Prof. Dr.-Ing. Ralf Steinmetz
Multimedia communications Lab

Dipl. Inf. Robert Konrad
Polona Caserman, M.Sc.

TECHNISCHE
UNIVERSITÄT
DARMSTADT

„Game Technology“
Winter Semester 2017/2018

Relevant topics and example questions

This document contains the most relevant topics for each lecture and a set of example problems for the exam. Note that if no problem for a topic is included, there can still be one in the final exam.

General notes:

- The C++ and hardware-parts of the lectures are not relevant for the lecture.
- The info in the “history” slides, unless noted otherwise here, is not relevant.

Lecture 1 - Input and Output

Relevant Topics

Basic attributes of waves (light and sound)
Color reception (Cones and rods)
Depth perception
Color representation in computers
Gamma Curves and Gamma Correction
Double Buffering and Vertical Synchronization

Example Problems

1) What problems can occur related to superposition of waves in game engines?
2) Which means does our eye use to see?
3) What are the two kinds of cells in our eyes? What is the difference between them?
4) Why does it make sense to encode colors with more bits for the green channel than for others?
5) Name one binocular cue the brain uses to perceive depth
6) Does our brain use only binocular cues for depth perception? If not, name one monocular cue it uses.
7) What causes tearing in graphics?
8) What can be done against tearing?
9) Why is gamma correction required?
10) You implement a tint functionality for your GUI. An input color is multiplied with the texture values to color them. You cannot reproduce the exact colors your designer provided you with as a reference. What could be the cause?
11) Gamma-correct the following color value: ….
12) How is gamma-correction carried out?

Lecture 2 - The Game Loop

Relevant Topics

Motion blur
Multithreading Types

Virtual frame time

Types of animation

Game loop

Example Problems

1) Name the basic steps in a game loop.
2) List and explain the most common problems in collision handling
3) How does motion blur occur in real life?
4) How can we approximate motion blur in games
5) What is the difference between cooperative and preemptive multithreading
6) An endless loop is created. Where is it more destructive - cooperative or preemptive
7) Virtual frame time - why
8) Procedural vs. iterative animations - describe the difference
9) f(x) = x * y - which type of animation
10) f(x +1) = x * y - which type of animation
11) What are parts of the game loop?

Lecture 3 - Software Rendering 1

Relevant Topics

Raycasting
Raytracing
Rasterisation
Culling
Texturing
Geometric interpretation and application of scalar & cross product

Example Problems

1) List the basic steps of a rasterisation algorithm.
2) List the basic steps of a raytracing algorithm.
3) Explain the different kinds of culling algorithms used in 3D graphics.
4) Why are shadows, reflections etc. relatively easy to compute using rasterization?
5) In which paradigm are triangles transformed and rendered?
6) How is a triangle rasterized
7) Name and describe one culling method
8) Why can it be advantageous to render from front to back when using z-buffering
9) What is frustum culling

Lecture 4 - Software Rendering 2

Relevant Topics

Depth Buffering
Perspective correct texturing (no need for the formula)
Gimbal lock
Matrix calculations (only on a higher level, you won’t have to do matrix multiplications by hand)

- Translation
- Rotation
- Shearing
- Scaling

Quaternion calculations (also only on a higher level)
Typical setup for camera and model matrices
Basic local lighting
Per pixel vs. per vertex

Example Problems

1) Explain the benefits and shortcomings of the depth buffering algorithm.
2) Given a rotation r around the y axis (in the form of an angle), a translation t (in the form of a three component vector) and a mesh p – show how to calculate the rotation followed by the translation using only matrices and using matrices and a quaternion. You don’t have to show how the matrices and quaternions themselves are built
3) What are the shortcomings of per-vertex lighting?
4) Why does z-buffering require back-to-front rendering for transparent pixels?
5) Why is perspective correct texturing necessary for good quality?
6) What is gimbal lock? What is one common strategy against it?
7) What is the given matrix doing?
8) Is this a normal or a regular vertex? Why?
9) What are the normal matrices used in rendering?
10) Why is quaternion interpolation via SLERP better than a linear interpolation of rotation matrices?
11) Explain the intensity = L N term

Lecture 5 - Basic Hardware Rendering

Relevant Topics

Basic structure of a GPU, differences to CPU
Antialiasing - Causes and solution approaches
Texturing
Texture filtering
Mip mapping
Depth buffer
Blending - Additive vs. standard
Phong Lighting Model
Cube maps
Shadow Mapping

Example Problems

1) List and explain the non-programmable components of a GPU.
2) Explain the benefits and shortcomings of postprocess antialiasing.
3) What are the basic components of the Phong lighting model?
4) Explain the primary problem of shadow maps.
5) Is the rasterizer programmable in the same way as shaders
6) What causes aliasing in graphics?
7) What is the way to combat antialiasing (i.e. be over the Nyquist frequency)
8) What is edge, supersample, multisample, post processing AA?
9) Why do texture filtering?
10) How is bilinear filtering done?
11) What to do at texture borders?
12) Why mip mapping?
13) How is it doneß
14) What is trilinear filtering?
15) What are the terms of Phong lighting?
16) What are the vectors in the following image?
17) What is a cubemap? Why is it used?
18) What is the worst case for shadow mapping?
19) Why cascaded shadow maps?
20) How does shadow mapping work?
21) How many times do we need to render the scene when using shadow mapping?
22) Why is the effect in the image showing up (low resolution of shadow map)

Lecture 6 - Bumps and Animations

Relevant Topics

Bump Mapping
Normal Mapping
Displacement Mapping
Deferred Shading
Particle Systems
Vertex Animations
Skeletal Animations, “Candy Wrapper”/”Achselhölle”

Example Problems

1) What are the shortcomings of normal mapping?
2) What are the benefits and problems of vertex animations?
3) What is the primary problem of vertex skinning as used in skeletal animation systems?
4) Why is a normal map often mostly blueish (128, 128, 255)
5) How is a normal encoded in a normal map?
6) What is the difference between normal and displacement mapping?
7) What is the advantage?
8) Is this displacement or normal mapping?
9) When is lighting calculated in deferred shading?
10) Why does deferred shading require more memory?
11) Can vertex animation key frames be interpolated?
12) How does the “candy wrapper” phenomenon appear? Why?
13) Is it possible to mix skeletal animation and vertex animation?

Lecture 7 - Physically-Based Rendering

Relevant Topics

Light sources
Point
Directional

Specular vs. diffuse reflections
Input and output of a BRDF
BRDF properties
Materials (Dielectrics vs. conductors)
Image Based Lighting
Fresnel reflection
The Microfacet Model
Understand basic structure of the formula
Basics steps of a GPU

Example Problems

1) List the different kind of lights used in computer graphics and compare them to real lights.
2) Explain diffuse light reflection.
3) The Fresnel effect gets stronger based on which value?
4) Which data structures are used for image based lighting in realtime graphics?
5) What properties do the facets have to fulfill that add light to the rendered image in a microfacet model?

New Tasks

a) Name two different kinds of light sources used in game engines.

b) Is the position of a directional light important?

c) Is the rotation of a directional light important?

d) Can a point/directional light be optimized away (i.e. if the area of influence is not visible?)

e) Which of the shown is diffuse and which specular?

f) Does diffuse light depend on the view direction?

g) What are the inputs and outputs of a BRDF?

h) In which part of the GPU is the BRDF evaluated? Per pixel?

i) How is image-based lighting implemented?

j) Name one shortcoming of a BRDF

k) Name and explain one properties of a physically-based BRDF

l) Is a BRDF that does … physically correct?

m) What is the main difference between a dielectric and a conductor concerning pyhsically based renderingß

n) Explain Fresnel reflection in your own words

o) Explain parts of the microfacet model formula

p) Active microfacets: What does this mean?

q) Normal distribution function: What does it do?

r) Shadow masking function: What does it do?

s) Vertex Fetch, Vertex Shader, Rasterization, Pixel Shader,

Lecture 8 - Physics 1

Relevant Topics

Newton's laws, especially no. 2 (F = ma)
Understand force, mass, acceleration
D'Alambert's principle (→ accumulator for forces)
Particles
Particle (systems), difference between particle and rigid body
Particle system parameters
Particle rendering methods

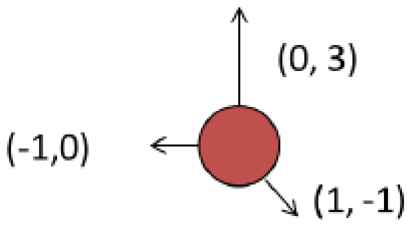
Euler integration for physics simulation
→ Get from F = ma to the differential equations
Rigid Bodies
Center of mass, application of forces on-center
Collision Detection
Sphere-sphere
Sphere-plane
Collision Response
Collision normal
Coefficient of restitution
Separating velocity
Calculate impulses
Solve interpenetration
Apply impulses

Example Problems

1) Newton's Laws

a) Name the terms of the formula F = ma and explain their meaning.

b) Calculate the overall force acting on the depicted object using D'Alambert's law.



2) Particle Systems

a) Name the main property that distinguishes particles from rigid bodies

b)

Given the view matrix of a camera, explain how a particle can be rendered as a quad that always faces the camera.

3) Rigid Bodies

Name the main property of rigid bodies that distinguishes them from particles.

What information is needed for a rigid body?

4) Collision Detection and Response

a) How can we determine whether two circles are intersecting? State which properties of the situation are needed and how they are used.

b) Assume that two circles are intersecting. Name the relevant properties we need for collision response and how they would be used.

What is the effect of the COR? What physical property does it approximate?

The separating velocity of two objects is positive. Does this always mean that the objects are not intersecting?

Why are impulse-based physics engines not physically accurate? In which main regard do they differ from reality?

Why is it bad to move objects out by just moving their positions?

Lecture 9 - Physics 2

Relevant Topics

Separating Axis Test

Understand how and why the SAT is applicable Understand relevant features

SAT for sphere-sphere SAT for triangle-sphere

Space partition algorithms Understand the purpose Regular grid Quadtree/Octree

KD-tree

Binary Space Partition

Broad Phase vs. Narrow Phase Understand the difference **Rotation, Torque**

Explain the difference between angular velocity + acceleration compared to linear Mass moment of inertia, Inertia Tensor

Torque Compute torque **Friction** Coulomb's law

Understand the relationship between friction force, coefficient of friction and normal force Explain the purpose of a contact basis

Example Problems

1) **Separating Axis Test**

b) Which of the following statements is true about the Separating Axis Test? Explain each answer.

1) When more than 50% of the individual feature tests have been negative (= no separating axis found), we can stop the testing.

2) When more than 50% of the individual feature tests have been positive (= a separating axis was found), we can stop the testing.

3) When the first negative feature test is found, we can stop calculating.

4) When the first positive feature test is found, we can stop calculating.

5) We always need to test all features.

6) For a circle, the SAT can not be applied as it has no distinct features.

2) Space partitioning algorithms

a) What is the main difference between a quadtree and a kd-tree?

3) Broad phase vs. narrow phase

a) In which phase of collision detection would we typically use a space partitioning algorithm as in task 2.2? Explain your answer.

4) Rotation, Torque

a) Imagine an object in outer space without gravitation or other forces acting on it. Is it possible to apply a single force to this object that does not induce a rotation? Explain your answer.

b) Which formula is used to compute the torque t acting on an object based on force f and the point of application of the force, p?

5) Friction

a) Which three values are used in Coulomb’s law? What is their meaning and what is the relation between them?

Why is it beneficial to work with a contact basis when calculating the effects of friction

Lecture 10 – Compression, Streaming and PCG

Relevant Topics

Texture Compression
General concept of block-based algorithms
DXT1 compression
How does it work?
Strenghts, weaknesses
Normal maps compression
Streaming
Course vs. fine-grained
MegaTextures (understand the basic concept)
Basic types of image filters and combinations thereof
Basic concepts of Perlin Noise

Example Problems

1) Why can PNG compression not be used for compressed textures?
2) Explain the difference between coarse and fine grained streaming.
3) Why is it not feasible to use a format such as JPEG for GPU textures?
4) What are design goals for a texture compression algorithm for GPU textures?
5) Why do texture compression algorithms work with less than 8bits per color?
6) Why does the RGB565 format make sense?
7) What are the two factors that lead to loss of information of DXT 1 compression?
8) What is the best case for DXT1?
9) What is the worst case for DXT1 compression?
10) What is the disadvantage of using 4x4 blocks in DXT1?
11) Why is Compression of Normal Maps different from diffuse textures?
12) How can we save a normal with only 2 values?
13) Name and describe on kind of multitexturing approach
14) Explain how MegaTextures work from a conceptual approach
15) Which is worse for MegaTextures: A high-speed racing game or a slow exploration game in which players walk? Why?
16) What is the advantage of sparse voxel octrees compared to uniformly sized voxel grids?
17) Describe one method of handling game objects in streaming worlds
18) Is it possible to create a game with streaming assets in a client-server setting?
a) How can the server handle streaming?
b) How can the client handle streaming?
19) How is a filter specified by a kernel applied to an image?
20) Explain the effects of changing the frequency and amplitude of Perlin Noise.

Lecture 11 - Multiplayer

Relevant Topics

Network architectures
Describe how they work, what data is sent, ...
Advantages/Disadvantages
Performance
Latency and its handling

Peer-to-Peer Lockstep
Determinisim
Cheating, Cheat Prevention

Client/Server

Client-Side Prediction

Network protocols
Basic properties of TCP and UDP

Example Problems

1) Compare the Peer-to-Peer and the Client/Server model – what are the pros and cons?

2) What is the problem with Client-Side Prediction and how can it be mitigated? **New Tasks**

a) Identification of Multiplayer Architecture

In Byte Magazine, Volume 5 No. 12 (1980), Ken Wasserman and Tim Stryker describe one of the first multiplayer games:

Putting It All Together

Just having the capability to transfer bytes back and forth between two machines does 'not guarantee success in writing multimachine games. We now need a general strategy for controlling the flow of information between the various machines in such a way that the moves made by each player are processed in a consistent manner by all machines involved. Among other things, the strategy used must ensure that all of the machines involved agree as to the order in which the various players' moves are to be processed. Only one such strategy, the key-oriented strategy, will be discussed here. Although many other ap- proaches to the problem do exist, this one is particularly "clean" and therefore easily debugged; it is also reasonably efficient in both space and time.

The information transfers addressed by any general strategy of this kind fall into two groups: those that occur at initialization time and those that occur during the ac- tual play of the game. The key-oriented strategy calls for all information pertinent to the initial state of the game, including information that may be kept secret from one or more players, to be made known to all machines at ini- tialization time.

Then, during play, a continuous conversation is set up among the machines in which the only information changing hands consists of individual keystrokes generated by the players at their keyboards. If a player generates no keystroke to be sent on a given pass, a zero byte is sent out to the other machine(s) to indicate this fact. Every machine maintains the full status of every player but only displays the information its own player is supposed to see.

Identify which multiplayer architecture is described in the article. Explain which indicators from the article informed your decision.

b) In a Peer-to-peer lockstep multiplayer game, is the frame rate the clients render at dependent upon the network latency?

c) Is the whole game state transmitted in a peer-to-peer lockstep multiplayer game?

d) Which information is transmitted in a peer-to-peer lockstep multiplayer game?

e) In a peer-to-peer lockstep multiplayer game, one client takes much longer than the other clients in sending commands. What is the effect on the other clients?

f) Is it possible to do prediction in peer-to-peer lockstep multiplayer?

g) A peer-to-peer lockstep multiplayer game is built to allow players with different platforms (PC, mobile, …) to play together. In the implementation, the platform's random function is used. What problems could arise in the game?

h) What information is sent in a client-server game?

i) A client is slow in sending commands in a client/server game. What is the effect on other clients?

j) Who calculates predictions in a client/server game with prediction? Why?

k) Which network protocol is usually used to send updates in a networked multiplayer game? Why?

l) Describe how a player can cheat in a lockstep multiplayer game.

m) In a lockstep multiplayer game, a client is cheating by waiting for the other commands and then reacting to them. How can this cheat be eleminated?

Lecture 12 - Sound

Relevant Topics

Structure of a low-level audio api
Sound mixing
Basic structure of music
Sound localization
Doppler Effect

Example Problems

1) What is the basic data structure used to send audio signals to the audio hardware?
2) How does an algorithm for sound mixing work?
3) What are the basic elements of a piece of music?
4) What algorithms does the brain use to differentiate sounds coming from the left and the right?

Lecture 13 - Artificial Intelligence

Relevant Topics

Model of AI

Understand the different layers and associated systems and explain how their relationships are

Movement

Understand kinematic vs. dynamic Behaviours (kinematic and dynamic)

- **Seek, Flee**
- **Arrive**
- **Wander**
- **Pursue, Evade**

Obstacle and wall avoidance – Problems, possible solutions Combination of behaviours – Blending, Priorities

Problems ��Equilibria

Pathfinding

Review basics of A* - no need to calculate

Methods for generating graphs: hand-made, tile-based, visibility points, nav-mesh Hierarchical pathfinding

Decision Making Decision Trees State Machines

Hierarchical State Machines Fuzzy logic, fuzzy rules

Goal-Oriented Behaviour, Goal-Oriented Planning

Tactical and Strategical AI Understand the purpose Execution Management Frequency and phase of tasks Wright's method

Interruptible, Anytime algorithms AI Level of Detail

Example Problems

2.1 AI Model

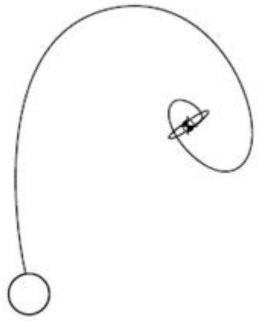
a) What are the three main levels of AI tasks that we looked at in the lecture?

b) Name one example of a method usually found on the tactical and strategic AI level.

2.2 Movement

a) What is the difference between a kinematic steering behavior and a dynamic steering behavior?

b) To which steering behavior can the diagram below belong? (Include kinematic or dynamic in your answer). The character is the circle in the lower left, the target is the point on the upper right. The line indicates the trajectory of the AI character. Explain your answer.

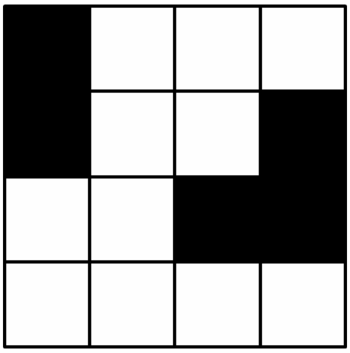


2.3 Movement

a) An AI uses blending for combining several steering behaviors. Construct a situation in which an equilibrium is found and the AI is in a deadlock (i.e. will be stuck unless the conditions around it change). You may draw or describe the situation.

2.5 Pathfinding

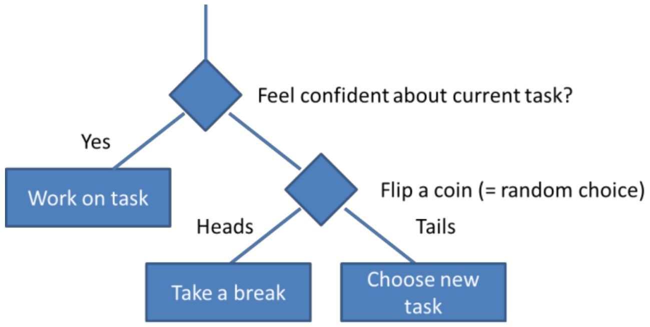
a) A game uses a tile-based map. Below, you can see a subset of this map. Explain how you could use the tile- based design of the map to find a navigation graph automatically. Black squares are blocked, white squares are walkable.



b) Explain the concept of hierarchical pathfinding.

2.6 Decision Making

a) The following decision tree describes the behavior of a student AI in an exam. If it were implemented in a game, what problem could arise? (The decision tree is evaluated often, e.g. every 5 frames).


Feel confident about current task?
Yes
Work on task
Flip a coin (= random choice)
Heads
Tails
Take a break
Choose new task

2.7 Fuzzy Logic

In an RPG, a healing character has the following fuzzy state variables:

(A) self-injured: 0.2
(B) closest-teammate-injured: 0.8
(C) egoistic: 0.3

We want to derive the two following states: cast-heal-on-self
cast-heal-on-teammate

The rules for this are:
self-injured AND egoistic THEN cast-heal-on-self
closest-teammate-injured AND NOT egoistic THEN cast-heal-on-teammate

a) How are the rules given above formulated using fuzzy logic? cast-heal-on-self =

cast-heal-on-teammate =

b) What are the resulting state variables? cast-heal-on-self =

cast-heal-on-teammate:

2.8 Goal-oriented behaviour

a) What is the advantage of a character that uses Goal-Oriented Action Planning over a character that chooses actions one at a time?

2.8 Execution management

a) What is the purpose of applying a phase to a recurring AI task?