GEOG 8990, Spring 2018, Tobias Fimpel

Summary/Reflection Write-up 3

The objective of developing this ArcGIS Pro Configuration was to gain a high-level understanding of what ArcGIS Pro Configurations are and how they work. Going into this development activity I did not have a clear vision for what my configuration should accomplish. After spending a considerable amount of time learning about configurations (videos, online tutorials, etc.) and experimenting with configurations hands-on I arrived at a product that focuses the user interface to a very specific task: buffering US Interstate Highways. (I don’t have a real purpose to do so in mind, it is just an example.) Focusing the user interface towards a specific task is a common purpose for configurations. Another common purpose is branding, and I did to some extent add a custom look to the application.

I took this development activity as an opportunity for understanding how the ArcGIS Pro SDK relates to the geoprocessing framework and tools, which are prevalent throughout the ArcGIS ecosystem. The custom ribbon included in this configuration opens and executes the “buffer” and the “truncate table” geoprocessing tools. I also incorporated the use of an ArcGIS Pro Project Template file (.aptx) because I wanted to understand the features and limitations of .aptx files.

What I did not want to do while developing this configuration is getting hung up on specific ideas that happened to turn out much more challenging than anticipated – there are numerous things I attempted and worked hard on but for the moment decided to give up on in the interest of understanding the breadth of what ArcGIS configurations and add-ins can accomplish more easily. I am happy that I ended up with a unique, functional product which has come quite a long way from the template I started out with.

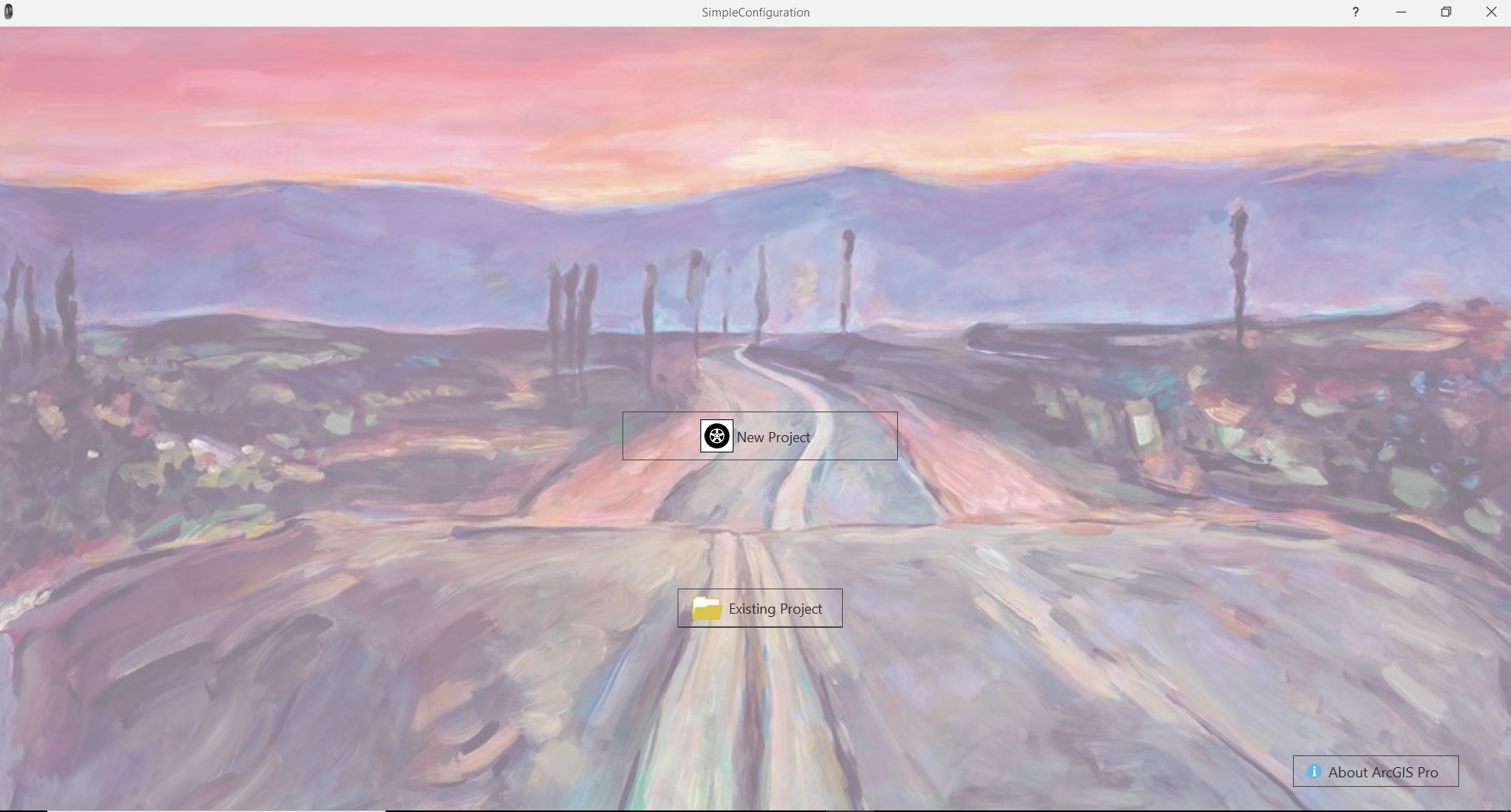
The following pages show screenshots and key pieces of code. The complete Configuration code is available at <https://github.com/TFimpel/ArcGIS-Pro-February-Sandbox-Configuration>.

----------------------------------------------------------------------------------------------------------------------------------------------------------------

1. A good way to make a configuration available to users is via a shortcut. For example, for this configuration I created a shortcut with

Target = "C:\Program Files\ArcGIS\Pro\bin\ArcGISPro.exe" /config:SimpleConfiguration

Following a splash screen, the application shows the start page. The start page is a .xaml document. In my case it holds three buttons. I will focus on the “New Project” button since the other two buttons (“Existing Project” and “About ArcGIS Pro”) weren’t added by myself but were part of the template code I started out with.



1. The “New Project” button is bound to a command “NewProjectCommand” that creates and opens a new project based on a .aptx project template file. The template is set up with a map that includes specific layers.

public ICommand NewProjectCommand

{

get

{

CreateProjectSettings projectSettings = new CreateProjectSettings()

{

//Sets the name, tempalte, and location of the project that will be created

Name = s,

TemplatePath = "C:\\Users\\fimpe\\Documents\\ArcGIS\\ProjectTemplates\\MyTemplate4.aptx",

LocationPath = "C:\\Users\\fimpe\\OneDrive\\MGIS\\GEOG 8990 Spring 2018\\sondbox February 2"

};

if (\_newP == null)

\_newP = new RelayCommand(async () => await Project.CreateAsync(projectSettings));

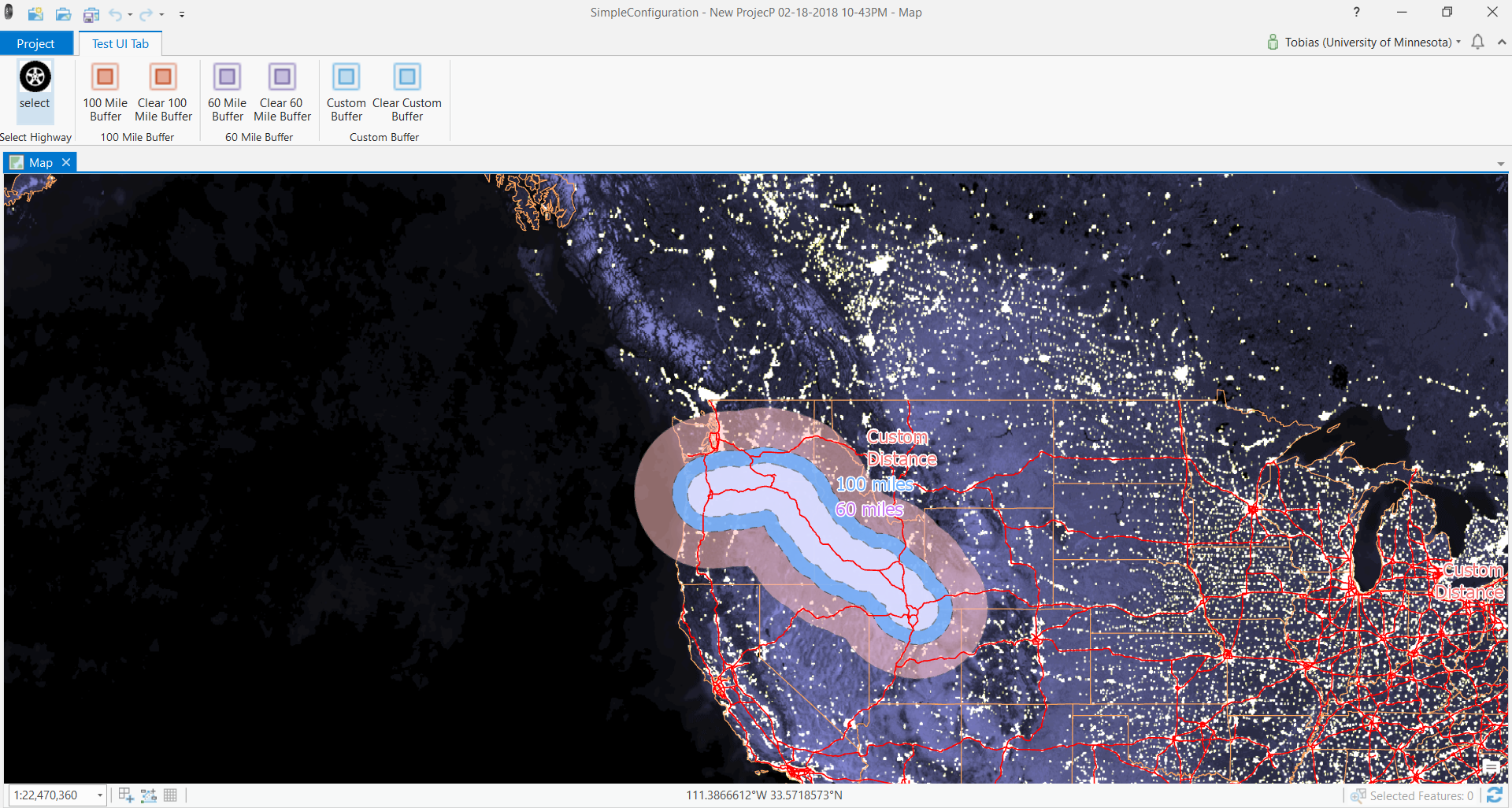
return \_newP;

}

}

The new project’s user interface is simplified by an override of the OnUpdateDatabase method that iterates over all tab elements in the UI and removes them (except for the one custom tab I created).

The custom tab is essentially an Add-In. The Config.daml file defines the content and the layout (1 tab, 4 groups, 3 of them containing 2 buttons and 1 of them containing 1 button).



1. The leftmost button is a “Select” button. It’s onClick event does two things:
   1. Sets the “US Interstate Highways” layer to be the only selectable layer
   2. Activate the ArcGIS Pro out-of-the-box “Select by Rectangle” tool

await FrameworkApplication.SetCurrentToolAsync("esri\_mapping\_selectByRectangleTool");

The onClick events of the other six buttons are set to either execute or open two geoprocessing tools (depending on which button is clicked, different input parameters are used)

For example, the “100 Mile Buffer” button:

[…]

var inputlayer = (MapView.Active.Map.Layers.First(layer => true) as FeatureLayer);

string output\_polys = System.IO.Path.Combine(System.IO.Path.Combine( ArcGIS.Desktop.Core.Project.Current.HomeFolderPath, "bufferOutPuts.gdb"), "buffer\_60miles");

string buffer\_dist = "60 Miles";

string line\_side = "FULL";

string line\_end\_type = "ROUND";

string dissolve\_option = "ALL";

string dissolve\_field = "";

string method = "GEODESIC";

var param\_values = Geoprocessing.MakeValueArray(inputlayer, output\_polys, buffer\_dist, line\_side, line\_end\_type, dissolve\_option, dissolve\_field, method);

var result = await Geoprocessing.ExecuteToolAsync("analysis.Buffer", param\_values);

[…]

And the “Clear 100 Mile Buffer” button:

protected override async void OnClick()

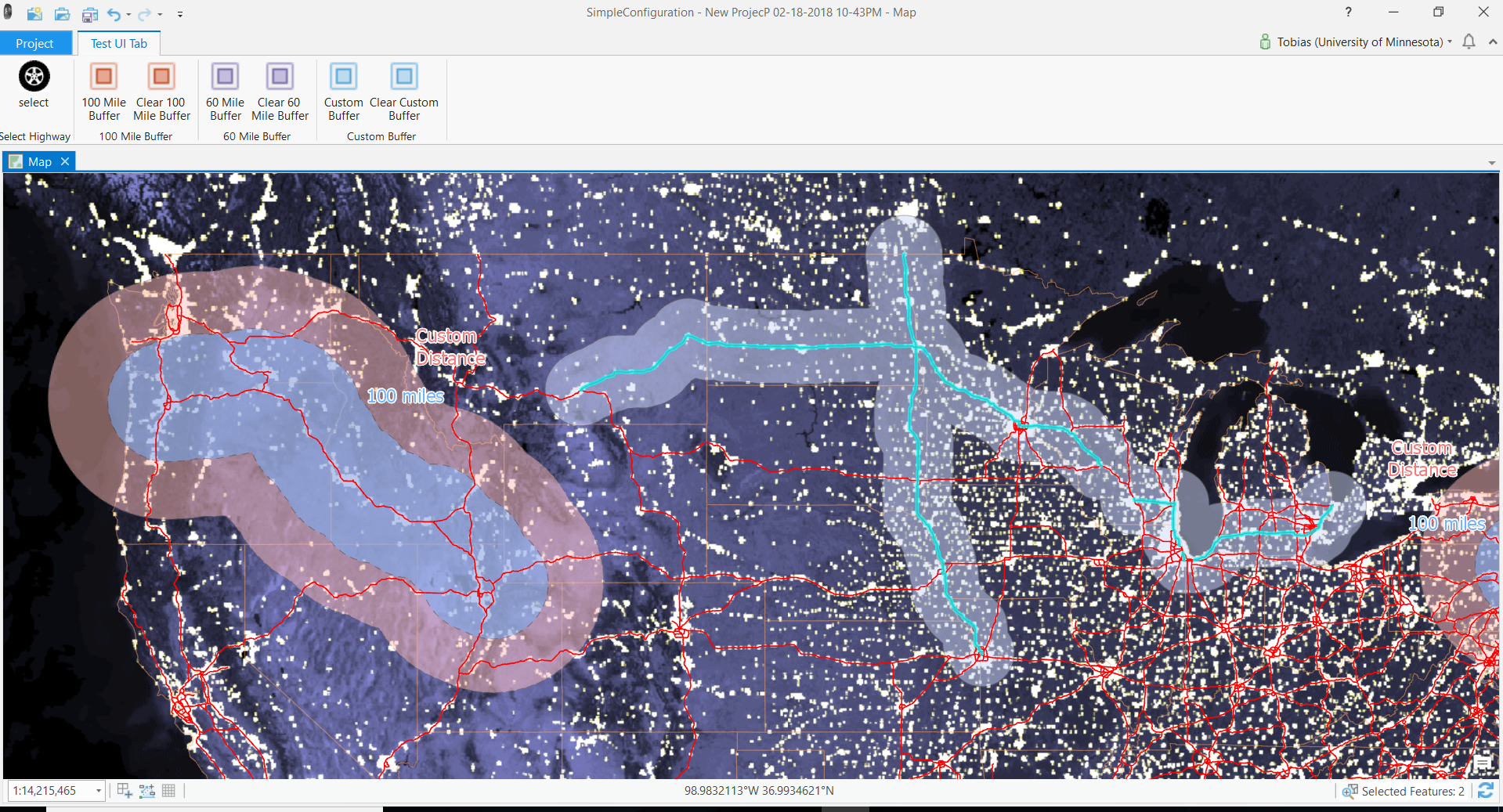
{

string output\_polys = System.IO.Path.Combine(System.IO.Path.Combine(ArcGIS.Desktop.Core.Project.Current.HomeFolderPath, "bufferOutPuts.gdb"), "buffer\_100miles"); // @"C:\data\ca\_ozone.gdb\ozone\_buff";

var param\_values = Geoprocessing.MakeValueArray(output\_polys);

var result = await Geoprocessing.ExecuteToolAsync("TruncateTable\_management", param\_values);

}



1. The “Custom Buffer” button opens the buffer geoprocessing tool with the Input Features and Output Feature Class parameters filled out, and leaves it up to specify the remaining parameters and run the tool.

Geoprocessing.OpenToolDialog("analysis.Buffer", param\_values);

