**Lesson Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Teacher**: | **Class/Set/Ability Group**: | | | **Date**: |
| **Lesson**: GPU | **No of Boys**:  **No of Girls**: **Other**: | | | |
| **Subject: Computer Science** | **Department: Computer Science** | | | **Total:** |
| **Prior Attainment Summary**:  Should have covered the Structure of the CPU topic completely.  Prior knowledge of MIMD and SIMD  Prior knowledge of JavaScript syntax | | **Main Objectives (OCR A-level Spec)**:  A-Level – Types of processors – GPU, function, purpose and practical use of. | | |
| **Lesson Objectives/Learning Intentions**:   * To understand the differences between the CPU and the GPU particularly parallelism. * Be able to give examples of uses for GPUs * To demonstrate the use of GPUs through a programming example. | | | | |
| **Learning Outcomes**:   * Students have documented a summary of the GPU * Students have done a practical activity making use of the GPU and how its parallel power is far superior to the multicore CPU. * Students can offer examples of other uses of the GPU outside of graphics. | | | | |
| **Resources**:  ShaderToy.com  [https://editor.p5js.org/jonas.dewanckel/](https://editor.p5js.org/jonas.dewanckel/sketches/IJvyUK5su) | | **Key Words**:  GPU, CPU, SIMD, MIMD, Parallelism, Shader core | | |
| **Opportunities to Develop Literacy; Speaking & Listening, Reading and Writing Skills**:  Technical keywords from this topic | | | | |
|  | | | | |
| **Lesson Progression** | | | **Assessment to Support Learning** | |
| 1. **Starter** –Introduction to session, GameChangers and Avalanche Studios (slides 1-8 including presenter notes). [5 mins] 2. Introduce the GPU using slides 9-13 (Slide 13 has a long video, it’s good, but most of it is not relevant for the A-level, its worth showing times 0:00-2:00, 7:12-8:30(or the end). [5-10mins] 3. Activities, on worksheet [30mins]    1. For this exercise they should add code to create a light source to the scene. Point the students to earlier code that has been used to create a circle.   Solution  float light\_ball = Circle(uv, mouse\_coord, 0.03, 0.0);  then after the line setting col  col += light\_ball \* vec3(1., 1., 1.);   * 1. For this exercise they need to apply a shadow to the main mask image from the light source. This is achieved by multiplying the dot product of a unit vector with the mouse coordinates. The result of this is then multiplied with the mask.   Solution  float light = clamp(dot(normalize(mouse\_coord), uv), 0.0, 1.0);  mask \*= light;  multiplying values above 1.0 before the dot will brighten the light, whilst adding to the clamp can add ambient light to the whole scene see below:-  float light = clamp(2.0 \* dot(normalize(mouse\_coord), uv), 0.0, 1.0) + 0.1;  mask \*= light;   * 1. This exercise is to experiment with the unit vector uv, and apply transformations to the main image. Try introducing sin and cosine to the unit vector.   Solution  Do this near the top after uv has been assigned.  uv.x += 0.01f \* sin(uv.y\*100.f);  Play with the values of 0.01f and 100f to see the effect.   1. Return to slides to show advanced uses of the ShaderToy, and show how this is still only JavaScript code. [5mins] 2. Class discussion on alternative uses for the GPU. Within graphics and outside of graphics. [5-10mins] | | | One note and PowerPoint on screen.  Students should be taking notes to summarise what has been discussed  Students working on practical activity to experiment with the ShaderToy tool.  Plenary questions on worksheet. | |
| **Homework/Independent Learning Task.** | | | | |
| **SEN/Able/Dyslexia/Pupil Premium Considerations**: | | | | |