Hi there! If you are reading this, it is most likely because you or your team members are working on Stemming From Disaster, and have been tasked with creating new content for the simulation. This document will give you a basic rundown of everything you will need to add to make your own disaster. Before we begin, there are some basic concepts you should know about / understand:

- Unreal Engine 4 / 5's layout
- Blueprints
- C++
- Static Meshes

With that, let's get started!

**HOW TO CREATE A NEW DISASTER**

1. In Unreal Engine, open the start map in the editor.
2. Place a new start button in the map, next to the already existing buttons.
3. Click on the start button, and fill in the following info in the details panel:
    - Simulation Name (With the name of your simulation)
    - Map Name (Where you are going to travel after the button is clicked, probably a variant of the airplane map)
    - Text (With the name of the simulation, as it will appear on the button)
4. Open Visual Studio, and open the Game Instance class (SD_GameInstance.cpp).
5. In start simulation, after the variables have been reset, add an if statement **if (dis ==** Simulation Name**)**, and then put empty brackets after the statement. These will be filled in later.

This should be all you need to do to get a new simulation ready

**HOW TO CREATE A NEW MAP**

1. To create a new map, it is best to duplicate an existing regular map, as the lighting and atmosphere has been created. After duplicating the map, delete all of the static meshes in the map, excluding a floor of some kind and the narrator. Rename your map to what it will represent.
2. Open the level blueprint. Scroll to the right where the **Setup Map** function is called. Change the parameter to something relevant to the map.
3. Open Visual Studio, and open the Game Instance class (SD_GameInstance.cpp).
4. Make the following changes to the game instance:
    - Add your map to the list of travelable maps by adding **travelableMaps.Add(**Your Map Name**);** underneath the if statement that was created for your disaster in the **startSimulation** function
    - Set the spawnLoc vector to where the player character spawns in your level. Most likely, it will be **FVector(0.0f, 0.0f, 6.25f);**. If this is the case, or the parameter you called SetupMap with in Blueprints with the other map names that have a spawnLoc of 0.0f, 0.0f, 6.25f. This is used for the fade out cube.
    - In the **mapNameLookup** function, add **else if(name ==** your map name**) return TEXT(**Human readable map name**);** to the list of if statements. This is used to

transform your map name into a readable map name, with spaces and proper formatting. This would be a switch statement, but unfortunately you cannot switch on strings.

This should cover the creation of a new empty map. Now it needs to be populated with data.

**HOW TO POPULATE A NEW MAP**

1. To create investments, open Visual Studios and Open InvestData.cpp. Inside, you will need to edit 2 functions:
    - In **createInvestmentData**, check if the map name is equal to the parameter you called **Setup Map** with in blueprints. If so, add investment data by writing **retMap.Add(TPair<FString, int>(**investment item, cost**));**. In this map, the string represents the name of the item being invested in, and the int represents how much it costs.
    - In **createCareerData**, check if the map name is equal to the parameter you called **Setup Map** with in blueprints. If so, add an investment career by writing **retMap.Add(TPair<FString, FString>(**investment item, career name**));**. In this map, the first string represents the name of the item being invested in, and the second string represents what career is relevant. You will need to use the same item that you created in **createInvestmentData** for an investment to be correlated to a career.
2. To create dialogue, open ExplainData.cpp. You will need to edit 1 function:
    - In **createExplainData**, check if the map name is equal to the parameter you called **Setup Map** with in blueprints. If so, add a dialogue option by writing **redArr.Add(TPair<FString, FString>(**Dialogue Name, Dialogue Code**));**. The dialogue name is the name of a dialogue option as it appears on the button, while the dialogue code is the name of the sound cue that will be loaded and played by the narrator.
    - To create a sound cue, import a wav sound file into the VoiceClips folder in the Unreal Editor. Right click on the voice clip, and select **Create Cue**. Then name the new sound cue according to this convention:
        - If playing this sound clip will not result in the creation of any new voice clips, name it **F_**Sound Clip Name.
        - If playing this sound clip will result in the creation of a new voice clip, name it **T_**3 Digit Code Sound Clip Name. For Example, **T_000test** is a valid name.Then, in the narrator class, go to the **dialogueLookup** function, and add your new clip to the switch statement. Call the **generate dialogue** function with the first parameter being the Dialogue name you are going to create, and the second parameter being the name of the sound cue you will load.
3. To create interactables, open the map in Unreal Editor. Go to C++ Classes / StemmingTheDisaster / Public (Note that the project was created as Stemming The Disaster instead of Stemming From Disaster due to a communication error). Click and drag the **Interactable** class into the world. You will then need to make the following changes to it in the details panel:

- Attach a static mesh to the interactable.
- Set the name of the interactable.
- Set the description of the interactable. Note, you can make a new line in the text field by hitting shift and enter.
- Add relevant careers to the interactable.

With that, you should know how to do everything to create new maps and disasters inside of Stemming From Disaster. Best of luck on your project!

-David, Original Team Lead