Michela Proietti 1739846

EXTRA CREDIT → ADAPTIVE RENDERING

- I have changed the file yocto_pathtrace.h adding some structures that I needed in order to implement adaptive rendering and I also added some additional fields in pathtrace_state and two functions, get_actual _progress and get_max_progress, that I used to get the parameters for the print_progress function.
- I have modified the following functions in yocto_pathtrace.cpp:
  - init_state, in which I added the initialization of the new fields I added in pathtrace_state.
  - render_samples, in which I added a boolean variable that says whether or not we want to do adaptive rendering and therefore I added an if-else to handle the two cases. In render_samples, we trace an initial number of samples, then we start a loop in which we look for all the pixels that are below the desired quality and we call the function trace_until_quality, and then we look for the pixels that are near pixels that have been sampled by quality using the indices obtained with create_sample_spread and then we call the function trace_by_budget on those pixels. Finally, we do some updates and restart the loop, until checkEnd becomes true.
- Then, several functions have been added in yocto_pathtrace.cpp:
  - checkEnd, that is used to understand if we have reached a point in which we need to terminate the rendering. In particular, it checks whether we have reached the desired number of samples per pixel, the desired rendering time or the desired quality of the image.
  - trace_sample that, as the name suggests, is used to render new samples and it is called by other functions. There is a cycle in which a specified number of samples is traced. At the end of the cycle, we compute the per pixel error and based on that we update the quality of that pixel.
  - Create_sample_spread, that gives the indices of the pixels that are close to the pixels that have been sampled by quality.
  - Trace_until_quality, that traces new samples for the pixel we are considering, until the pixel quality becomes greater or equal to the quality that we give as parameter or until chechEnd becomes true.
  - Trace_by_budget, that calls trace_sample in order to render a number of samples equal to the parameter sample_budget for the pixel we are considering.
- Finally, I have added two new executables ypathtrace_adp.cpp and yipathtraces_adp.cpp

In order to test adaptive rendering, you need to set the boolean variable adp_rendering in render_samples to true. Moreover, when you run the executable, you can specify the desired quality by adding *-q <desired quality>*, if you don't the default desired quality is set to 5. I used two test images and I rendered 6 different versions for each of them, starting from quality 0 until quality 5, because the rendering time becomes very high. However even with the 6 versions of the images I rendered, it is possible to see how the quality of the image increases each time.