exam

Anton Gerdelan gerdela@scss.tcd.ie

Trinity College Dublin

outline

- Worth 80% of course grade
- Questions very similar to previous 2 exams
- Most lecture topics covered
- Not required to write
  - OpenGL code
  - GLSL code
- Not required to memorise
  - mathematical functions
  - matrix layouts
- I re-organised the structure a bit

structure

- 6 “questions” (topics) worth 25 marks each
- Choose any 4 of the 6
- Each “question” has 4 parts
  - 1 basic practical question (5 marks)
  - 1 basic theory question (5 marks)
  - 1 problem to solve (10 marks)
  - 1 test of deeper knowledge (5 marks)

what you should revise

- Go over all the lecture topics
- Major themes that we covered
  - the graphics **hardware pipeline** and **shaders**
  - **linear algebra** and geometric problems (dot and cross products)
  - transformation pipeline (**matrices**)
  - virtual **cameras**
  - **texture-mapping**

what you should revise

- More major themes that we covered
  - **lighting**
  - **animation**
  - modelling - no Blender-specific questions
  - multi-pass rendering - no specific question
  - **optimisation** and **scene management**/data structures
  - splines and curves – no specific question

you will be asked

- “draw a diagram of”
- “write pseudo-code for”
- “write a shader”
- “explain why this uses that”
- “explain how you would solve this problem”
- “give 2 features/advantages/algorithms that”
- “give equations for”
- “calculate the result of”

intended grading scheme

- significant gaps in knowledge < 50%
- basic working knowledge of computer graphics – 50%
- can solve some variations to common problems – 60-70%
- broad knowledge of basic theory and practice – 80%
- has also thought about side-topics, has more advanced theory – up to 100%

actual grading scheme

- is good at doing exams – 70%+

sample basic Q (5 pts)

- “What is the purpose of using **two colour buffers**, where one is to draw into while the other is being displayed?”

sample basic Q (5 pts)

- “Explain each of the following terms:
  - i. Isometric projection
  - ii. Clipping”

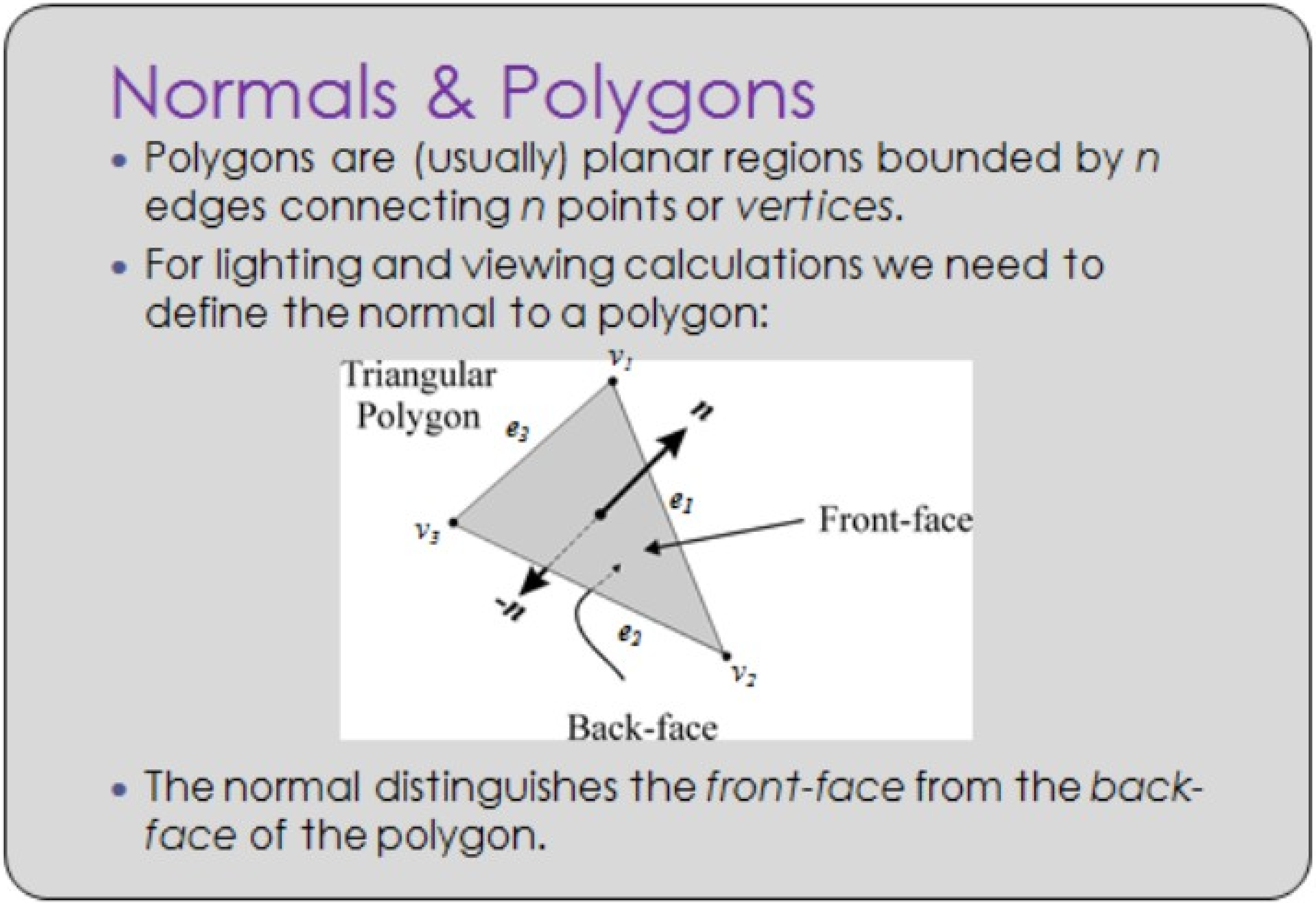
sample practical problem (10 pts)

- “Devise a method for testing whether one planar polygon is fully on one side of another planar polygon.“

sample practical problem (10 pts)

- “Devise a method for testing whether one planar polygon is fully on one side of another planar polygon.“
- How to do yourself a favour
  - Use a diagram
  - Use some psuedo-code or maths equations
  - Add some explanation
- No answer at all (0 pts)
- Some correct parts (some pts)

answer to polygon question

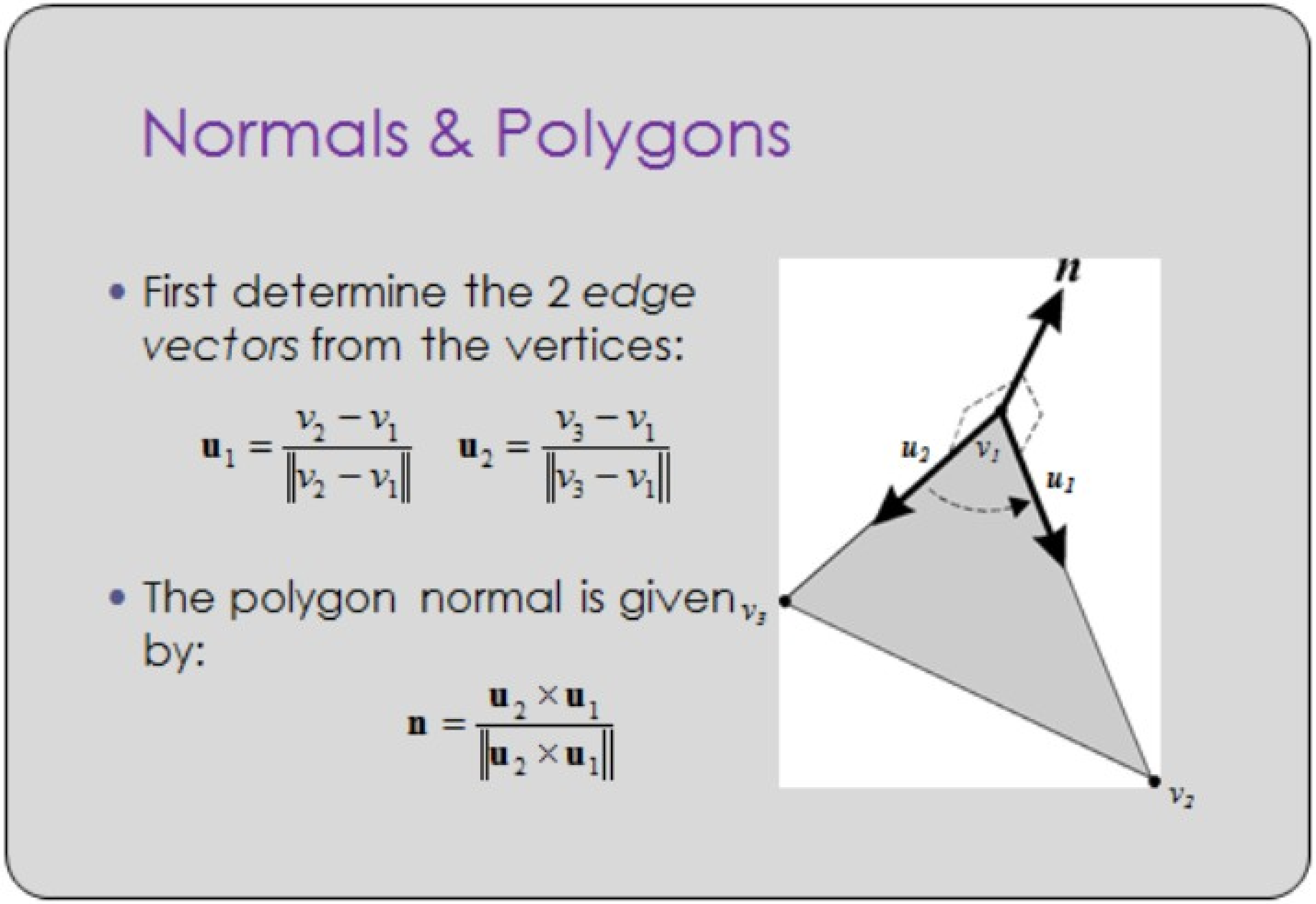


Normals & Polygons

- Polygons are (usually) planar regions bounded by *n* edges connecting *n* points or *vertices*.
- For lighting and viewing calculations we need to define the normal to a polygon:

- The normal distinguishes the *front-face* from the *back-face* of the polygon.

answer to polygon question



Normals & Polygons

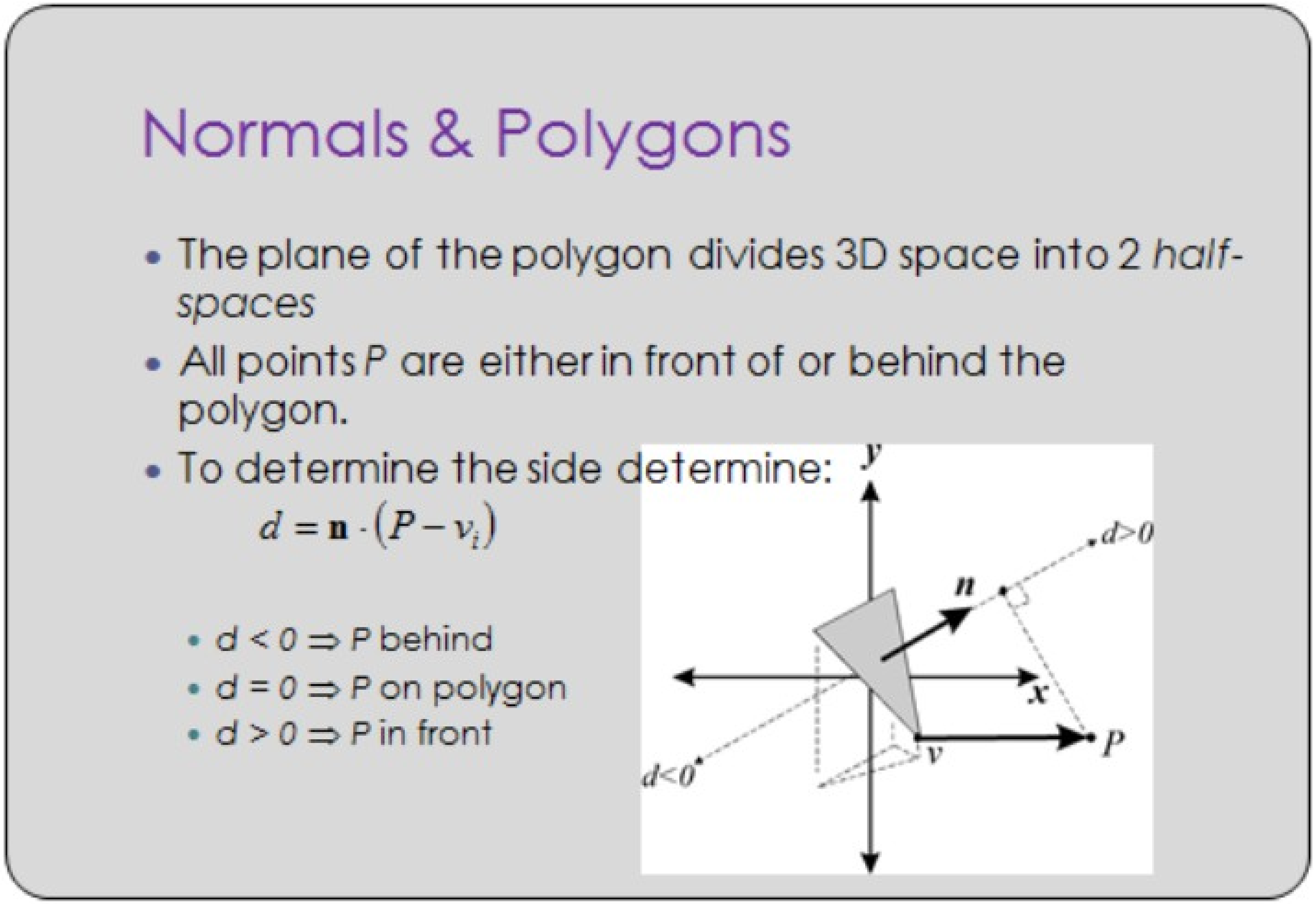
- First determine the 2 *edge vectors* from the vertices:

$$\mathbf{u}_1 = \frac{v_2 - v_1}{\|v_2 - v_1\|} \quad \mathbf{u}_2 = \frac{v_3 - v_1}{\|v_3 - v_1\|}$$

- The polygon normal is given by:

$$\mathbf{n} = \frac{\mathbf{u}_2 \times \mathbf{u}_1}{\|\mathbf{u}_2 \times \mathbf{u}_1\|}$$

answer to polygon question



Normals & Polygons

- The plane of the polygon divides 3D space into 2 *half-spaces*
- All points *P* are either in front of or behind the polygon.
- To determine the side determine:

$$d = \mathbf{n} \cdot (P - v_i)$$

  - $d < 0 \Rightarrow P$ behind
  - $d = 0 \Rightarrow P$ on polygon
  - $d > 0 \Rightarrow P$ in front

sample deeper understanding Q (5 pts)

- “What does a perspective matrix do to the **w** coordinate of a 4d vector when multiplied in the form **P * v**?”

prepare yourself

- There are always questions about vectors
- Phong reflection uses a lot of vectors
- Know how to draw diagrams for algorithms like Phong
- Know what the most important parts of equations are
- Be able to list techniques or alternatives for major topics
- Most practical problem Qs are variations on lab problems
- Be able to talk about a few side-topics or advanced algorithms

previous exams

- I'll upload them if you can't get them
- some of the code questions use older OpenGL
- answer the code questions with pseudo-code or maths
- I don't ask you to give any matrix contents
  (I can't remember them – I have a cheat-sheet)
- I do expect you to know how different matrices and maths functions work

potential problems

- “I don't know the intended answer, but have another idea”
  - try your idea anyway, we have anticipated many alternative answers
- Mental block / forgot a term
  - draw a little diagram and add description to explain
- “This question doesn't make sense!”
  - ask the adjudicator (I will be available also)

potential problems

- “This question is too hard/obscure”
  - write down anything you know – a sentence or two may get some marks
- “I'm not familiar with this term”
  - If it's a final 5pt expert question – guess! If not – ask!
- “I'm terrible at exams and I ruined my grade”
  - I'm bad at exams, but pretty good at graphics!

like gfx? keep your exam script

- I deliberately asked questions that I know are in job interviews
- I deliberately asked questions that are asked in other university courses' final graphics exams
- I feel like this makes the exam more valuable to you than the evaluation grade for a rapid-pace intro course

Q?