

# *Interactive Computer Graphics*

Prof Abhijeet Ghosh and Dr Bernhard Kainz

[abhijeet.ghosh@imperial.ac.uk](mailto:abhijeet.ghosh@imperial.ac.uk)

[b.kainz@imperial.ac.uk](mailto:b.kainz@imperial.ac.uk)

# Intro

- <https://www.youtube.com/watch?v=wAu8w7n4LHM>

lecture



Dr Bernhard Kainz  
[b.kainz@imperial.ac.uk](mailto:b.kainz@imperial.ac.uk)



Prof Abhijeet Ghosh  
[abhijeet.ghosh@imperial.ac.uk](mailto:abhijeet.ghosh@imperial.ac.uk)

coursework CSL



Matthew Baugh + GTA team  
[matthew.baugh17@imperial.ac.uk](mailto:matthew.baugh17@imperial.ac.uk)

Arvin Lin, Hadrien Reynaud,  
Liu Li, Sarah Cechnicka, Yiming Luo

# *Interactive Computer Graphics*

- Please note that this course has been timetabled for 2 hours per week:
  - Friday 11-12, Huxley 311 and MS Teams
  - Friday 12-13, MS Teams Lab queue
- However, not all timetabled slots will be used every week so **please check the timetable** on the webpage for more information:

<http://wp.doc.ic.ac.uk/bkainz/teaching/60005-co317-computer-graphics/>

# *Interactive Computer Graphics*

- Printouts:
  - Lecture notes & tutorials:
    - Please print your own if you want a hardcopy
- Lectures:
  - All lectures have slides available via <https://scientia.doc.ic.ac.uk/2324/modules/60005/materials>
  - Some lectures have orthogonal notes that are available via <https://scientia.doc.ic.ac.uk/2223/modules/60005/materials>
  - Lectures are partly flipped classroom with pre-recorded high-quality lectures you might want to watch during the week, recap and discussion during class sessions.
- Tutorials:
  - All tutorials have sample solutions that are available a few days after the tutorial on <https://scientia.doc.ic.ac.uk/2324/modules/60005/exercises>

# *Interactive Computer Graphics*

- Course overview:

- Syllabus, timetable and news on

<http://wp.doc.ic.ac.uk/bkainz/teaching/60005-co317-computer-graphics/>

- See notes on vector algebra revision ([link](#))

- Course materials and notes:

- Look at Scientia for lecture notes, tutorials & coursework

# *Information for non DOC students*

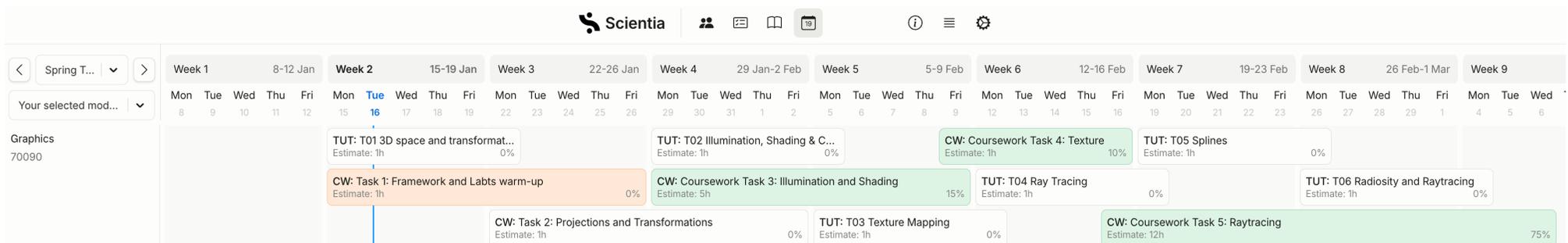
- Apply at <https://dbc.doc.ic.ac.uk/externalreg/>
- Your department's endorser will approve/reject your application
  - No access after a few days? Check status of approval and contact relevant person(s)
- Key Dates:
  - Exam registration opens end January for 2-3 weeks
  - Exams for DoC 3rd/4th year courses take place at the end of the Term in which the course is taught – courses that are co-scheduled on the time-table will have their exams co-scheduled
- If in doubt, read the guidelines available at the link above

# Courseworks

- There will be six practical coursework tasks; three of them are assessed:
  - Task 1: Framework
  - Task 2: Transformations
  - Task 3: Illumination (assessed 15%)
  - Task 4: Color
  - Task 5: Texture & Render to Texture (assessed 10%)
  - Task 6: GPU ray tracing (75%)
- All practical courseworks require programming experience (very basic C) and google skills

# Logistics

- 6 tasks, 3 assessed
  - 1-5 One per week
  - Task 6: 2 weeks
- Description and framework already available for all exercises, but
- Necessary knowledge in each lecture per week
- Submission electronically via LabTS! - Matt



# *Effects previous years' student feedback and COVID-19*

- Redesigned the coursework to better match the content of the lectures in each week
- Made the framework available to everybody through a browser implementation (no computing lab requirement anymore)
- Provide an open-source implementation of a custom OpenGL GLSL IDE
- Reduced the workload to three assessed tasks, revising assessment. Tasks 1,2,4 are voluntary. Removed one task to reduce workload
- One exam question will be based on what you learned during the coursework!
- LabTS submissions with automated immediate feedback
- Re-implemented the framework for the coursework for a second time:
  - It is now the most advanced teaching framework for computer graphics
  - updated to WebGL 2, which provides most recent programming features
- Everybody can use it now from their own laptops without needing to install anything: <http://shaderlabweb.doc.ic.ac.uk/>
- We listened to your feedback from the last years!
  - Please fill in the feedback questionnaire at the end of this course!

# *CSL and TAs*

- Educational support leader and GTAs:
  - Matthew Baugh [matthew.baugh17@imperial.ac.uk](mailto:matthew.baugh17@imperial.ac.uk)

## *Labs in MS Teams Lab queue/311*

- Week 3: Friday 12-13
- Week 4: Friday 12-13
- Week 6: Friday 12-13
- Week 8: Friday 12-13

# framework

- <http://shaderlabweb.doc.ic.ac.uk/>

The screenshot displays the ShaderLabWeb web application interface. At the top, the browser address bar shows the URL <http://shaderlabweb.doc.ic.ac.uk/>. The page header includes the logo "ShaderLabWeb" and "Imperial College London" on the left, and a "log in" link on the right.

The main content area is split into two panels. The left panel shows a 3D rendering of a red teapot model against a black background. The right panel displays the GLSL shader code for the teapot, organized into two tabs: "base" (with sub-tabs for "vertex" and "fragment") and "Quad/R2T" (with sub-tabs for "vertex" and "fragment"). The code is as follows:

```
1 //vertex position in world coordinates
2 attribute vec3 vertex_worldSpace;
3 //surface normal at the vertex in world coordinates
4 attribute vec3 normal_worldSpace;
5 //texture coordinates at that vertex
6 attribute vec2 textureCoordinate_input;
7
8 //model Matrix (Identity in our case)
9 uniform mat4 mMatrix;
10 //view Matrix
11 uniform mat4 vMatrix;
12 //projection Matrix
13 uniform mat4 pMatrix;
14
15 //main program for each vertex
16 void main() {
17     vec4 vertex_camSpace = vMatrix * mMatrix * vec4(vertex_worldSpace, 1.0);
18     gl_Position = pMatrix * vertex_camSpace;
19
20
21 }
```

Below the 3D view, there is a "Log" section with tabs for "Camera", "Model", and "Uniforms". The log entries show the successful compilation and linking of the vertex and fragment shaders for both the "base" and "Quad/R2T" models.

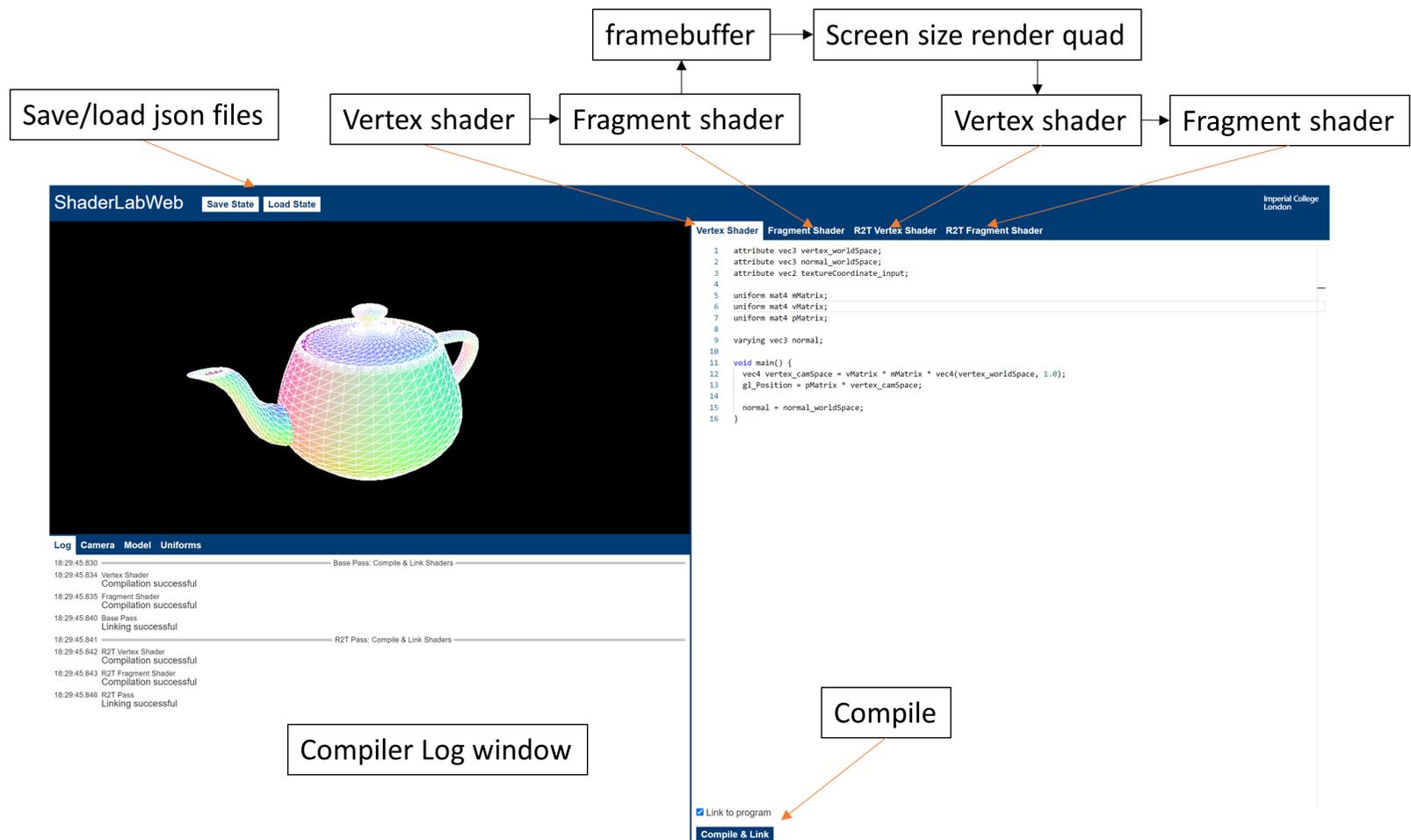
At the bottom of the interface, there are buttons for "Download State: .json .zip", "Upload State: .json .zip", and a "Compile & Link" button.

## *starting the framework*

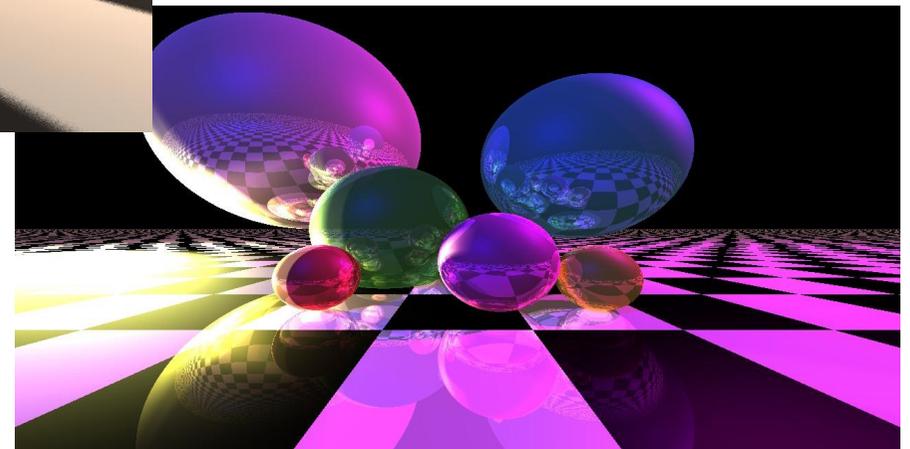
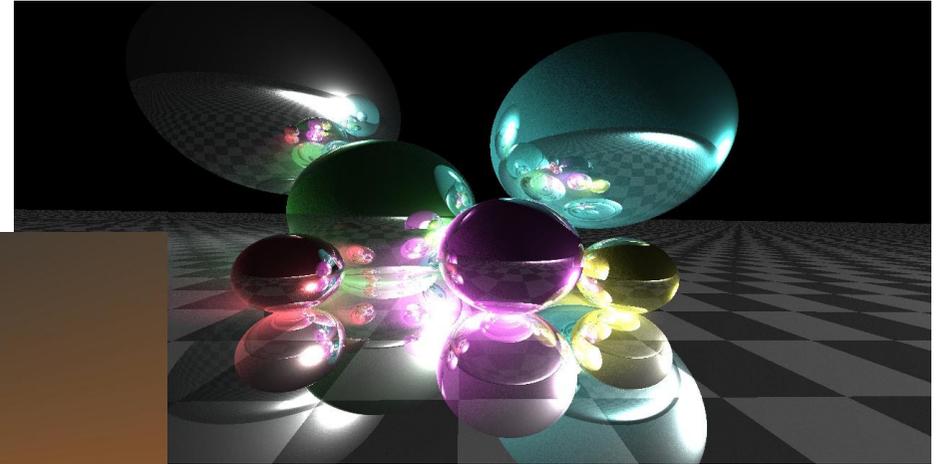
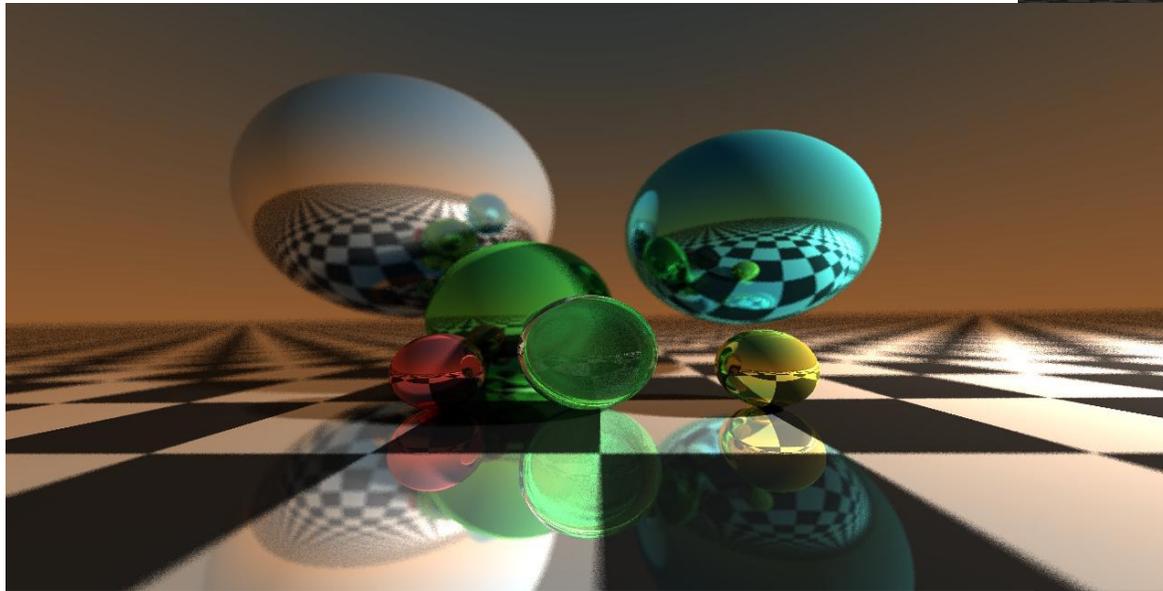
- Open a browser (preferably Google Chrome)
- Enter <http://shaderlabweb.doc.ic.ac.uk/>

# Tasks

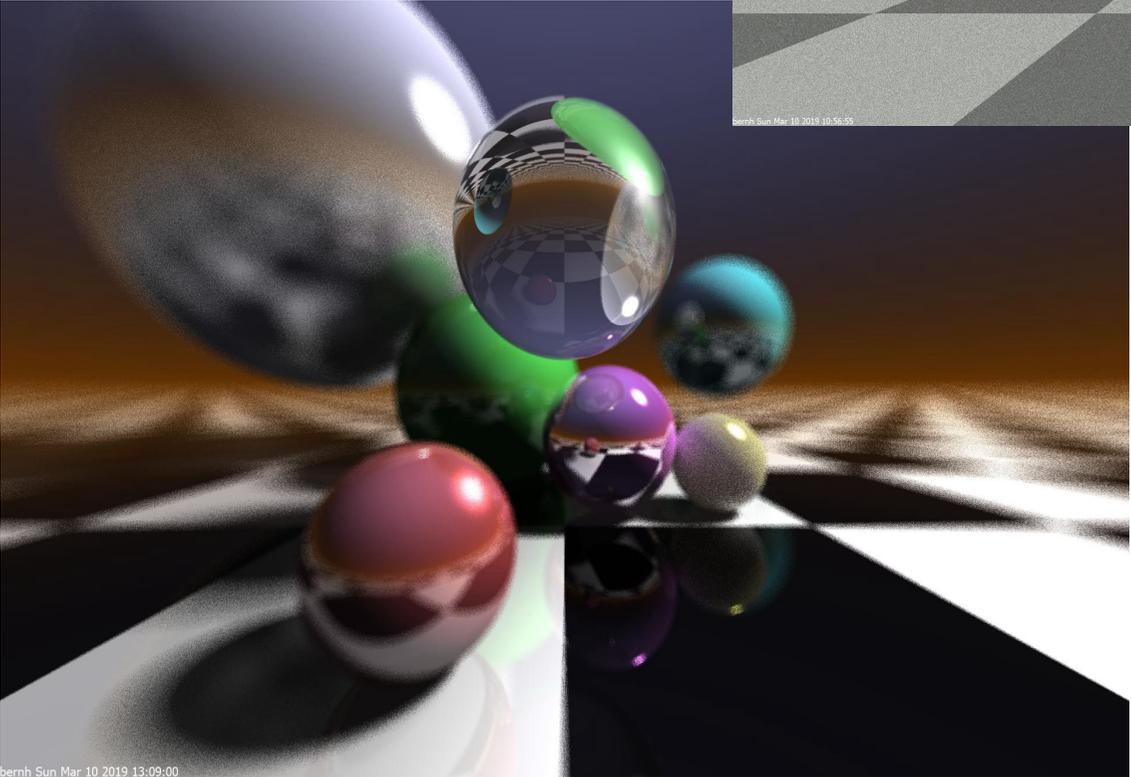
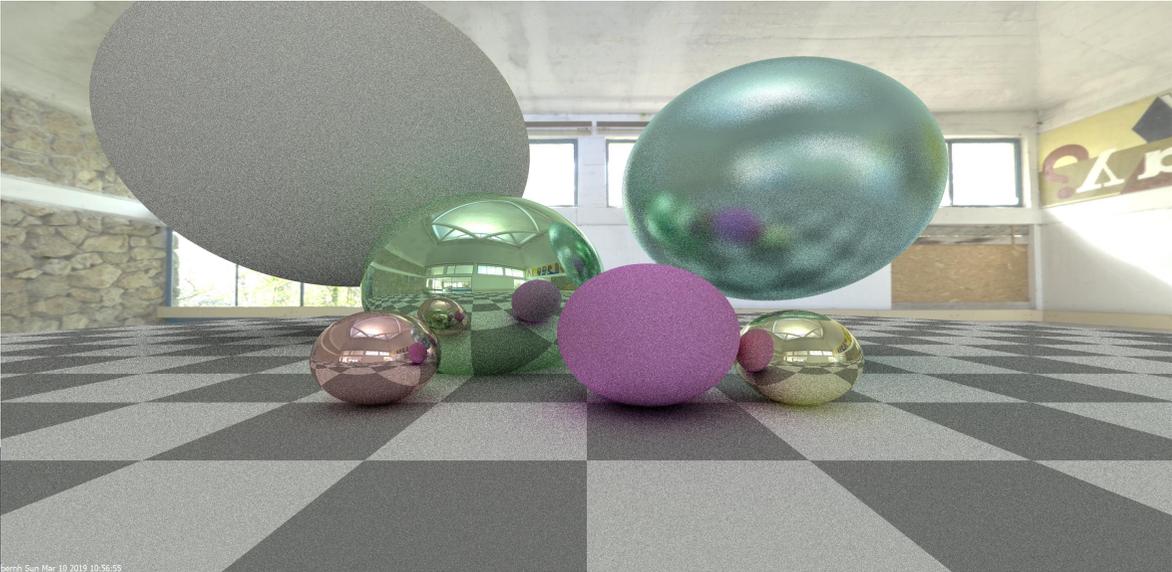
- Task 1:  
Get familiar with the framework



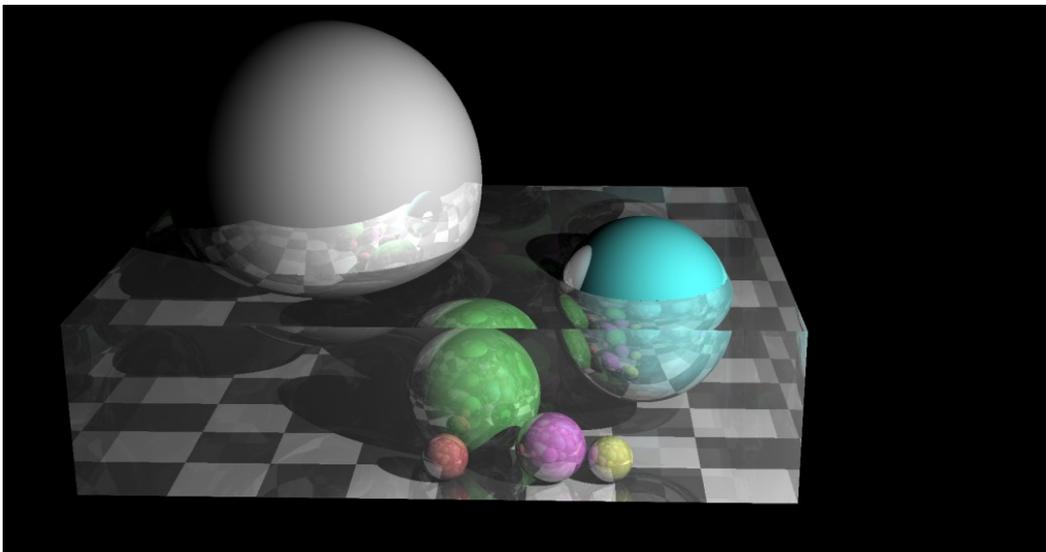
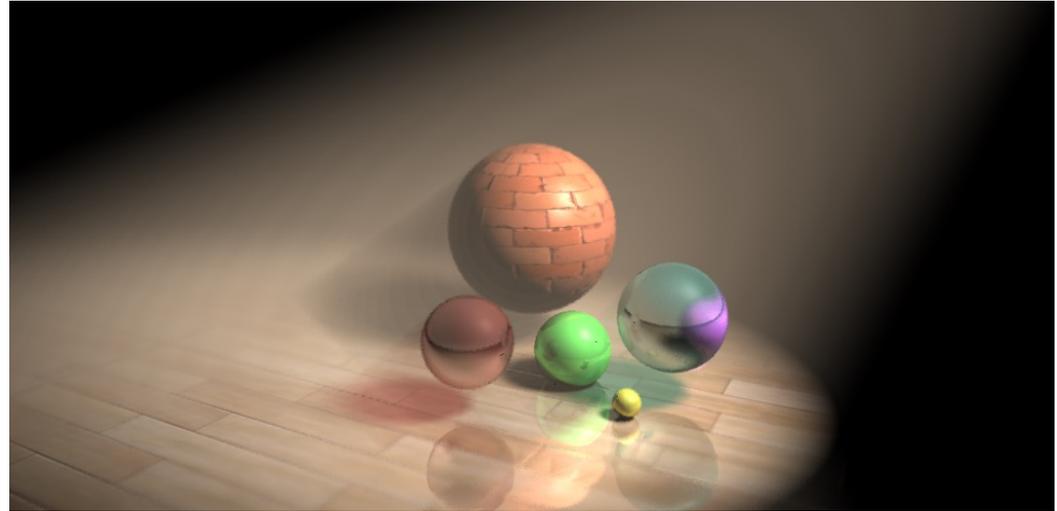
# Task 6: Student solutions



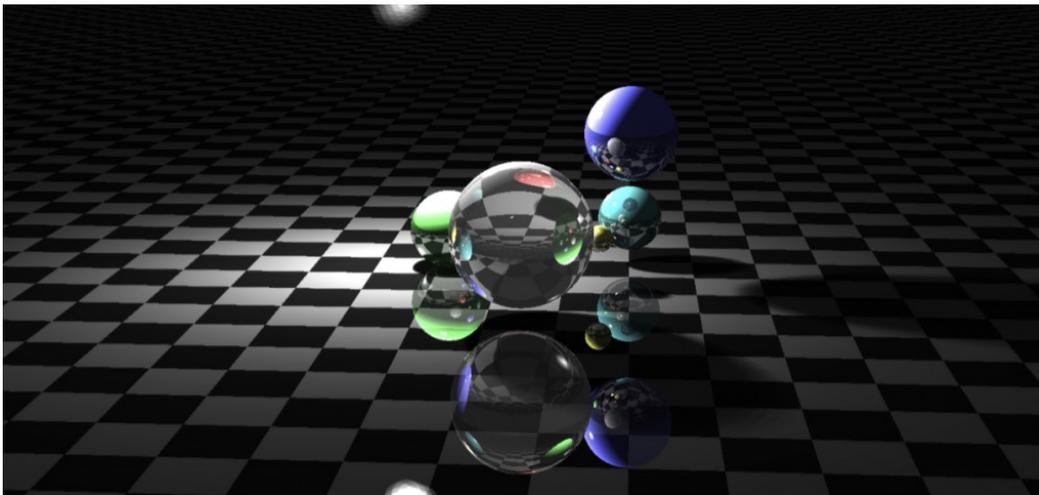
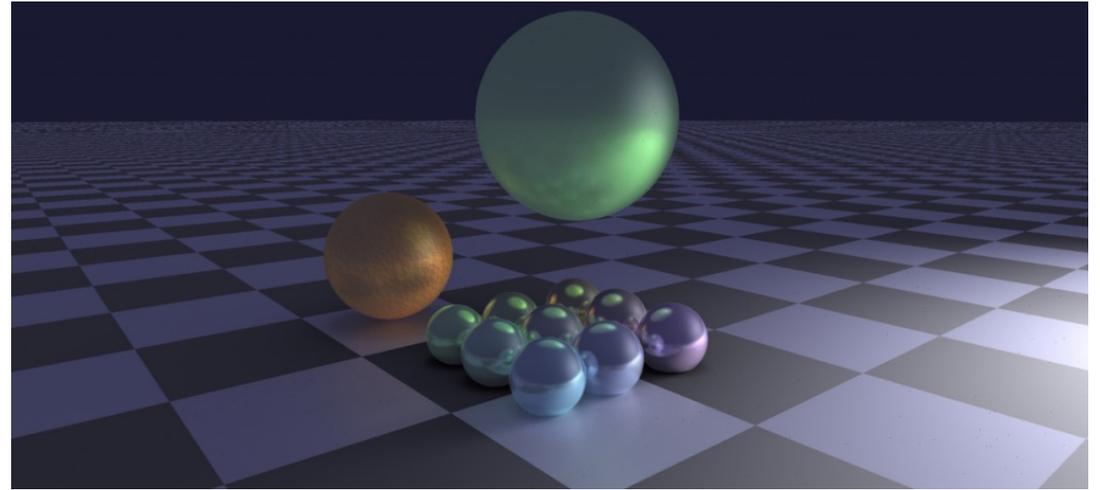
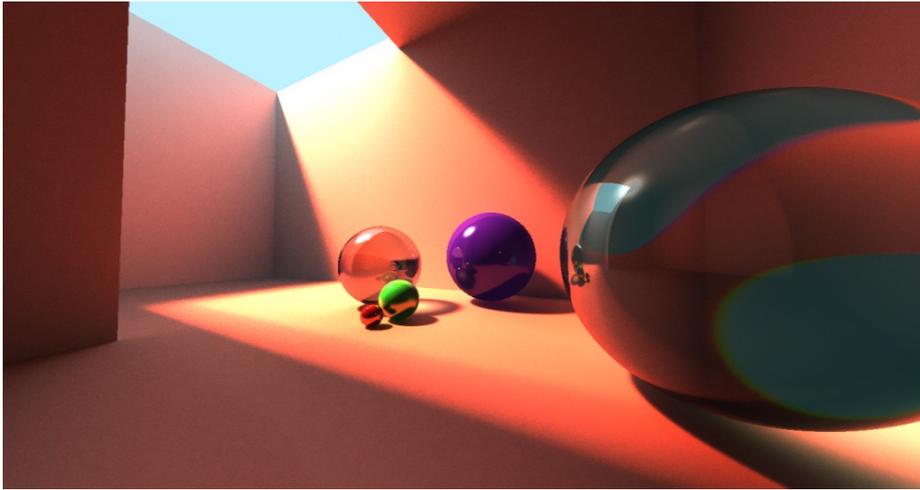
# *Student solutions*



# *Student solutions*



# *Student solutions*



***Questions:***

<https://edstem.org/us/courses/46822/discussion/>

**Have fun!**