|  |
| --- |
|  |
| Creator’s Toolbars |
| Capstone Proposal |

|  |
| --- |
| Taylor Garcia |

**Project Description**

This project will provide a collection of four toolbars that will overlay on top of the user’s screen, adding utilities that will be available no matter what programs they are working with underneath. Each toolbar will contain a set of tools meant to assist with a specific type of task or problem that developers and designers frequently face. The user will be able to minimize, expand, open, close, and move each of the toolbars around their screen according to their own preferences. The visual style of the application and each toolbar will be created from scratch, so that no part of it represents the default Windows window style.

**Target Users**

This application will be used by developers and designers. A developer, in this context, is anyone who works with general coding. This can mean a software programmer, a game engine developer, or someone who works with scripting. A designer is anyone who uses a computer to work with more visually-based projects, for example a web designer, graphic designer, or animator. This separation is made to emphasize the fact that this application is not meant to target solely those working with back end development or solely those working with front end development, but both general fields.

Additionally, this application does not target users of any specific technical skill level, but is meant to be intuitive, generalized, and user-friendly enough to be useful to a very wide range of experience levels. For example, it will have the basic tools that a beginning student could easily make use of, while also containing more general and advanced tools that even a very experienced industry worker can use throughout years of working in their field.

The application is also made to be used by those who are perhaps not actively designing or developing anything, but are instructing others in the techniques to do so. This includes teachers, instructors, and tutors from any education level. Visually demonstrating abstract or theoretical concepts is a frequent need in both development and design education, and quick access to tools for such will be made available with this project.

**Features List**

* **The launcher**
  + The launcher will be the core of the application and will be the one part that is consistently present when the application is executed. It will act as a sort of toolbox in that it holds the means to launch each toolbar. The way it will work is that it will start as a compact icon that can be clicked and expanded to show each individual toolbar's icon, which can then be double clicked to launch it. In addition, the launcher will provide access to application wide settings, including general application settings as well as the settings specific to each toolbar. To toggle the "stay on top" feature of the application's toolbars as well as the launcher icon, the user can right click the icon and select "Stay on top," the status of which will be denoted with a check mark. Minimizing the launcher will place it down in the task bar so that it is no longer on top of the screen. Closing the launcher, either through clicking "x" on the corner of the icon, or by right clicking and choosing "exit" will close the application, including any open toolbar.
* **Toolbars**
  + A toolbar, keeping to its name, is a set of tools presented together on a visual bar. Each toolbar will have tools that pertain to a specific type of task. One, for instance, would have tools that all deal with colors, and another would have tools that deal with measurement. While each toolbar provides utilities for a different subject, they will all share the following commonalities:
    - Unless the user has toggled this setting via the launcher icon, all toolbars will stay on top of the screen. This means that the user can switch windows and perform different operations within other application windows, and the toolbars will still be visible and immediately accessible.
    - All toolbars will have compressed and expanded states. When compressed, the toolbar will be made smaller and more discrete, with the toolbar's icon still visible to indicate which toolbar it is. This is to allow the user to move the toolbar out of the way in case they don't immediately need it, but do not wish to close the toolbar entirely and have to re-launch it. To continue using the toolbar's tools the user can then expand the toolbar and it will return to its full size with all tools visible and accessible.
    - All toolbars can be oriented vertically or horizontally. This helps the user when they wish for the toolbar toward the top or bottom of the screen versus the left or right sides of the screen.
    - The user can drag the toolbars to any part of the screen they wish. There will be predefined "snap to" slots in common places like the 4 sides of the screen as well, similar to the way Windows 7 allows you to drag a window to one side to have it take up half the screen.
    - Individual toolbars can each be closed, and can then be re-launched through the launcher at any time.
    - Most if not every toolbar will have its own list of settings that are specific to the tools it contains. An example of this would be a setting that allows the user to choose which unit type they wish measurements to be displayed and calculated with.
  + **Color Toolbar**
    - The Color Toolbar will have tools used for choosing, editing, and saving colors.
      * Color Preview
        + This is a simple box that displays the color currently being edited or viewed.
      * Color value display
        + This section will contain text boxes that display the color values for the color currently in the Color Preview. This includes red, green, blue, and alpha (RGBA) values, hue, saturation, and brightness (HSB) values, and the hexadecimal representation of the color. These text boxes can be edited directly to edit the color, or simply used to copy and paste the values to use elsewhere.
      * Eye dropper
        + This toggle-able tool will act as a color picker, allowing the user to click and sample any color currently on their screen. This color will then become the active color in the Color Preview.
      * Color picking
        + In this area there will be three different methods of choosing or changing an active color, which can be switched between using a small drop down.

RGBA sliders - contains sliders for red, green, blue, and alpha channels

Monochromatic color picker with a hue slider - allows the user to use the hue slider to choose a specific hue, and then chose a tint or shade of that hue

Full spectrum color picker - Presents every hue and gradients from the bottom with black, through their shades, then tints, to white, allowing the user to quickly pick a color without the use of any sliders.

* + - * Color swatches
        + Here there will be a collection of swatches, or squares each holding a specific color, which the user can drag the Color Preview box into to save the current color for quick access later. In addition to blank spots for new swatches, the application will also come with a set of preset swatches of commonly used colors. Clicking a swatch will make it the active color, and editing it will not edit the saved version, allowing the user to make multiple variations on a color while keeping the original. A swatch can be dragged out of the grid to delete it. The colors here will be saved over between closing and re-opening the application.
  + **Screen Drawing Toolbar**
    - The Screen Drawing Toolbar will contain tools that allow the user to draw anywhere on their screen. As they switch to other windows or do things in other applications, their drawing will remain overlaid in place, staying, like the toolbars, "on top" of the other windows.
      * Fill and Stroke color selectors
        + The fill color is the color that shapes will be filled with. The stroke color is the color of the outline of these shapes, as well as the color that will be used by lines, the eraser, and the marker tool. There will be a predefined set of colors to choose here, or, if the Color Toolbar is active, the user can choose to use the currently active color of the Color Toolbar.
      * Stroke weight slider
        + The weight of the stroke is essentially its thickness. This controls the thickness of shape outlines, of the path left by the marker tool, and the thickness of lines drawn.
      * Lines
        + This dropdown contains a variety of line-based objects the user can draw on the screen by selecting one, clicking the start point of the line, and then dragging it out. This includes:

Line - a basic line segment

Arrow - a basic line segment with an arrow on one end

Single quadrant coordinate plane lines - this creates a set of two lines at a right angle, which, depending on the direction the user drags it, can be made to represent any one of the four quadrants of a 2D coordinate system

Full 2D coordinate plane - this creates two perpendicular lines, one horizontal and one vertical, intersecting in the middle, representing a full 2D coordinate system, the length and width of which is determined by the user's dragging

Full 3D coordinate plane - this is the same as the full 2D coordinate plane, except that it adds a "z axis" line that cuts diagonally between the two original lines

* + - * Shapes
        + This dropdown includes a choice of several simple geometric shapes, as well as one freeform shape tool that lets the user draw out the outline, and fills it when they let go.
      * Text
        + Using this tool, the user can click anywhere on the screen and begin typing. Once they are done typing, the text will be rasterized to the screen and will act as anything else that's drawn, and can be drawn over or erased.
      * Eraser
        + This is a simple eraser tool can be used to erase anything that has been drawn.
      * Marker
        + Using this tool the user can draw freely on the screen by holding down the left mouse button, behaving like a simple brush in most painting or image programs.
      * Clear All
        + This button will clear the entire drawing from the screen.
      * Save Drawing
        + This button will save the current drawing as an image file that can be specified by the user in the settings. The user can also specify whether or not they want the screen behind the drawing to be saved out with it, or to have the drawing saved out with a plane white background.
  + **Screen Measurement Toolbar**
    - This toolbar will have tools that allow the user to more accurately visualize and measure things like screen space and pixel measurements. It will include settings that allow the user to choose what unit type they wish to use while using these tools, which will include pixels, points, inches, millimeters, and centimeters. Another setting will allow the user to choose what type of screen space setup they would like to use. This includes choosing an origin point (0,0) of any of the four corners of the screen, as well as choosing an origin point of the center of the screen and whether or not the minimum and maximum x and y values will use actual screen space measurements or will range from separate user-defined values.
      * Display Grid
        + This tool includes two textboxes for the user to specify the number of rows and columns they want, as well as an input for how many subdivisions. With these is a button to toggle the display of a grid that splits the screen with lines according to these numbers. When the grid is displayed, the user can choose the free transform tool to scale and move the grid to different parts of the screen. Another button will be available to reset the grid to take up the full space of the screen again.
      * Horizontal and Vertical rulers
        + These two toggle-able tools will allow the user to display or hide rulers that span the width and height of the screen, displaying measurements according the unit and screen space settings specified.
      * Mouse Coordinate Display
        + Toggling this will display or hide the x and y coordinates of the mouse as the user moves it across the screen, using the unit and screen space settings specified.
      * Box Selector
        + This tool is similar to the box selection tool in many image editing programs, with the sole purpose of displaying the height, width, and center position of the selected screen area, using the specified user settings.
  + **Conversion Calculator Toolbar**
    - This toolbar provides tools for quick unit conversion. It also includes a simple calculation tool.
      * ASCII converter
        + This tool allows the user to convert between characters and their ASCII values. It includes 4 small text boxes. The first represents the binary value of the character, the second the decimal value, the third the hexadecimal value, and the last one the actual character glyph. The user can type the character into the last text box and it will display the corresponding values in the other boxes. Alternatively they can enter a proper value into any other box and receive the corresponding character.
      * Unicode converter
        + This tool allows for conversion between a Unicode character and its code and decimal value, with a text box for each of these. The user can type a character (or paste, for those Unicode characters that cannot be produced with a typical keyboard) and have the code displayed, or enter a proper code and receive the corresponding Unicode character. The only limitation involves those codes that do not map to a displayable character, which this converter will not be useful for.
      * Number converter
        + This converter has 5 text boxes and allows for conversion between binary, octal, hexadecimal, and decimal numbers, each corresponding to a labeled text box. A value can be added or edited in any box and the other text boxes will update as corresponding conversions.
      * Basic Unit Converter
        + This tool acts as a typical general unit converter. The user can select a general unit category, like length or angles, from a drop down, and then select a “to” and “from” unit type from another drop down.
      * Basic Calculator
        + This tool is useful for very quick, very simple calculations like addition, subtraction, division, or multiplication. It is meant to help a user figure out quick mathematical operations and is not meant to replace a full, scientific calculator.
* **Settings and preferences**
  + Including the individual settings listed above for each toolbar, there will be a variety of settings that will allow the user to customize their use of the application. This will include a theme selection of either a light or dark theme. Additionally the user can select which toolbars will automatically be launched when they start the application.
* **Stretch goals**
  + An API documentation quick search toolbar
    - This toolbar would allow the user to select a specific API, then enter a keyword or phrase and search that API's documentation, which would return a small list of matching results.
  + A full scientific calculator
    - This would either add more advanced functionality to the conversion toolbar's calculator, or add an advanced calculator as a separate toolbar altogether.
  + Additional theme presets and customizable themes
  + Add a tool to the screen drawing toolbar that would allow the user to insert images, or paste them from the clipboard

**Technical Specifications**

I will be using Visual Studio 2012 to write, compile, and test the code for this project. The application and the necessary GUI aspects will be written using Windows Presentation Foundation from the .NET Framework, version 4.5. Additionally, for profiling and optimization purposes, I will be using the WPF Performance Suite. The application itself is meant to be installed and run as a standalone application. It will be created to run on any computer with a Windows Operating System of Windows Vista or later.

**Phase 1 Plan**

* Create and test the Launcher
  + Create a clickable icon that will expand to show toolbar icons
    - Must be dynamic to fit any number of toolbar icons
  + Create functionality to minimize the launcher and place it in the taskbar
* Create a basic working User Control template for Tools and Toolbars
  + Implement and test launching multiple instances of generic toolbars from the Launcher
  + Create and test the "stay on top" functionality
  + Create and test toolbar compressing and expanding functionality
  + Create and test generic "compact" tools that are small and fit within the toolbar
  + Create and test generic "expandable" tools that are larger and can be clicked to expand outside the toolbar

**Justification**

While programming, I often find myself in situations where I have a small problem I need to solve, or a concept I'm having trouble visualizing. The solutions to situations like these are theoretically very simple and very quick, but because I don't have any relevant utilities on hand I end up opening largely irrelevant programs, just to use some small feature of theirs. An example of this is trying to create a color pallet for the visual aspects of a program and having to open up Adobe Photoshop, just to use its color edit tool. Often I receive suggestions to download an application, plugin, or extension that specializes in solving that problem, but doing this consistently would only lead me to downloading a large number of applications, each only helping with a single specific issue, and each only ever being used once or twice. To make it worse, a lot of these applications come with a very hefty price tag for the very limited functionality they would give me access to.

This dilemma is what finally lead me to this project idea. I want to create a set of tools that will help with a wide variety of specific issues, as well as provide tools that are much more generalized and could be used on a much more daily basis. As someone who has a passion for visual design as well as coding, I made the decision to not gear these tools solely towards programmers, but to provide things that a designer could make use of as well.

When it comes to the technical depth of this project, a large part of it is obviously made up by the actual functionalities of the application and its different tools. My past experience in creating and working with a GUI, specifically in WPF, is limited to very basic, single window applications, with few if any complicated mechanisms behind them. Some of the techniques that I will be required to learn and implement to make my planned features work are the use of multiple windows in a single application, including the launching and closing of these windows on an individual basis; mechanisms to use and toggle "stay on top" behaviors; the several mechanisms required to work with letting the user draw on the screen and still manipulate and navigate the windows behind the drawing; the "taking over" of cursor functionality for different tools, and the ability to return the cursor to its normal state again.

While core functionality is clearly very important, there is an additional property of the application that will require heavy technical focus. After graduating, my goal is to work in the video game industry. One of the major features of game development that really interests me is the user interface side of a game, specifically things like menus, heads-up displays, and player inventory systems. As I have experienced, many of these mechanisms can be very clumsy and unintuitive, while others are extremely easy to learn and adapt to. Some developers may see this as a lower priority aspect of a game compared to its overall gameplay, but I've found that a game's overall UI can either contribute to an enjoyable playing experience, or drop it to a frustrating stutter.

Keeping with this kind of a mindset, I've recognized that a toolset like this, especially as it will be frequently used alongside other heavier processes, can quickly become useless if it is not as user friendly as possible and optimized to run efficiently. In my education, a real focus on this kind of user experience standpoint has often been sacrificed for a focus on functionality due to time constraints. As such, there are several things that I will be learning on my own for this project, including the editing, persisting, and implementing of user defined settings and presets; the creation and implementation of several style templates and quick switching between them; the building of a project for actual production, including the providing of easy download and install utilities; thorough memory and time optimizations of an application, and the use of window animations for smooth, aesthetically pleasing changes and transitions.

Overall I believe that these two major aspects of the application are what make this project Capstone worthy. I will be learning and implementing a wide variety of techniques and systems to create the functionality I want, as well as a variety of mechanisms to give the application the quality UI and UX that it needs.