Luca Polenta 1794787 – Adaptive Rendering

Changed files:

- yipathtraces.cpp and ypathtrace.cpp: In order to make it run interactively, I've added some option to parse new input from command line: trace by quality using "-q" or "--quality", trace by spp (sample per pixel) using "--spp" and trace by time (seconds) using "--seconds". Furthermore, I've moved the iteration over the samples into the yocto_pathtrace.cpp file.
- yocto_pathtrace.h: I've added some parameters into the pathtrace_state and into the pathtrace_params in order to support the adaptive rendering and I've added some get_functions in order to obtain some information about them during the execution.
- yocto_pathtrace.cpp: Here I've added and changed various functions:
    - init_state: It has been modified in order to initialize all the parameters of state.
    - checkEnd: check if I have reached a default condition or one of those expressed in the command line input to stop adaptive rendering.
    - trace_sample: it traces a block of samples. It accumulates the information about the pixels and at the end it defines the state->render, computes the error and defines the quality.
    - trace_until_quality: call trace_sample until the quality is reached.
    - trace_by_budget: call trace_samples with the step equal to the budget.
    - create_sample_spread: it allows to find the neighbours of the considered pixels.
    - all_image_ij: This function obtains all the pixels from the image.
    - parallel_pixels_in_list: This feature allows to speed up computation by using threads to process pixels.
    - render_samples: It progressively computes an image by calling trace_samples multiple times. At first it traces a minimum number of samples and then starts a loop until one of the checkEnd conditions is reached. In the loop it selects the pixels that are below the required quality and calls trace_samples until they reach it. Furthermore, it checks that the neighbors of each considered pixel also have the same sample number, otherwise call trace_by_budget on them.