The `Code` directory is structured as follows:

```
Code
|   Data
|   |   SST01.txt
|   |   SST02.txt
|   |   .
|   |   .
|   |   SST09.txt
|   Images
|   |   1.png
|   |   2.png
|   |   .
|   |   .
|   |   9.png
|   SurfaceTemperature.py
|   output.gif
|   README.pdf
```

The `Data` directory should have data with the extension `.txt` and the temporal ordering should be same as the alphabetical ordering for the animation to be correct. The rendered images can be found in the `Images` directory, but the script can be run even if the directory is empty, but the directory needs to be present for the script to execute.

The `SurfaceTemperature.py` file is arranged as follows:

```
# Imports
import ...
# Gets the minimum of maximum of all the Latitudes from a file
def get_mins(filepath:str):
    ...
# Saves the data for each latitude
def process(filepath:str, min_of_max):
    ...
# Generates plots for each timestep and saves them as images
for i, data in enumerate(data_lst):
    ...
# Compile all images together to create a gif
def create_gif(image_paths, output_path, duration=250):
    ...
```

The maximum value for each of the latitudes is calculated during the parsing of each of the files by detecting `'** line too long **'` at each line. Using the list of maximums, the minimum of maximum is calculated and stored in `min_of_max` which is used for termination while storing the data.

Next, the extracted latitudes, longitudes and values are casted to create a NumPy array. All the values equal to -1x10^{34} are replaced with -1x10^{10}. Cartopy is used to create the global map and coastlines to get a projection. This projection is then used to get the contour plot by giving the latitude, longitude, values and the contour levels with a dpi of 600.

Finally, all the images are loaded using the Python Image Library and a GIF is created and saved called `output.gif` . This file can also be found in the Images folder.