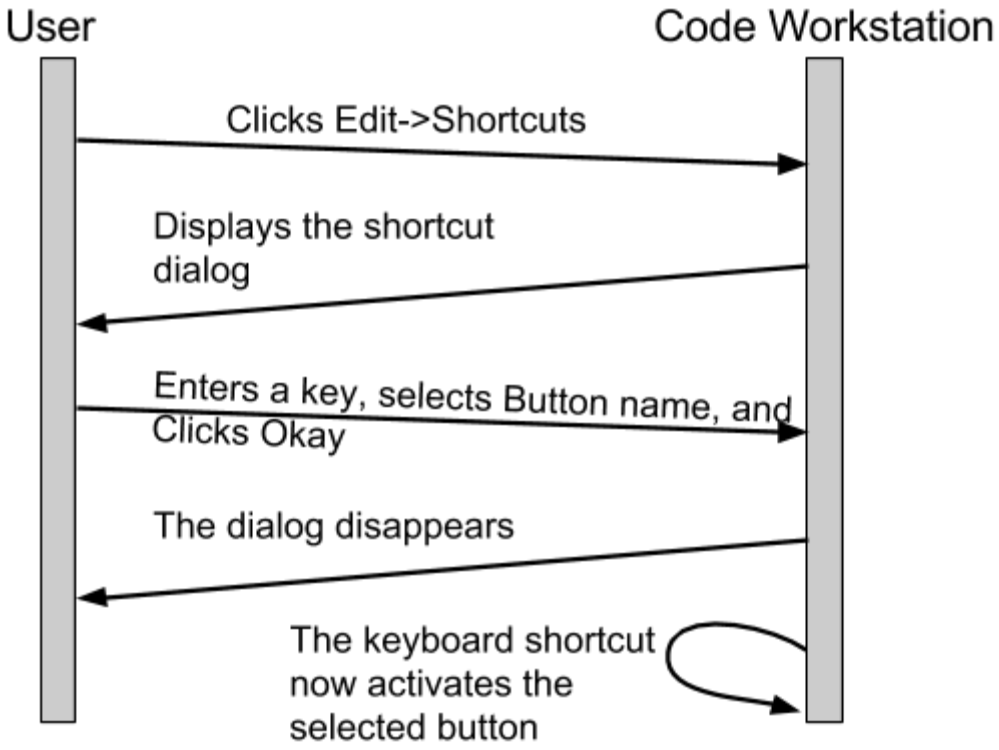
Title: Create a new Button



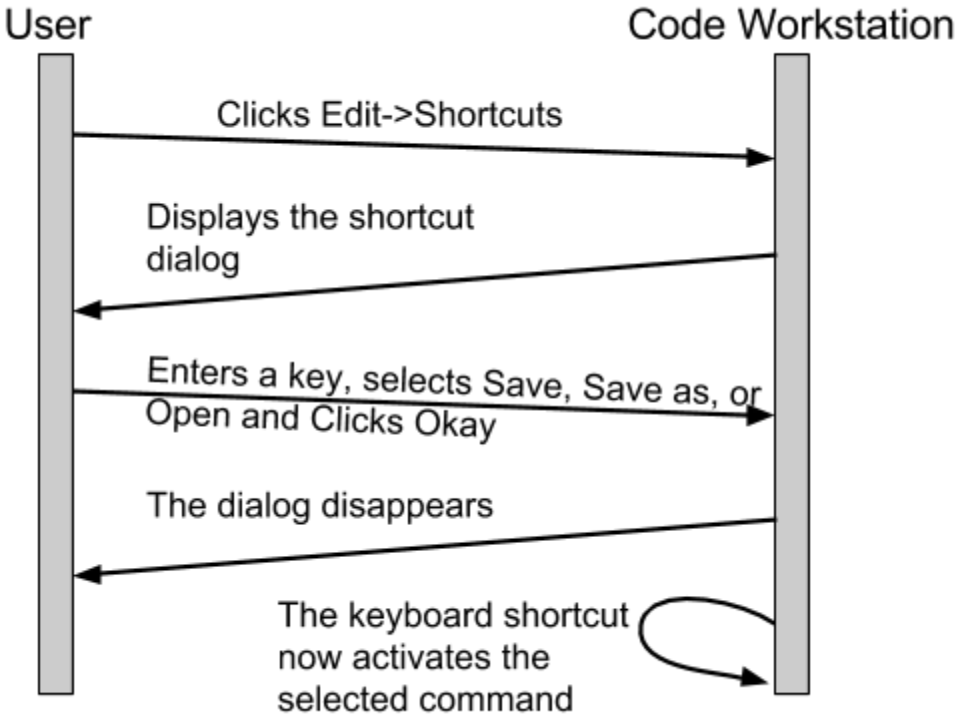
Title: Create a new Command Box



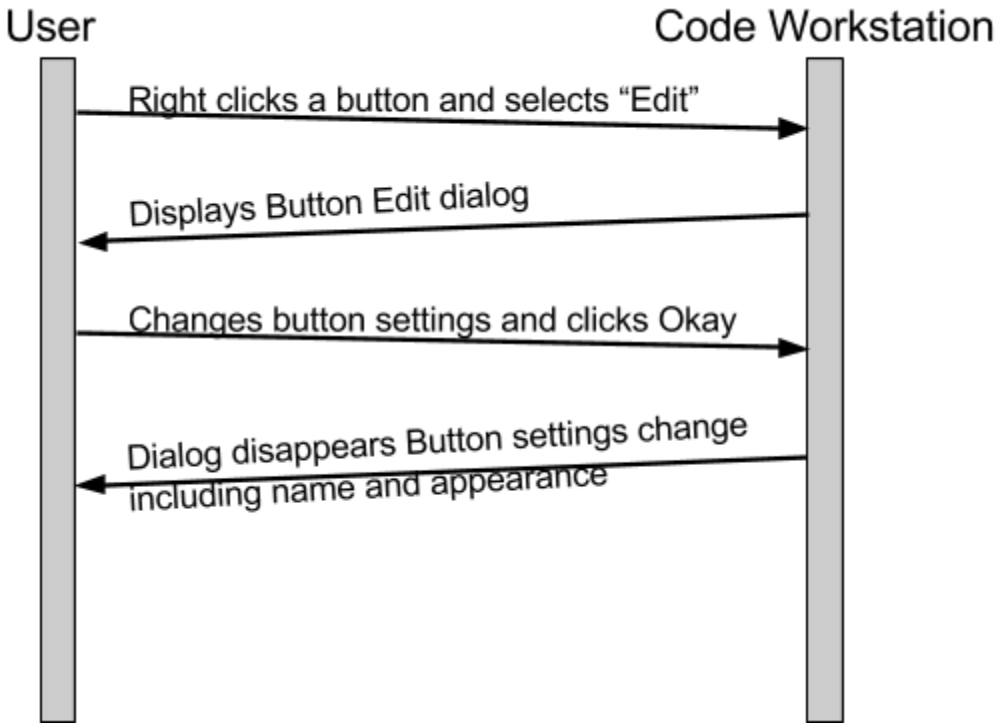
Title: Create a shortcut for a Button



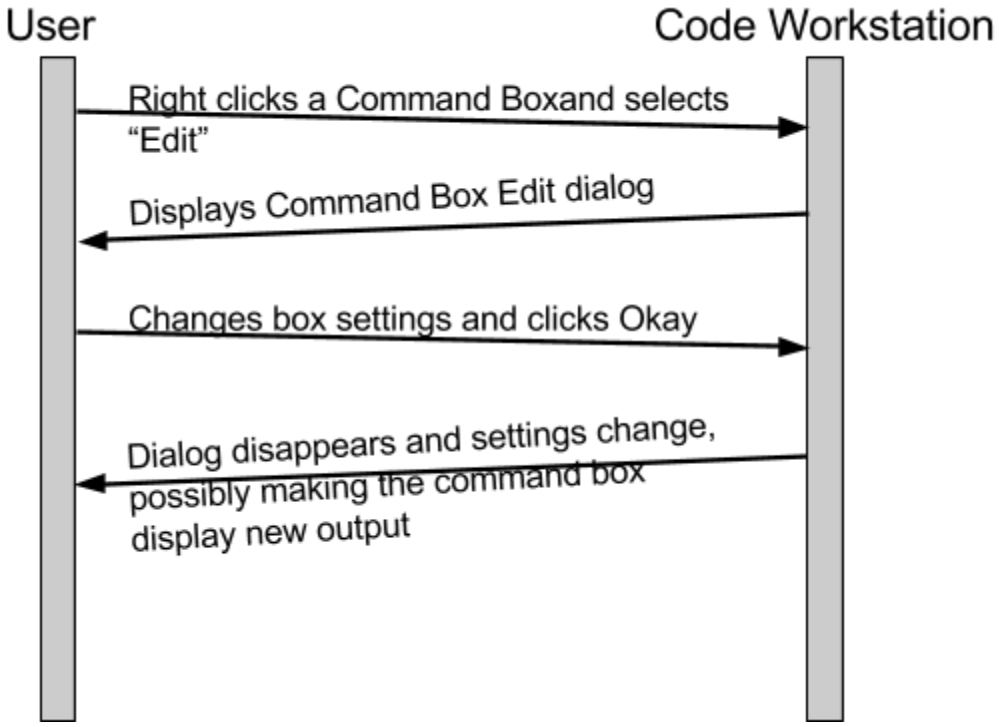
Title: Create a shortcut for Save, Save as, or Open



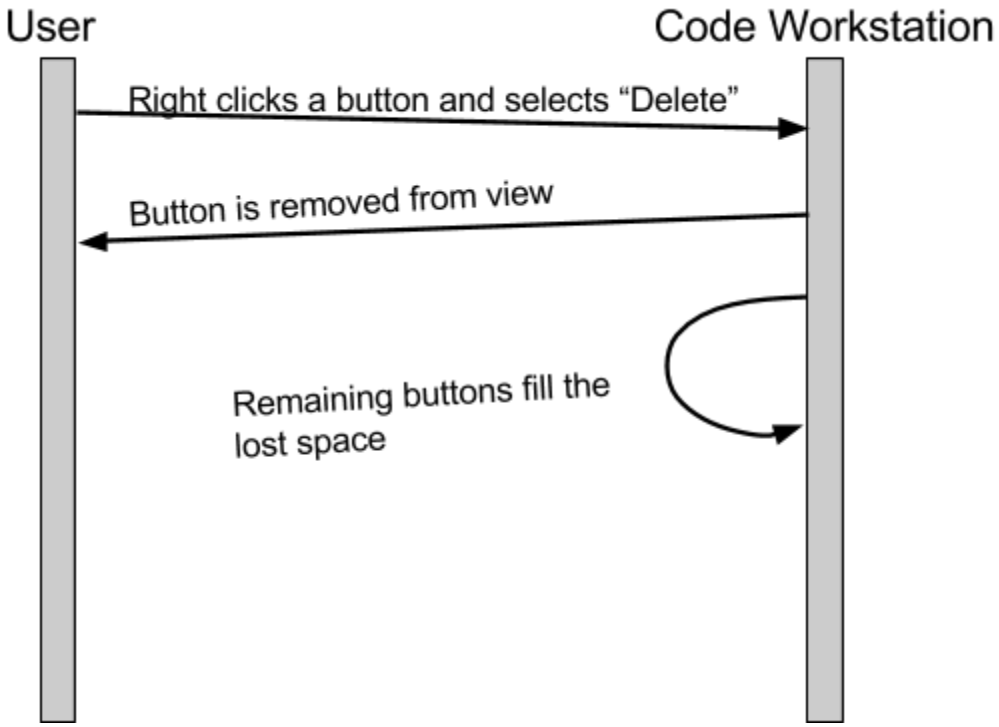
Title: Edit a Button



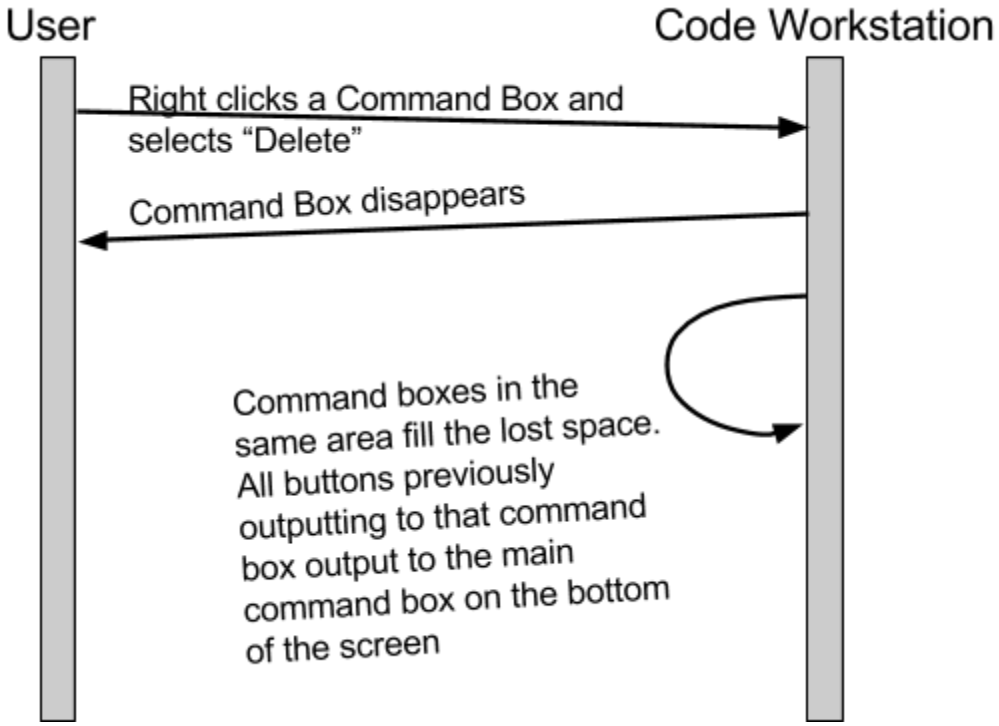
Title: Edit a Command Box



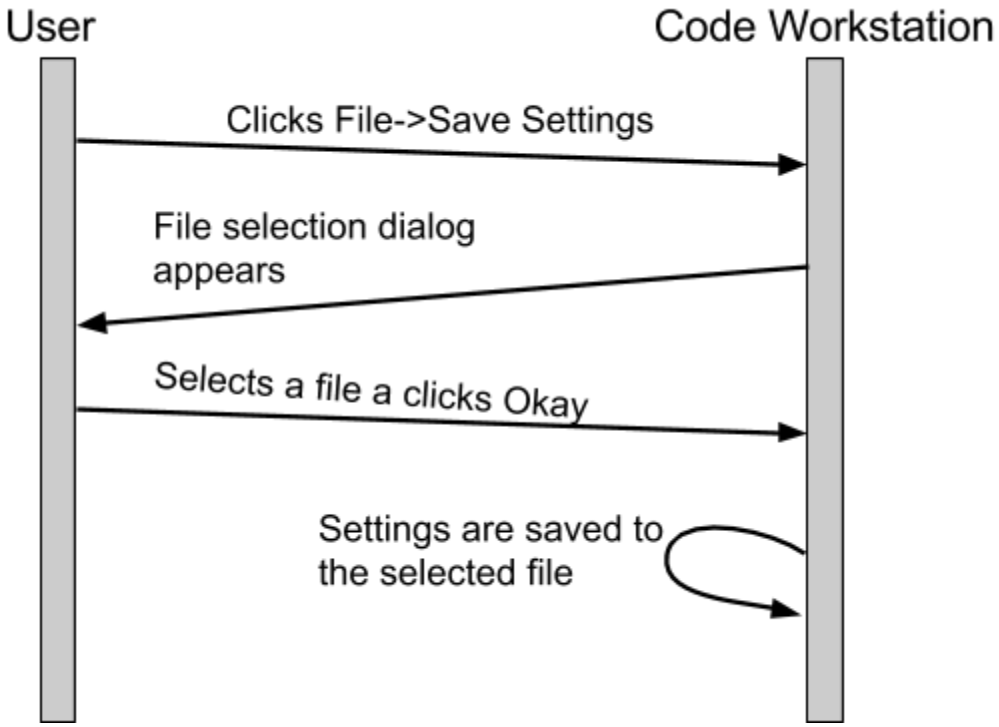
Title: Delete a Button



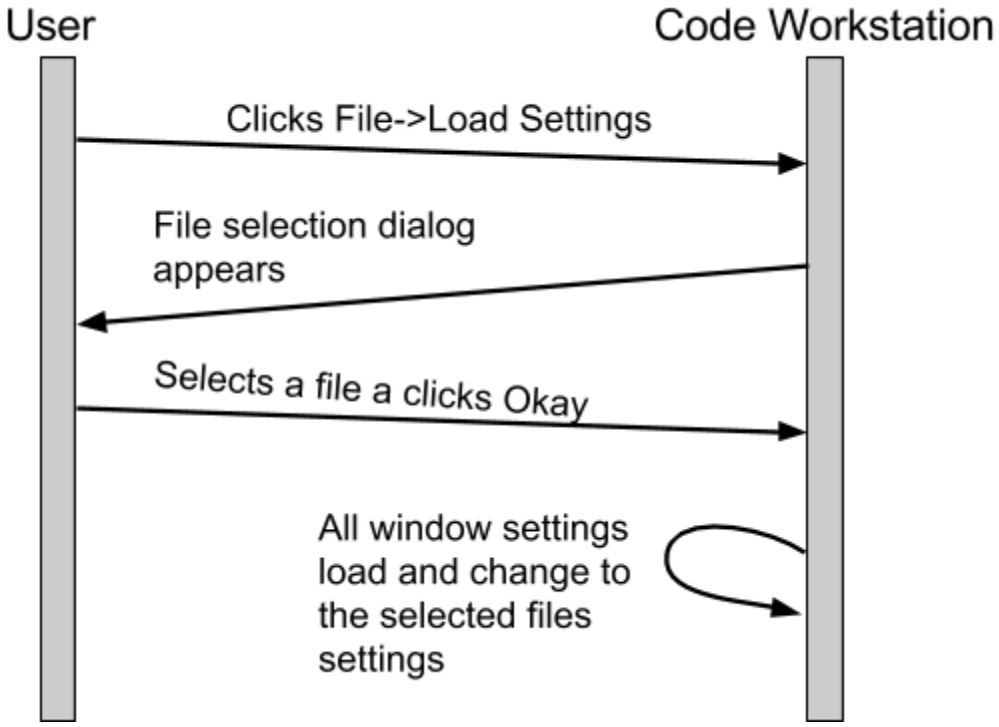
Title: Delete a Command Box



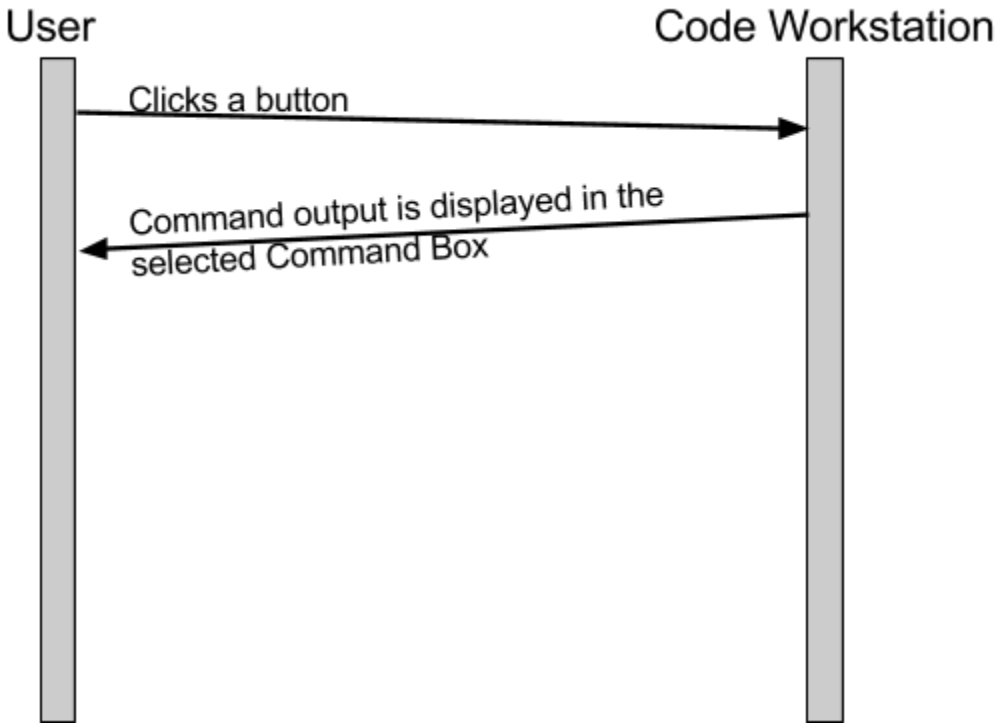
Title: Save Settings



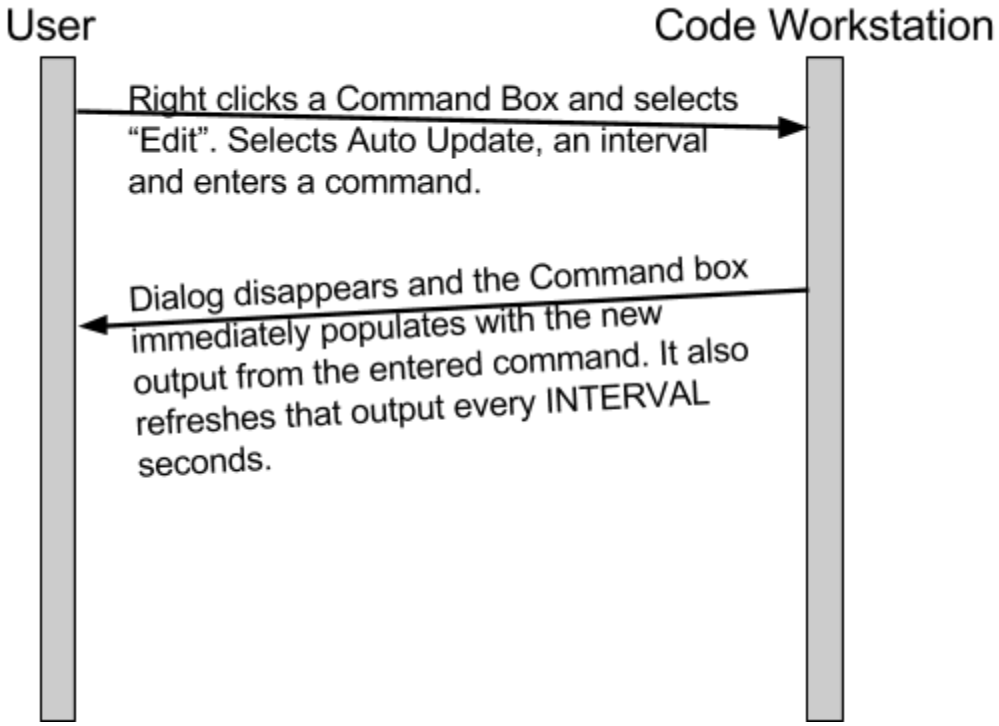
Title: Load Settings



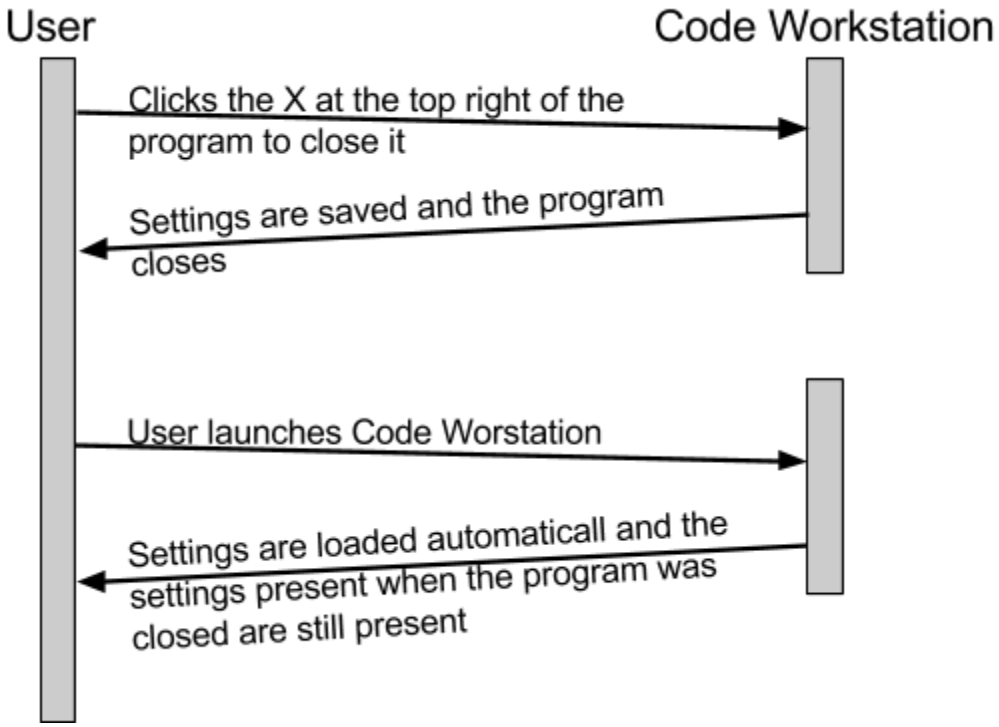
Title: Run a command with a Button



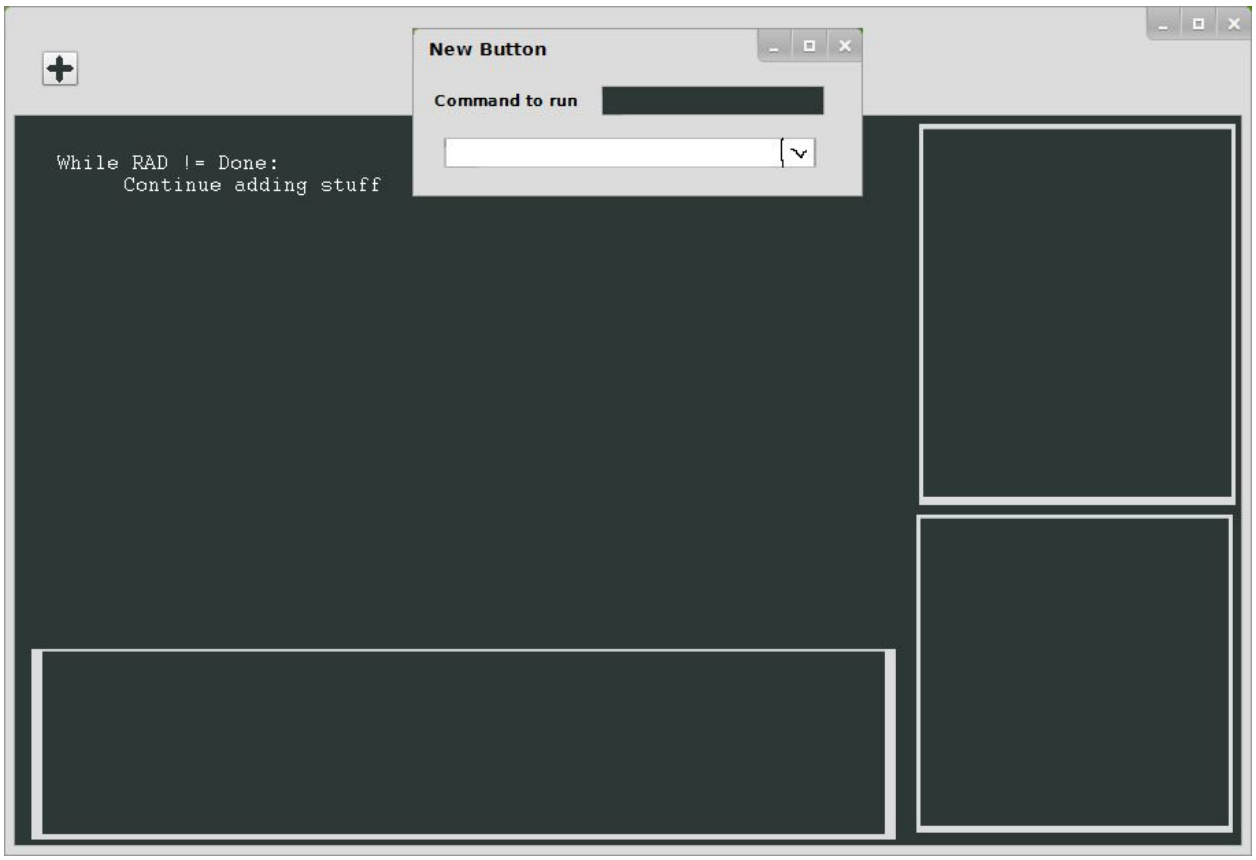
Title: Run a command with a Command Box



Title: Close and Open Code Workstation



User Interface Sketches



Glossary

Shell - A operating system interface or interface language
Console Command - A command run in a shell console
Linux - An open source operating system with many different distributions

References

Only self knowledge so far but might as well list some places relevant to the project (I will probably take these out in the final version unless they are wanted)

http://www.scintilla.org/SciTE.html
https://wiki.python.org/moin/PyQt
https://riverbankcomputing.com/software/pyqt/intro