

Osnova 6. cvičení v C++

Milí studenti a studentky předmětu PV112. Tato osnova slouží jako pomůcka pro opakování *po cvičení*, například když něco nestihnete. Rozhodně neslouží k tomu, abyste si ji prošli už *před cvičením* a zkusili všechno naprogramovat, potom byste se na cvičení nudili a to nikdo nechce :-).

- Použití interface bloků pro proměnné z VS do FS v shaderech pro kostku
 - Seskupit do VertexData
 - Použít instance_name inData a outData
 - Odstranit prefix VS_
 - Úkol pro studenty: udělat totéž i pro konev
 - Celkem cca 15 minut
- Použití interface bloků pro uniformní proměnné
 - V C++ kódu jsou připraveny struktury s daty a proměnné. Jen je třeba vytvořit buffery pro OpenGL, aktualizovat je, upravit shader a navázat buffery jako vstupní data.
 - Upravuji paralelně oba shadery, pro kostku i pro konev, nejprve světlo
 - Seskupím do jednoho interface, nesmím zapomenout na std140
 - Získám index LightData bloku a nastavím binding point 0
 - V C++: ukázat strukturu Light, vytvořit UBO, aktualizovat data, navázat buffer
 - Zakomentovat nastavování dat přes uniformní proměnné
 - Cca 15 minut
 - Úkol pro studenty: použít instance_name light (5 minut)
 - Úkol pro studenty: udělat totéž pro kameru, dát dohromady projekční matici, pohledovou matici a pozici oka, použít binding point 1, spoustu věcí stačí odkomentovat (15 minut)
 - Celkem cca 35 minut
- Instancování
 - Nejprve kostka, vysvětlit princip a cíl: naráz vykreslit 100 kostek, 10x10, postup: data všech kostek do UBO. U kostek měním pouze pozici
 - Připravit strukturu CubeData v VS se 100 maticemi, použít gl_InstanceID
 - V C++: struktura odpovídající CubeData, připravit data, získat index, nastavit binding point na 2, navázat, použít glDraw*Instanced
 - Cca 20 minut
 - Úkol pro studenty: totéž pro konvičku, každá instance má pozici a barvu, použít binding point opět 2 (20 minut)
 - Celkem cca 40 minut