EXTRA GRADE PROJECT

[Eye Gaze Contingent Rendering]
100 points (worth 3% of overall maximum class grade)
Due Date: December 2nd, 2009 11:55p.m.
CS 5929 Design and Analysis of Algorithms
Department of Computer Science, Texas State University

Motivation: Many virtual reality worlds require tremendous processing power from GPU with even additional hardware such as physics simulation hardware required to computer realistic effects. Does all the resolution and the quality of the effects is required uniformly on the screen? The answer is no. Effective resolution power of our eye is such that we can see with the high quality only a small fraction of the screen (2 degrees of visual angle) and the peripheral part of our vision is effectively blurred. The sensitivity function for our vision can be defined as:

$$S(x,y) = \frac{1}{1+0.24\theta_E(x,y)} \qquad (1)$$

Here, S is the eye visual sensitivity as a function of the image position (x,y), and Θ_E(x,y) is the eccentricity (the distance between the center of the screen and the current gaze location) in the visual angle .

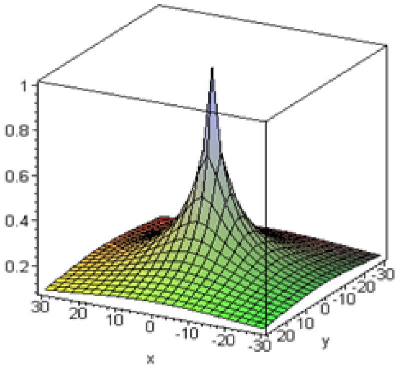

Figure 1. Eye visual sensitivity function when an eye is looking directly at the center of the screen.

Therefore if you have a 3D scene to propose and implement a resolution degradation algorithm that would keep a high mesh, high quality texture (or high quality effects) at the point where you are looking at and reduce quality according to the Visual Sensitivity Function in the periphery. To be able to successfully show the effectiveness of your algorithm you should think about a scenario where if everything is displayed with high quality, the framerate would be low and after you apply your peripheral degradation algorithm the framerate will go up. I understand that students don't have an experience working with an eye tracker therefore you can simulate an eye tracker by a mouse pointer, i.e., the high quality appears at the tip of the mouse cursor. You can see an example of software at http://www.svi.cps.utexas.edu/software.shtml. This software does not reduce average framerate, but it is a good visual example.

Project Main Steps:

- Select teammates (up to three students can be on a team)

- Select a specific virtual environment where you would apply your algorithm. It can be an open source game or something that you can implement in a rendereding engine such as OGRE 3D (http://www.ogre3d.org/).
- Select a specific scenario where parameters such as resolution, quality, number of objects, effects, etc. are fixed and produce a low **AVERAGE** framerate case.
- Develop a specific application of the visual sensitivity function applied to this scenario. Example: resolution, texture, quality of effects is degraded toward the periphery. You can just do one aspect not all of them.
- After you apply your algorithm the **AVERAGE** framerate should go up. You need to be able to explain why exactly it goes up.
- You should implement your code using Windows, *nix – based implementations are not allowed!!.
- Put comments in your code. The comments should allow a person who is not familiar with your code to understand it
- Write report file CS5329_GazeContingentRendering_[Name of the Teammates].doc. Only **one** report per group is required. The report file must contain following information – points will be deducted from you projects if you don't provide following information
    - Your name and name of your teammates
    - Identify what responsibilities each teammate had in this project, e.g., coding, documentation, debugging, etc.
    - Approximate amount of time spent in design, coding, debugging, report writing in hours for **EACH** team member
    - Describe what software you used to implement this project and how to install this software on the computer.
    - Describe how to run your code.
    - Describe the hardware of your computer.
    - Describe specifically how you achieved **LOW** framerates at the beginning of your project. Specify such things as resolution, effects, or other factors contributing to low resolution.
    - Describe the algorithm how you apply a visual sensitivity function to this scenario. Provide high level pseudocode for your algorithm.
    - Specify what are the reasons that the framerate goes down.
    - Brief description of challenges you have encounter during the project. Also a brief description of how those challenges were solved. If shortcuts were used identify those shortcuts and the reasons why you have used them.
    - Provide all the references (scientific papers, websites) that you have read to accomplish the implementation of your algorithm.
- Be ready for a 15 minute presentation to be presented on December 2nd where you would present your software and talk about items outlined in the report.

Extra reading material on the topic:

Murphy, H. and A.T. Duchowski, *Hybrid image-/model-based gaze-contingent rendering*, in *Proceedings of the 4th symposium on Applied perception in graphics and visualization*. 2007, ACM: Tubingen, Germany. (can be downloaded at www.acm.org)

How to Submit:

On top of each file include your name, your team members, data and project number.
Add:

```
/**************************************************
Name/team members names:
Date:
Project/Question Number:
Design and Analysis of Algorithms
Instructor: Komogortsev, TSU
***************************************************/
```

Comment your source files
Create an archive (zip or rar) of your source files, designer specification document and report file. Name your archive CS5329_GazeContingentRendering_[Name of the Teammates].doc You need to upload these files on TRACS in one package. **EACH** member of the team must upload this file
You will receive a confirmation e-mail from TRACS when your files are uploaded. If you don't receive this e-mail, double check that you have submitted everything properly!
Once you have submitted your code you cannot re-submit it – please check everything carefully before submission.

Late submissions no late submissions will be accepted!!!.
In case when one group copies the code from another group ZERO grade would be assigned to each group.