DH2323 DGI18

COMPUTER GRAPHICS AND INTERACTION

PROJECTS

Christopher Peters

CST, KTH Royal Institute of Technology, Sweden

chpeters@kth.se

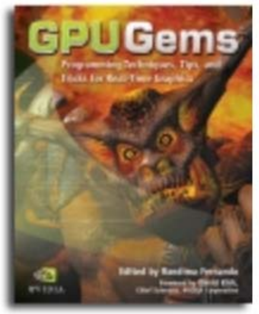
ROYAL INSTITUTE OF TECHNOLOGY

Project Ideas

1. www.gamasutra.com

GAMASUTRA
The Art & Business of Making Games

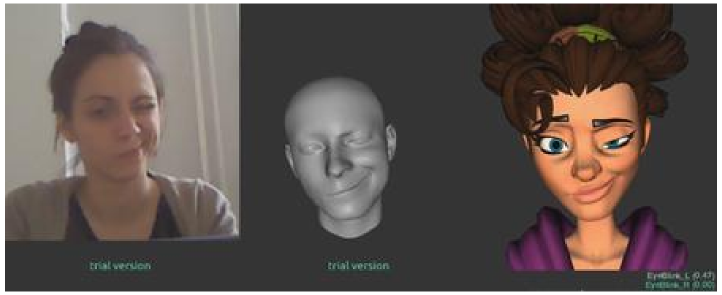
2. developer.nvidia.com/gpugems



GPU Gems

Project Ideas

DGI15 Project Blogs



My Expressive Avatar: https://myexpressiveavatar.wordpress.com

Restaurang - Pub - Café
STOCKHOLMS OSTRA
KA HÖGSKOLAN
Today, I want to give a brief introduction around KTH.
Now, I am located at the metro station near the university.

A short KTH student story: https://campussimulation.wordpress.com/

DGI16 Project Blogs

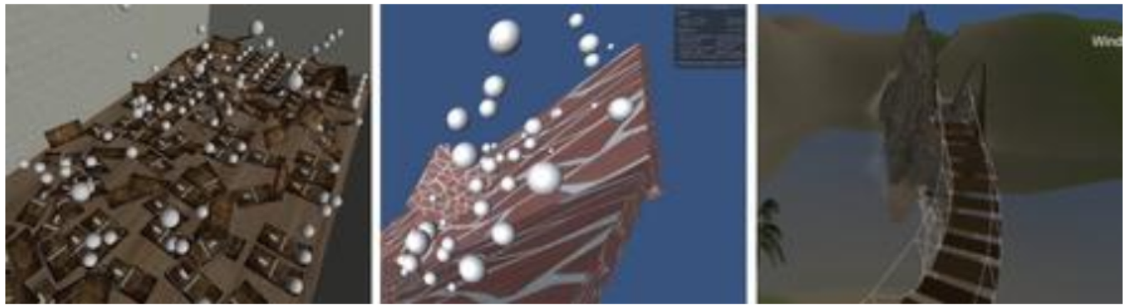
Norrsken: http://cyclooparticlesystem.blogspot.se/

Lightweight atmospheric shader: https://lightshaderdevlog.wordpress.com/

Project Ideas

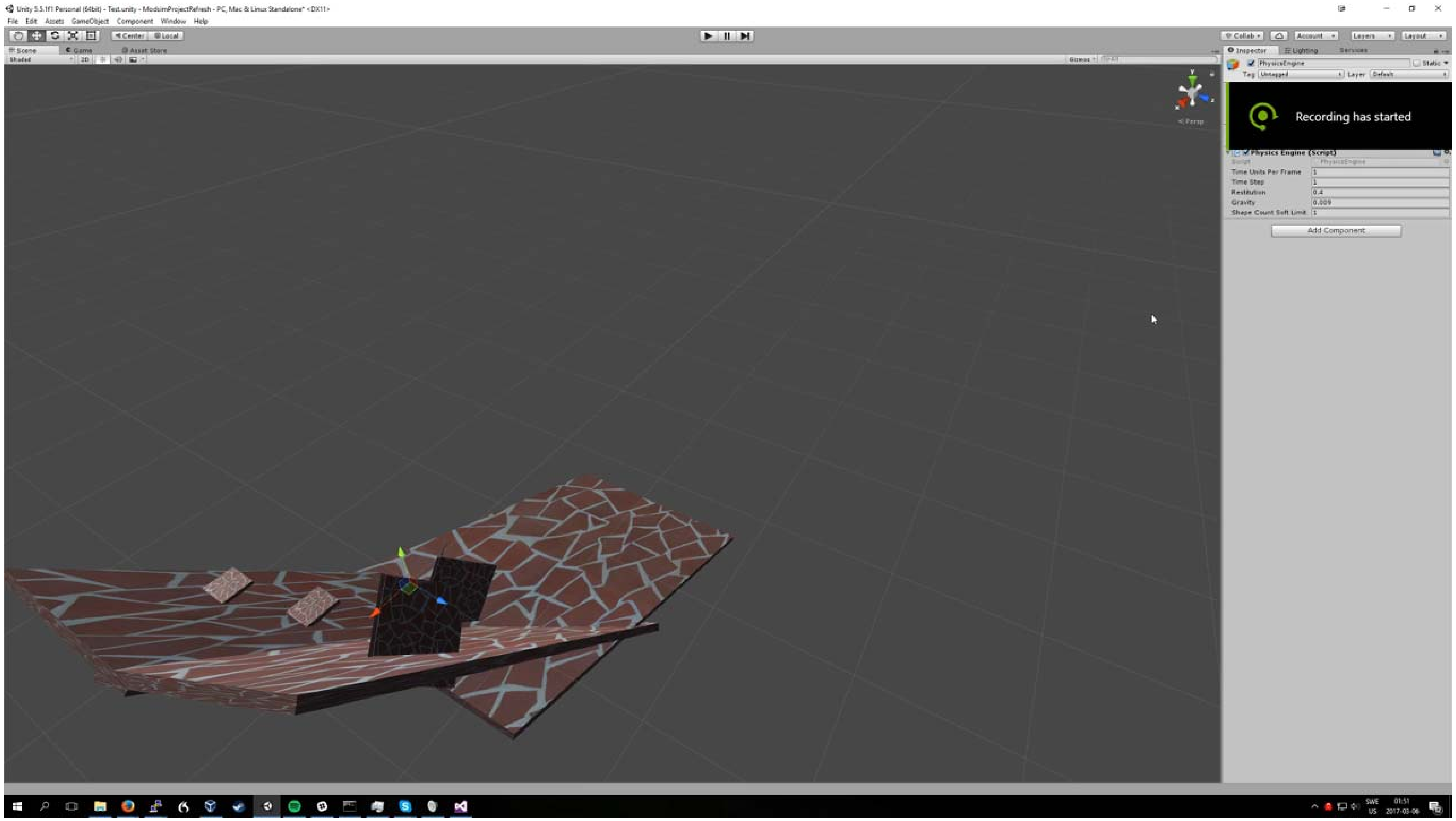
Models and Simulation

This is a course that deals with mathematical models, and numerical methods and algorithms for computer simulation. Modelling and simulation is increasingly important in science and technology, and is also used in entertainment such as physics engines for animation and computer games. Basic mathematical models as particle systems and mass-spring system are presented in the form of ordinary differential equations. The course focuses on practical aspects of methods and algorithms, and implementation of these in computer programs. The course includes a project where the methods are used to model a problem from reality, a scene, or to build a computer game.



The course webpage for a particular study term is found under "General" ("Allmänt") in the menu to the left. Here is a link to the DD1354 Canvas page.

Project Ideas


Unity 5.5.1f1 Personal (64bit) - Test.unity - ModsimProjectRefresh - PC, Mac & Linux Standalone* <DX11>
File Edit Assets GameObject Component Window Help
Center
Local
Scene
Game
Asset Store
Shaded
2D
Gizmos
Persp
Collab
Account
Layers
Layout
Inspector
Lighting
Services
Static
Add Component

Recording has started

PhysicsEngine

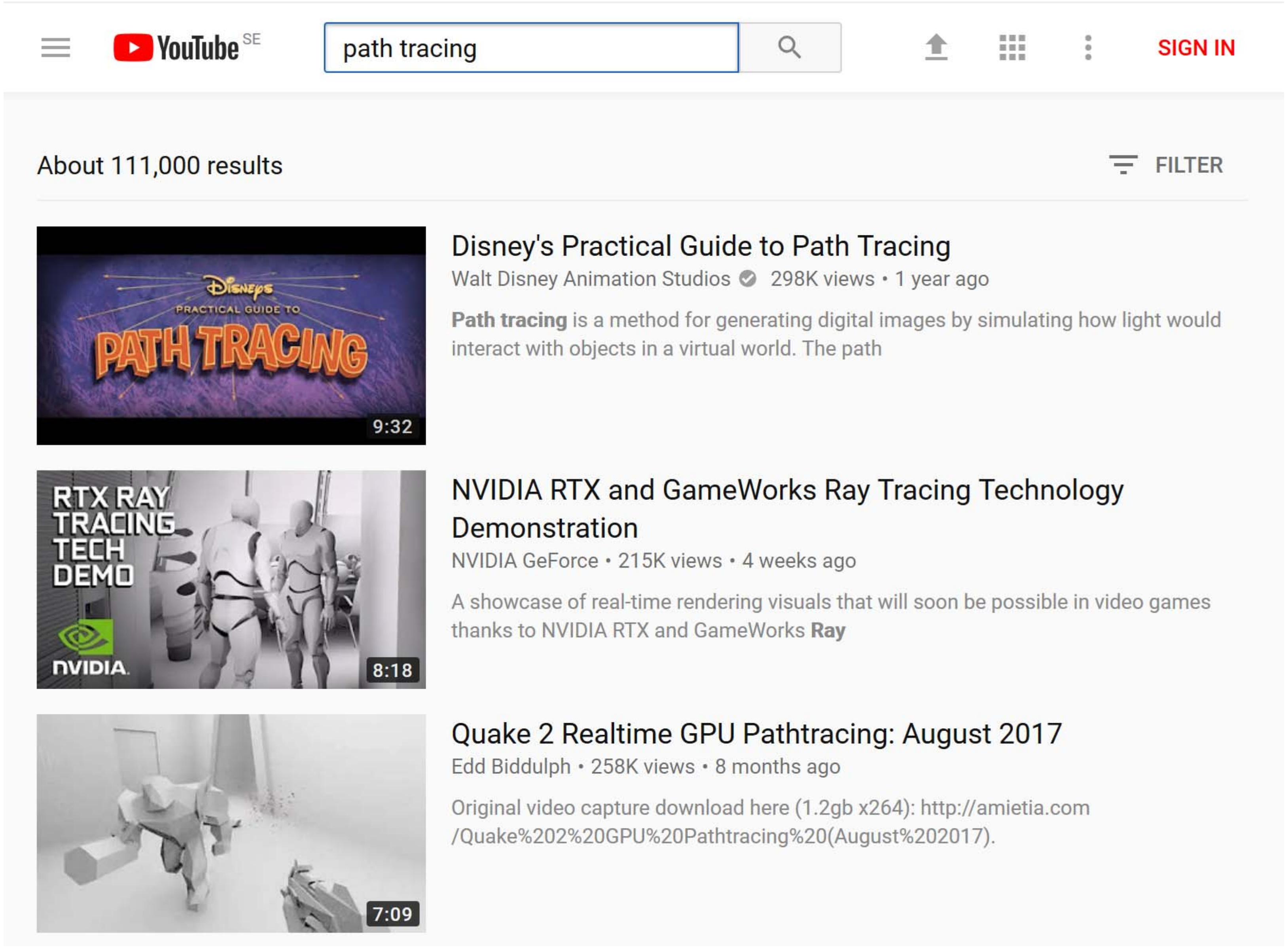
Tag: Untagged — Layer: Default

Physics Engine (Script)

Script	PhysicsEngine
Time Units Per Frame	1
Time Step	1
Restitution	0.4
Gravity	0.009
Shape Count Soft Limit	1

ROYAL INSTITUTE OF TECHNOLOGY

Project Ideas



path tracing

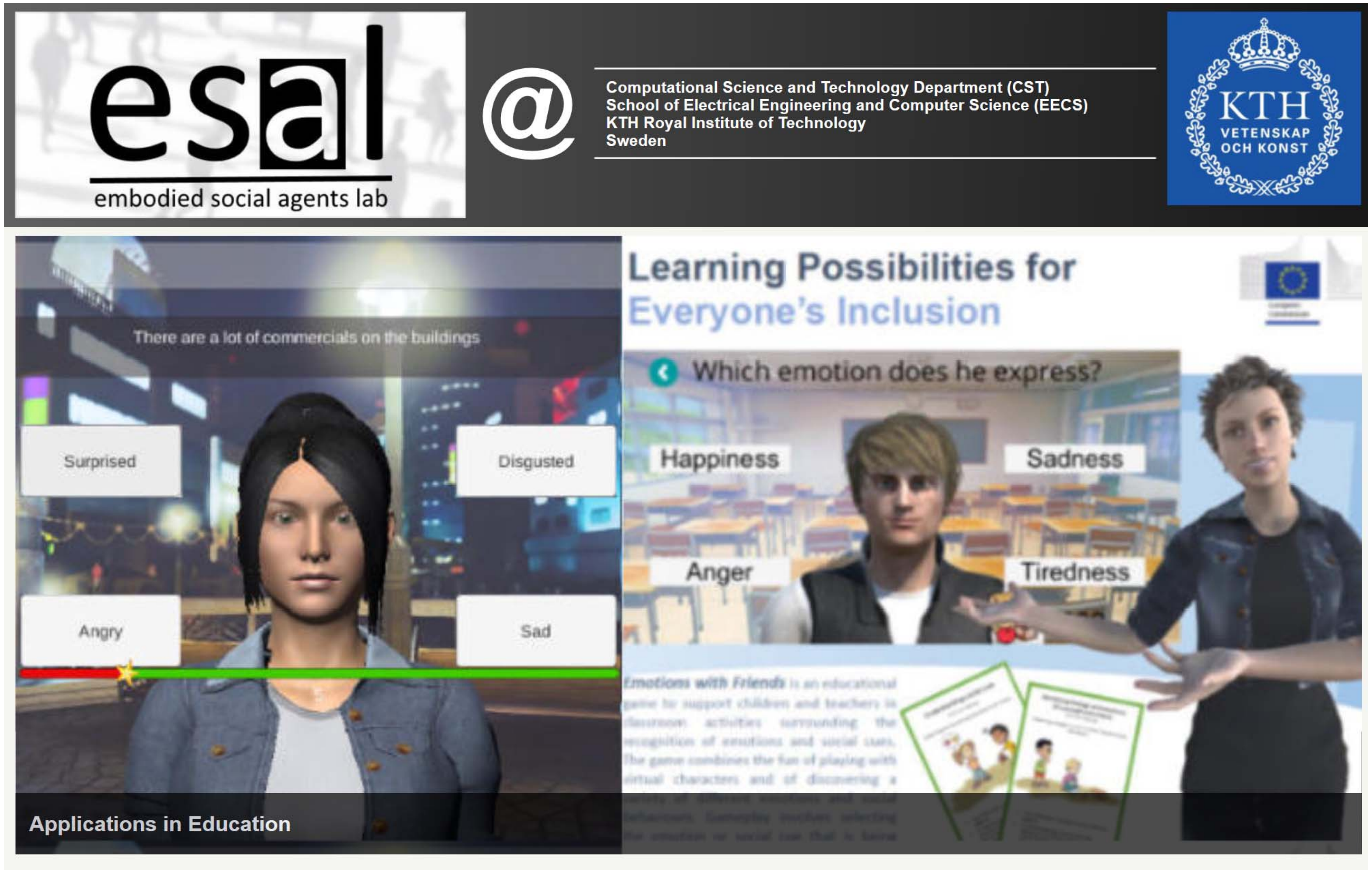
About 111,000 results

Disney's Practical Guide to Path Tracing
Walt Disney Animation Studios · 298K views • 1 year ago
Path tracing is a method for generating digital images by simulating how light would interact with objects in a virtual world. The path

NVIDIA RTX and GameWorks Ray Tracing Technology Demonstration
NVIDIA GeForce • 215K views • 4 weeks ago
A showcase of real-time rendering visuals that will soon be possible in video games thanks to NVIDIA RTX and GameWorks **Ray**

Quake 2 Realtime GPU Pathtracing: August 2017
Edd Biddulph • 258K views • 8 months ago
Original video capture download here (1.2gb x264): http://amietia.com /Quake%202%20GPU%20Pathtracing%20(August%202017).

Project Ideas


esal @

embodied social agents lab

Computational Science and Technology Department (CST)
School of Electrical Engineering and Computer Science (EECS)
KTH Royal Institute of Technology
Sweden

KTH
VETENSKAP OCH KONST
There are a lot of commercials on the buildings
Surprised
Disgusted
Angry
Sad
Applications in Education
Learning Possibilities for Everyone's Inclusion
Which emotion does he express?
Happiness
Sadness
Anger
Tiredness

Emotions with Friends is an educational game to support children and teachers in classroom activities surrounding the recognition of emotions and social cues. The game combines the fun of playing with virtual characters and of discovering a variety of different emotions and social behaviours. Gameplay involves selecting the emotion or social cue that is being

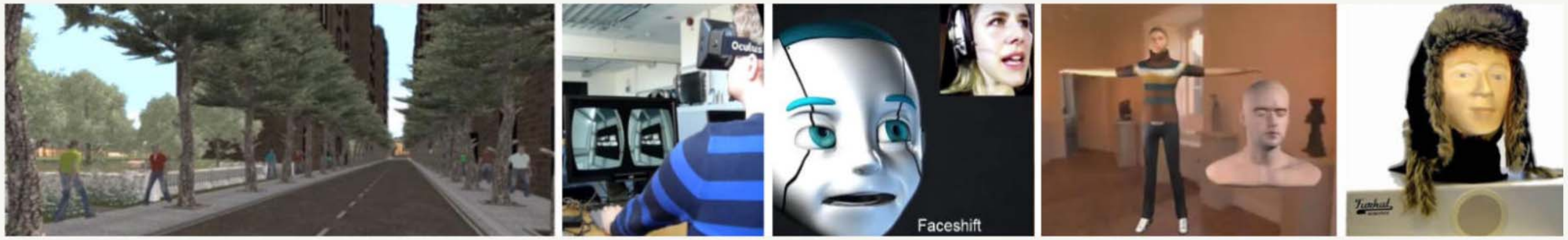
http://www.csc.kth.se/~chpeters/ESAL/

STUDENT PROJECTS

COMPUTER GAME TECHNOLOGIES * VIRTUAL REALITY * VIRTUAL CHARACTERS AND CROWDS * URBAN PERCEPTION

The images above are from student projects that I've supervised in courses I teach or am affiliated with, including DD1354, DH2323, DT2350, DD1392 MVK, SA104X, DD2463 (Advanced Individual Course), DD3336 and Exjobb (Master thesis).

I have a large variety of projects available to all levels and backgrounds of students at KTH. They relate to the areas of real-time computer graphics and animation, computer game technologies, perceptual computing and procedural generation. Applications range from architecture and urban modelling (using VR technologies such as the Oculus Rift, VIVE), to pedestrian behaviour and interaction with full-sized virtual humanoids using devices such as the Kinect and Leap Motion. Some project ideas are below. Note that many of these project areas include the opportunity to collaborate with research groups both within and outside KTH. You may obtain a further idea of the types of project areas that I typically supervise from the blogs here, here and here.



PROJECT IDEAS

Here are some examples of general project areas/ideas (with links to further references):

1) Scheduling algorithm for simulating the routines of **autonomous pedestrians** in a virtual KTH campus [1], [2].

2) Emotion recognition and mirroring game using **virtual characters' facial expressions and body motions** for learning scenarios between robots, agents and children [1].

3) **Inverse procedural generation** techniques for the automatic generation of stylised virtual urban environments [1], [2], [3].

4) **Machine learning** and **data-driven approaches** from mobile robotics applied to the simulation of small groups of virtual agents [1], [2].

Crowds

Crowds

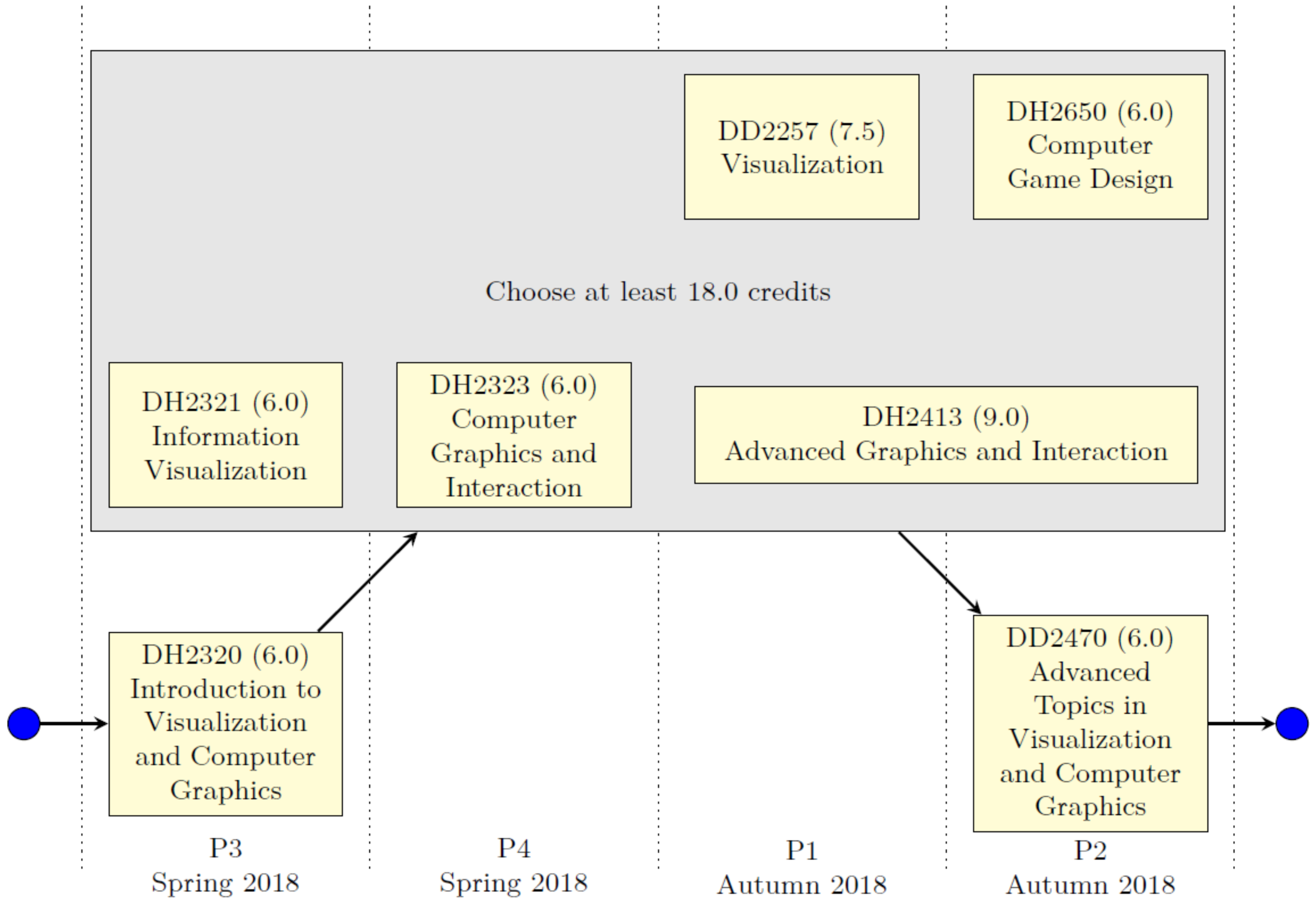
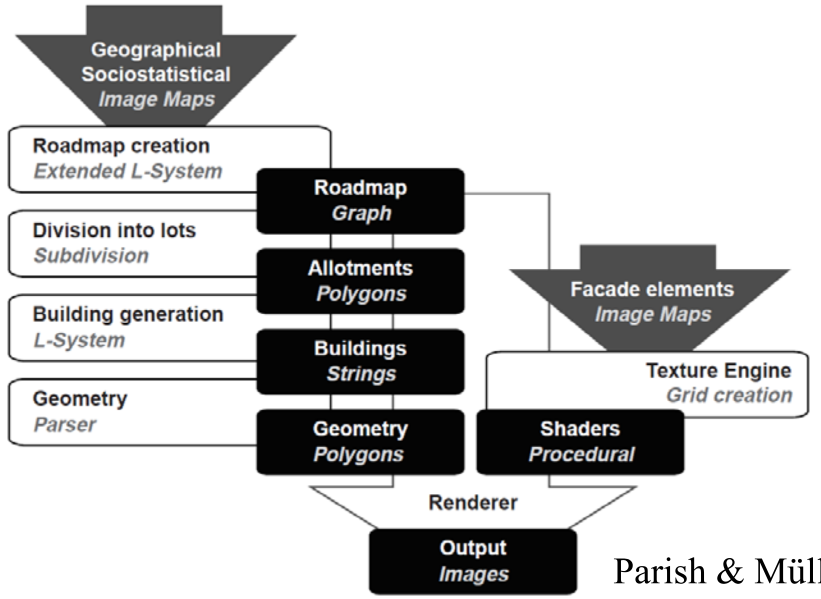

DD2257 (7.5) Visualization
DH2650 (6.0) Computer Game Design
Choose at least 18.0 credits
DH2321 (6.0) Information Visualization
DH2323 (6.0) Computer Graphics and Interaction
DH2413 (9.0) Advanced Graphics and Interaction
DH2320 (6.0) Introduction to Visualization and Computer Graphics
DD2470 (6.0) Advanced Topics in Visualization and Computer Graphics
P3 Spring 2018
P4 Spring 2018
P1 Autumn 2018
P2 Autumn 2018

Figure 7: CSVG — Visualization and Interactive Graphics

Procedural Generation

Vanegas et al. 2012



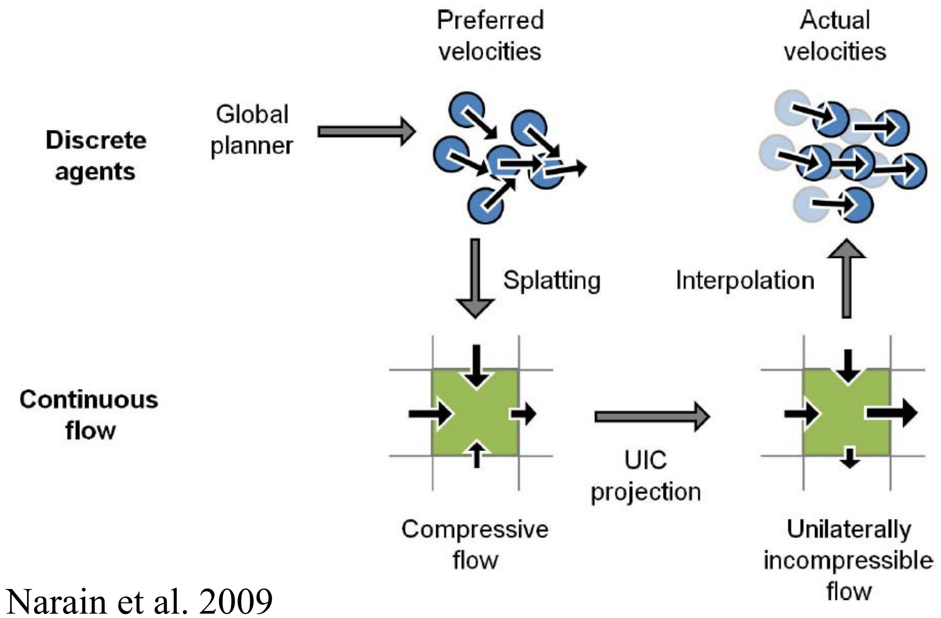
Parish & Müller 2001

- Generate building faces (Müller et al. 2007), street regions (Vanegas et al. 2012) and cities (Parish & Müller 2001) according to sets of rules
- Primarily concerned with the *automatic* generation of building, street and city *geometry*
- Challenging to enable easy user control and to create a mix of styles

ROYAL INSTITUTE OF TECHNOLOGY

DD2470 Advanced Project

Simulation and Rendering of Crowds

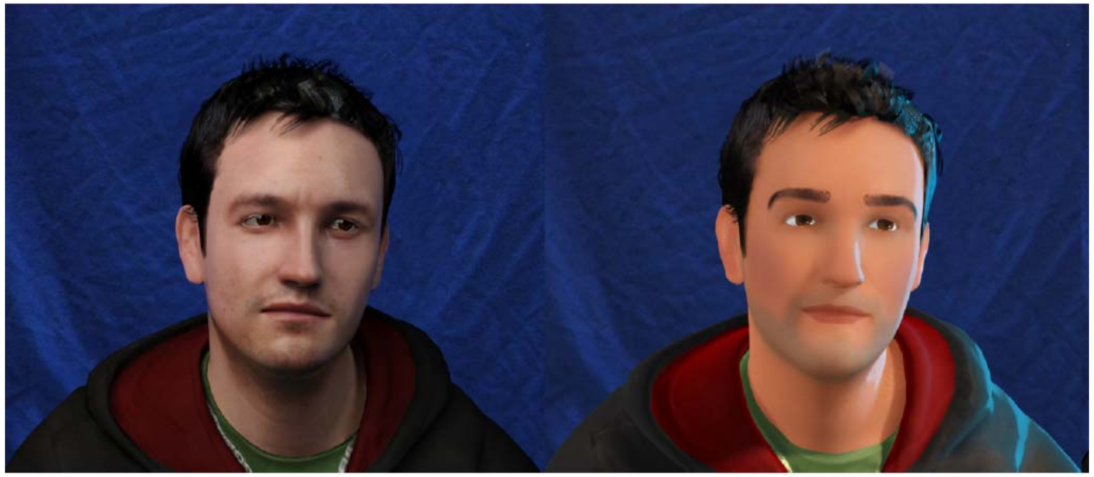


Narain et al. 2009

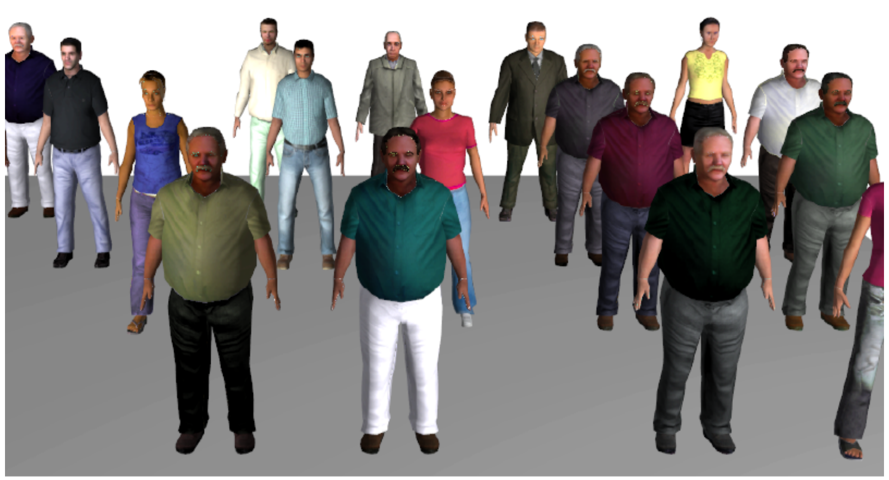
Dobbyn et al. 2005

- Enable large crowds of characters to walk together at high densities and to be rendered in real-time
- Simulation: Hybrid flow simulation and steering model (Narain et al. 2009)
- Imposter rendering: Complex 3D meshes are rendered from different camera angles as 2D sprites (Dobbyn et al. 2005)

Perception of Characters and Crowds



McDonnell et al. 2012



McDonnell et al. 2008

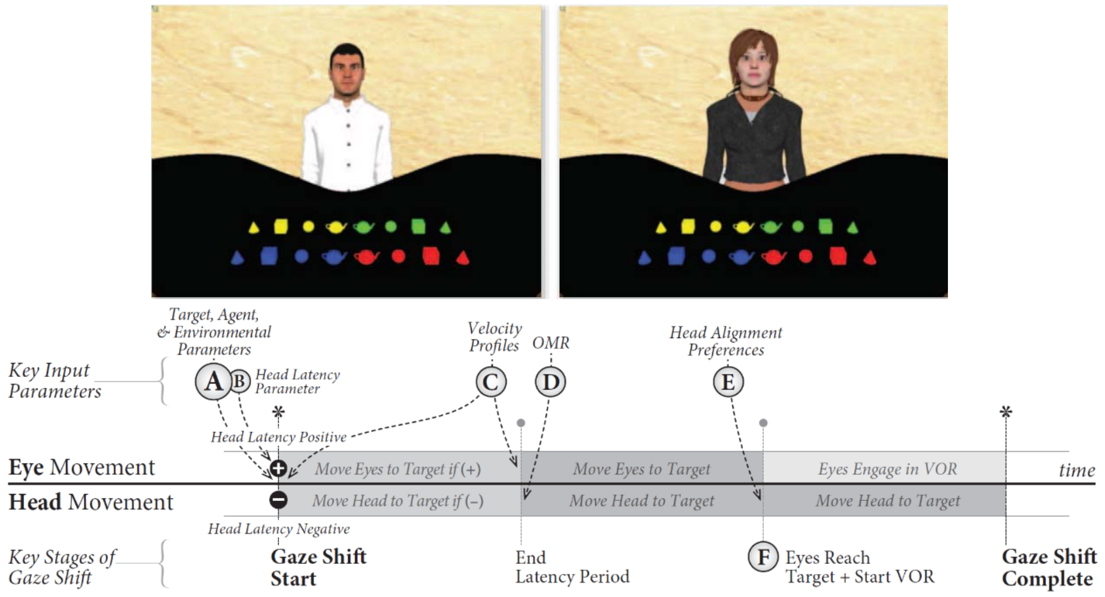
- How do people perceive different rendering styles for virtual characters (McDonnell et al. 2012) and at what point can we no longer identify clones in a crowd (McDonnell et al. 2008)?
- The above involve designing/doing user studies with participants
- Rendering styles: What impact do they have on our impressions of character emotions, behaviour and uncanniness?
- Clones: Only need to use the minimum amount of geometry necessary to represent a plausible crowd

Email: ***chpeters@kth.se*** **http://www.csc.kth.se/~chpeters/projects.html**

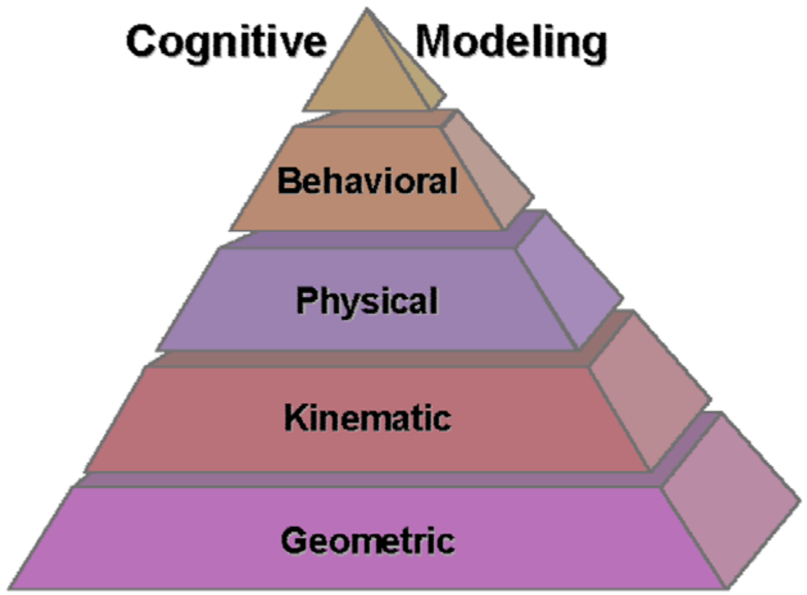
ROYAL INSTITUTE OF TECHNOLOGY

DD2470 Advanced Project

Controlling Animated Characters



Andrist et al. 2012



Funge et al. 1999

- From controlling the gaze movements of humanoid characters (Andrist et al. 2012) to creating behaviour and cognitive models of control that can drive them (Funge et al. 1999).
- *Physical*, *behavioural and cognitive* control: cognitive control as the apex of the computer graphics modelling hierarchy

Project Steps

1) Form groups (where desired)
2) Initial specification (+submit)
3) Feedback and iterate
4) Start blog
5) Implementation
6) Submission

Continue to work on your specification between steps 2 & 3

- Prototypes, feasibility

Forming a Group?

- Recommend max. group size of 2
  - Your choice
  - Post a message on KTH Social or go to lab session to find project partner
- Recommendation:
- Do the above as soon as possible!

Specification Contents

1. Grade you are aiming for
2. Some background to the area/problem
3. Implementation specifics (as many as possible)

   Technologies, problem, constraints/delimitations

Technologies (top-down)

ROYAL INSTITUTE OF TECHNOLOGY

VICSTHLM

VISUALISATION INTERACTION COLLABORATION

4K Screen, HTC Vive, Oculus Rift, Tabletops, Kinects, 3D Printer …

Specification Contents

1. Grade you are aiming for
2. Some background to the area/problem
3. Implementation specifics (as many as possible)

   Technologies, problem, constraints/delimitations
4. What final system will look like/do

   (include sketches if you like)
5. References (2-3)

Specification Contents

6. Potential risks/challenges

   And how they might be avoided/minimized

7. Idea for a perceptual study

   How could perception and interaction relate to your project? One paragraph.

8. Link to blog containing first blog entry

   Hint: your specification and a representative image can be your first blog entry

Specification Submission

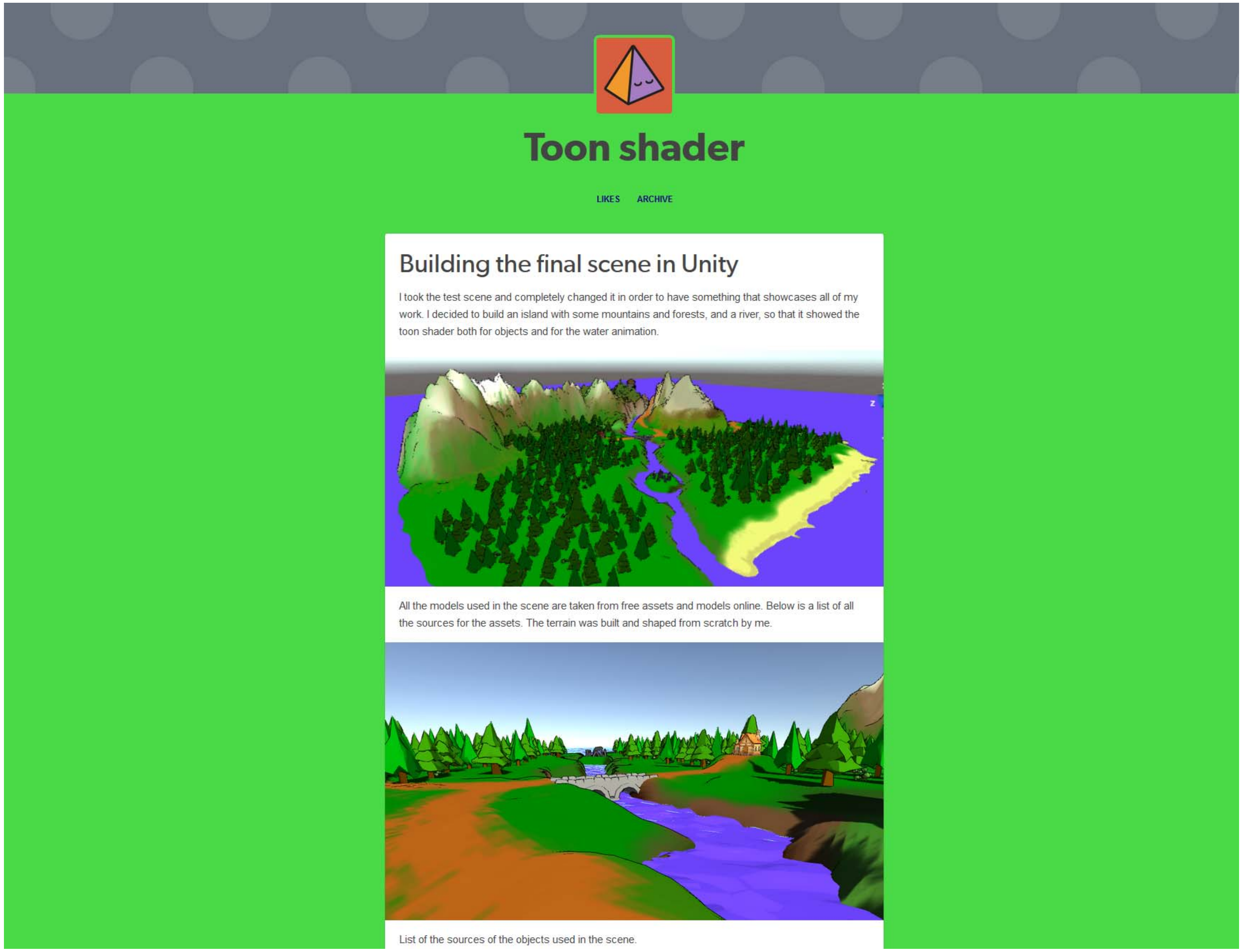
Submit through Canvas

- Only one group member needs to submit on behalf of the group
- Must communicate feedback/results to others
- Clearly label group members (names, email addresses) on specification
- Deadline: as soon as possible!

Example (detailed) specification...

ROYAL INSTITUTE OF TECHNOLOGY

The blog


Toon shader

LIKES ARCHIVE

Building the final scene in Unity

I took the test scene and completely changed it in order to have something that showcases all of my work. I decided to build an island with some mountains and forests, and a river, so that it showed the toon shader both for objects and for the water animation.

All the models used in the scene are taken from free assets and models online. Below is a list of all the sources for the assets. The terrain was built and shaped from scratch by me.

List of the sources of the objects used in the scene.

Implementation Phase

- Critical phase in the project
- Things will go wrong
  - Your specification will likely change
  - Your whole project idea may change
- Attend the lab sessions in order to:
  - Do group work together
  - Get feedback and guidance
- You will need to be proactive and ask

ROYAL INSTITUTE OF TECHNOLOGY

Project Submission

Final Deadline:

Friday 25th May 23:59 Stockholm

Submitted to Canvas as an archive (.zip)

Contents:

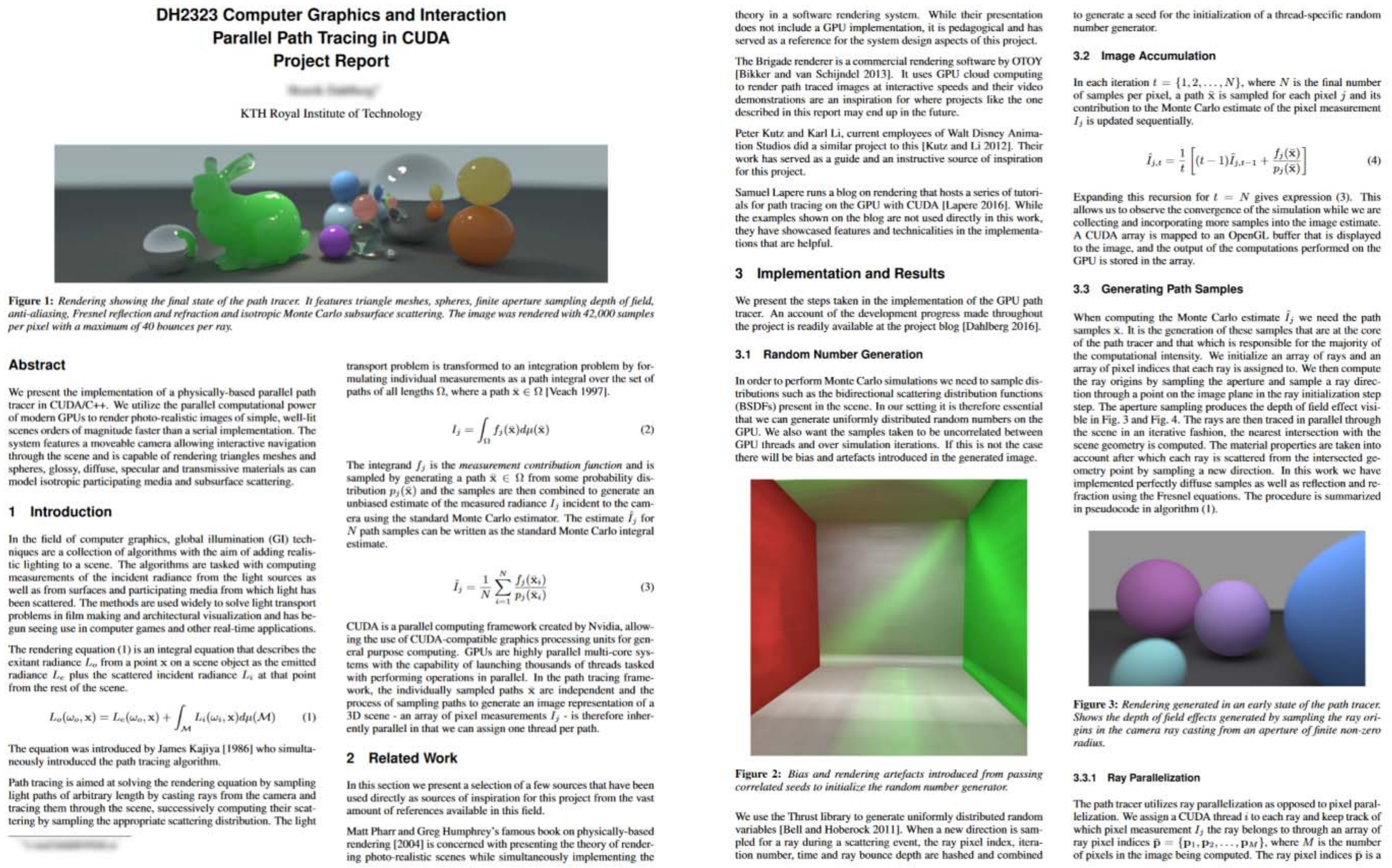
1. Report
2. All implementation source and asset files
3. Project specification

Project Report Contents

1) Group member names, IDs and emails addresses
2) Link to your blog
3) Grade you are aiming for in the project
4) Implementation background and description
5) Screenshots
6) For grade ‘A’: description of related perceptual study
7) Description of what each group member did
8) References

Formatting: ideally Latex (esp. grade ‘A’)

Project Report Example



DH2323 Computer Graphics and Interaction
Parallel Path Tracing in CUDA
Project Report

[illegible]

KTH Royal Institute of Technology

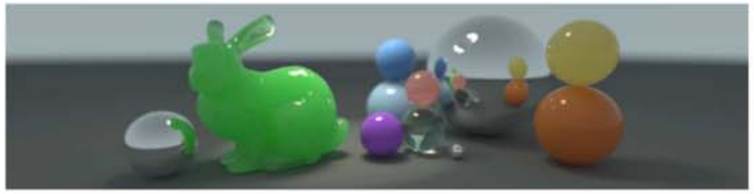


Figure 1: *Rendering showing the final state of the path tracer. It features triangle meshes, spheres, finite aperture sampling depth of field, anti-aliasing, Fresnel reflection and refraction and isotropic Monte Carlo subsurface scattering. The image was rendered with 42,000 samples per pixel with a maximum of 40 bounces per ray.*

Abstract

We present the implementation of a physically-based parallel path tracer in CUDA/C++. We utilize the parallel computational power of modern GPUs to render photo-realistic images of simple, well-lit scenes orders of magnitude faster than a serial implementation. The system features a moveable camera allowing interactive navigation through the scene and is capable of rendering triangles meshes and spheres, glossy, diffuse, specular and transmissive materials as can model isotropic participating media and subsurface scattering.

1 Introduction

In the field of computer graphics, global illumination (GI) techniques are a collection of algorithms with the aim of adding realistic lighting to a scene. The algorithms are tasked with computing measurements of the incident radiance from the light sources as well as from surfaces and participating media from which light has been scattered. The methods are used widely to solve light transport problems in film making and architectural visualization and has begun seeing use in computer games and other real-time applications.

The rendering equation (1) is an integral equation that describes the exitant radiance L_o from a point $\mathbf{x}$ on a scene object as the emitted radiance L_e plus the scattered incident radiance L_i at that point from the rest of the scene.

$$L_o(\omega_o, \mathbf{x}) = L_e(\omega_o, \mathbf{x}) + \int_{\mathcal{M}} L_i(\omega_i, \mathbf{x}) d\mu(\mathcal{M}) \qquad (1)$$

The equation was introduced by James Kajiya [1986] who simultaneously introduced the path tracing algorithm.

Path tracing is aimed at solving the rendering equation by sampling light paths of arbitrary length by casting rays from the camera and tracing them through the scene, successively computing their scattering by sampling the appropriate scattering distribution. The light transport problem is transformed to an integration problem by formulating individual measurements as a path integral over the set of paths of all lengths Ω, where a path $\bar{\mathbf{x}} \in \Omega$ [Veach 1997].

$$I_j = \int_{\Omega} f_j(\bar{\mathbf{x}}) d\mu(\bar{\mathbf{x}}) \qquad (2)$$

The integrand f_j is the *measurement contribution function* and is sampled by generating a path $\bar{\mathbf{x}} \in \Omega$ from some probability distribution $p_j(\bar{\mathbf{x}})$ and the samples are then combined to generate an unbiased estimate of the measured radiance I_j incident to the camera using the standard Monte Carlo estimator. The estimate $\hat{I}_j$ for N path samples can be written as the standard Monte Carlo integral estimate.

$$\hat{I}_j = \frac{1}{N} \sum_{i=1}^{N} \frac{f_j(\bar{\mathbf{x}}_i)}{p_j(\bar{\mathbf{x}}_i)} \qquad (3)$$

CUDA is a parallel computing framework created by Nvidia, allowing the use of CUDA-compatible graphics processing units for general purpose computing. GPUs are highly parallel multi-core systems with the capability of launching thousands of threads tasked with performing operations in parallel. In the path tracing framework, the individually sampled paths $\bar{\mathbf{x}}$ are independent and the process of sampling paths to generate an image representation of a 3D scene - an array of pixel measurements I_j - is therefore inherently parallel in that we can assign one thread per path.

[illegible]

2 Related Work

In this section we present a selection of a few sources that have been used directly as sources of inspiration for this project from the vast amount of references available in this field.

Matt Pharr and Greg Humphrey's famous book on physically-based rendering [2004] is concerned with presenting the theory of rendering photo-realistic scenes while simultaneously implementing the theory in a software rendering system. While their presentation does not include a GPU implementation, it is pedagogical and has served as a reference for the system design aspects of this project.

The Brigade renderer is a commercial rendering software by OTOY [Bikker and van Schijndel 2013]. It uses GPU cloud computing to render path traced images at interactive speeds and their video demonstrations are an inspiration for where projects like the one described in this report may end up in the future.

Peter Kutz and Karl Li, current employees of Walt Disney Animation Studios did a similar project to this [Kutz and Li 2012]. Their work has served as a guide and an instructive source of inspiration for this project.

Samuel Lapere runs a blog on rendering that hosts a series of tutorials for path tracing on the GPU with CUDA [Lapere 2016]. While the examples shown on the blog are not used directly in this work, they have showcased features and technicalities in the implementations that are helpful.

3 Implementation and Results

We present the steps taken in the implementation of the GPU path tracer. An account of the development progress made throughout the project is readily available at the project blog [Dahlberg 2016].

3.1 Random Number Generation

In order to perform Monte Carlo simulations we need to sample distributions such as the bidirectional scattering distribution functions (BSDFs) present in the scene. In our setting it is therefore essential that we can generate uniformly distributed random numbers on the GPU. We also want the samples taken to be uncorrelated between GPU threads and over simulation iterations. If this is not the case there will be bias and artefacts introduced in the generated image.

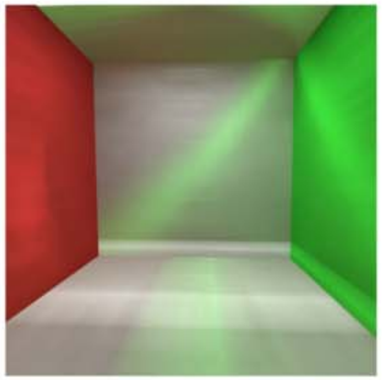


Figure 2: *Bias and rendering artefacts introduced from passing correlated seeds to initialize the random number generator.*

We use the Thrust library to generate uniformly distributed random variables [Bell and Hoberock 2011]. When a new direction is sampled for a ray during a scattering event, the ray pixel index, iteration number, time and ray bounce depth are hashed and combined to generate a seed for the initialization of a thread-specific random number generator.

3.2 Image Accumulation

In each iteration $t = \{1, 2, \ldots, N\}$, where N is the final number of samples per pixel, a path $\bar{\mathbf{x}}$ is sampled for each pixel j and its contribution to the Monte Carlo estimate of the pixel measurement I_j is updated sequentially.

$$\hat{I}_{j,t} = \frac{1}{t}\left[(t-1)\hat{I}_{j,t-1} + \frac{f_j(\bar{\mathbf{x}})}{p_j(\bar{\mathbf{x}})}\right] \qquad (4)$$

Expanding this recursion for $t = N$ gives expression (3). This allows us to observe the convergence of the simulation while we are collecting and incorporating more samples into the image estimate. A CUDA array is mapped to an OpenGL buffer that is displayed to the image, and the output of the computations performed on the GPU is stored in the array.

3.3 Generating Path Samples

When computing the Monte Carlo estimate $\hat{I}_j$ we need the path samples $\bar{\mathbf{x}}$. It is the generation of these samples that are at the core of the path tracer and that which is responsible for the majority of the computational intensity. We initialize an array of rays and an array of pixel indices that each ray is assigned to. We then compute the ray origins by sampling the aperture and sample a ray direction through a point on the image plane in the ray initialization step step. The aperture sampling produces the depth of field effect visible in Fig. 3 and Fig. 4. The rays are then traced in parallel through the scene in an iterative fashion, the nearest intersection with the scene geometry is computed. The material properties are taken into account after which each ray is scattered from the intersected geometry point by sampling a new direction. In this work we have implemented perfectly diffuse samples as well as reflection and refraction using the Fresnel equations. The procedure is summarized in pseudocode in algorithm (1).

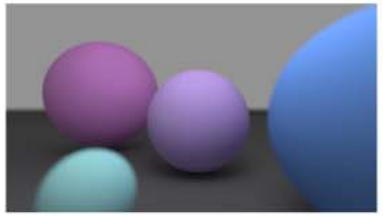


Figure 3: *Rendering generated in an early state of the path tracer. Shows the depth of field effects generated by sampling the ray origins in the camera ray casting from an aperture of finite non-zero radius.*

3.3.1 Ray Parallelization

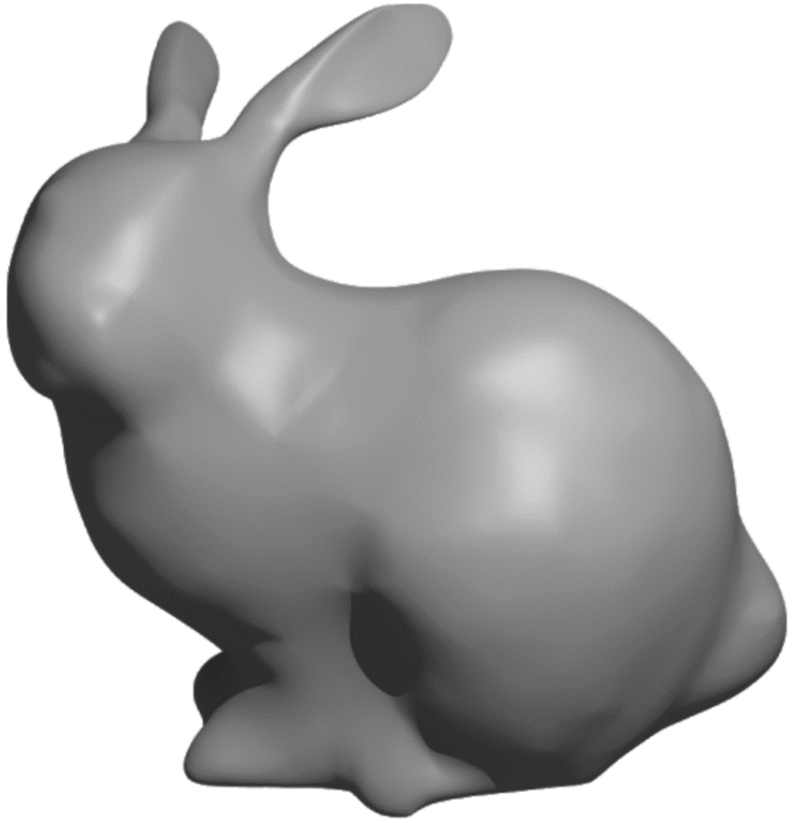
The path tracer utilizes ray parallelization as opposed to pixel parallelization. We assign a CUDA thread i to each ray and keep track of which pixel measurement I_j the ray belongs to through an array of ray pixel indices $\bar{\mathbf{p}} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M\}$, where M is the number of pixels in the image being computed. The ray pixel indices $\bar{\mathbf{p}}$ is a

Example User Study

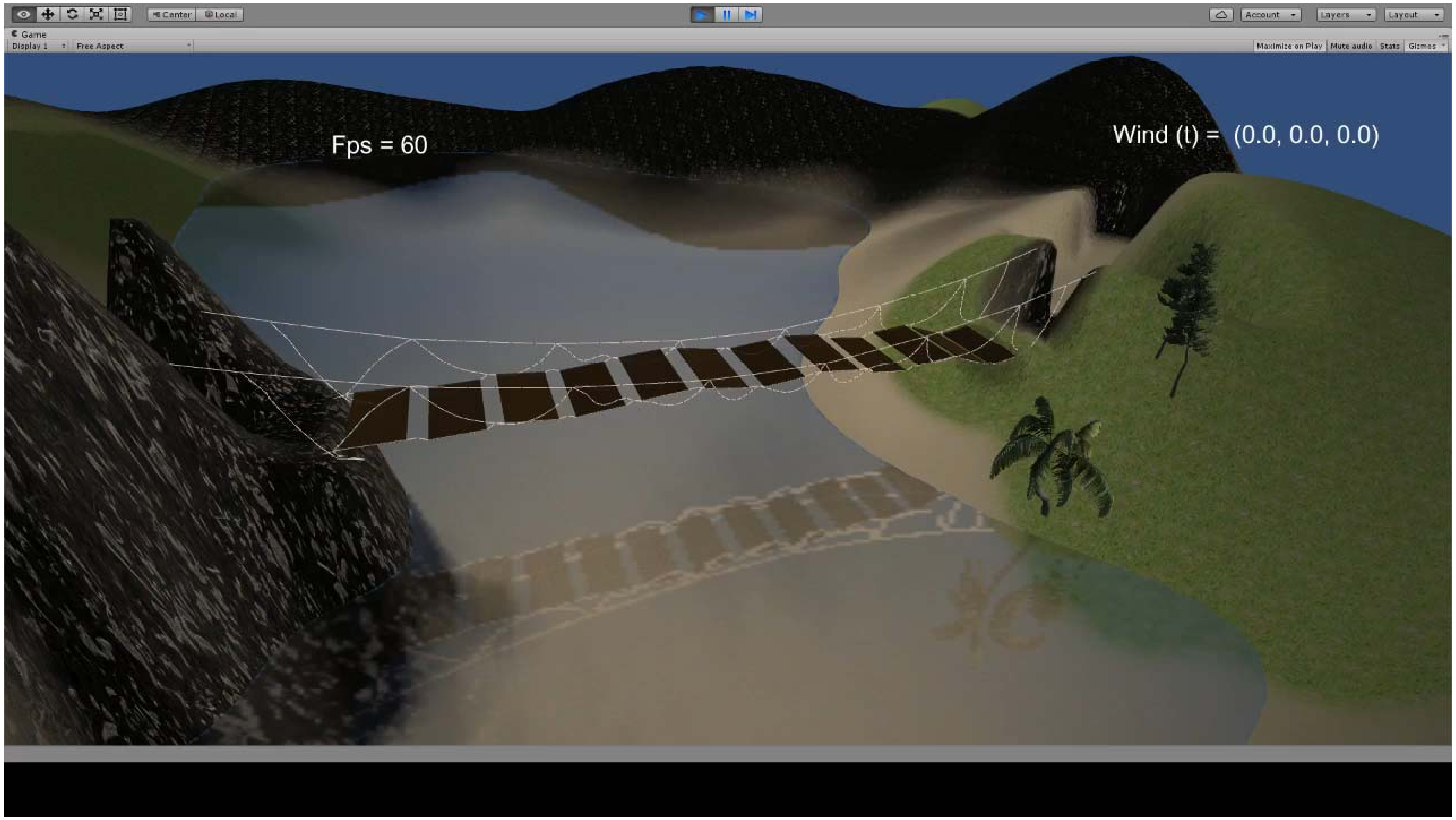
Exjobb on *subdivision surfaces*

Please try (and complete!) it:

https://goo.gl/forms/lkxBwx2sWa5ofjKa2



Example Final Demo



ROYAL INSTITUTE OF TECHNOLOGY

Assessment

- Exam (replaced by project)
- Project

  Individual or group project (1-3 members) on a topic related to computer graphics and interaction

Grading

- To pass:
  - Labs are P/F
  - Must do all the labs and a small project
    - Labs: Lab 1 + one of the *lab tracks*
    - Example small project: extend the labs (the lab tasks contain suggestions)
    - Grade D
- To excel:
  - More substantial projects lead to higher grades

Important!

- Please let us know if you need your *official* grades by a certain date!

- Include a message in the Canvas submission when you make it
- Also include a message at the start of your report

Coming Up Next

Guest Presentations:

Karl Arvidsson

KTH Master student

EA DICE

Topic: Real-time deformation simulation

Petter Lundahl

KTH Master student

ESAL

Topic: Anomalies in crowd simulation