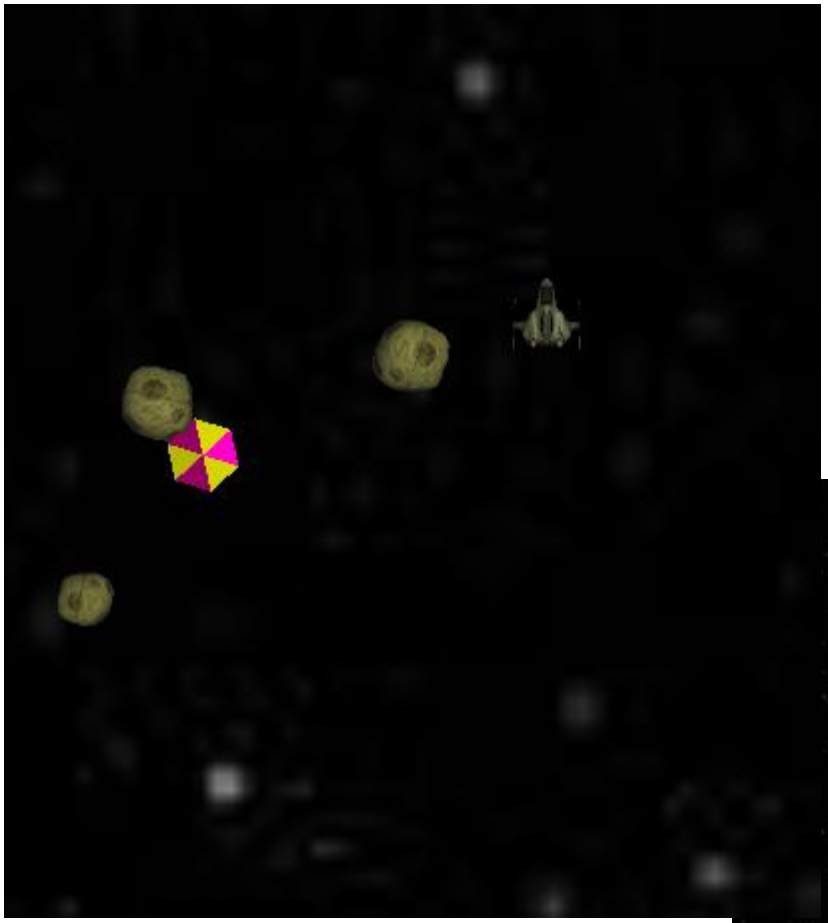
DCGI

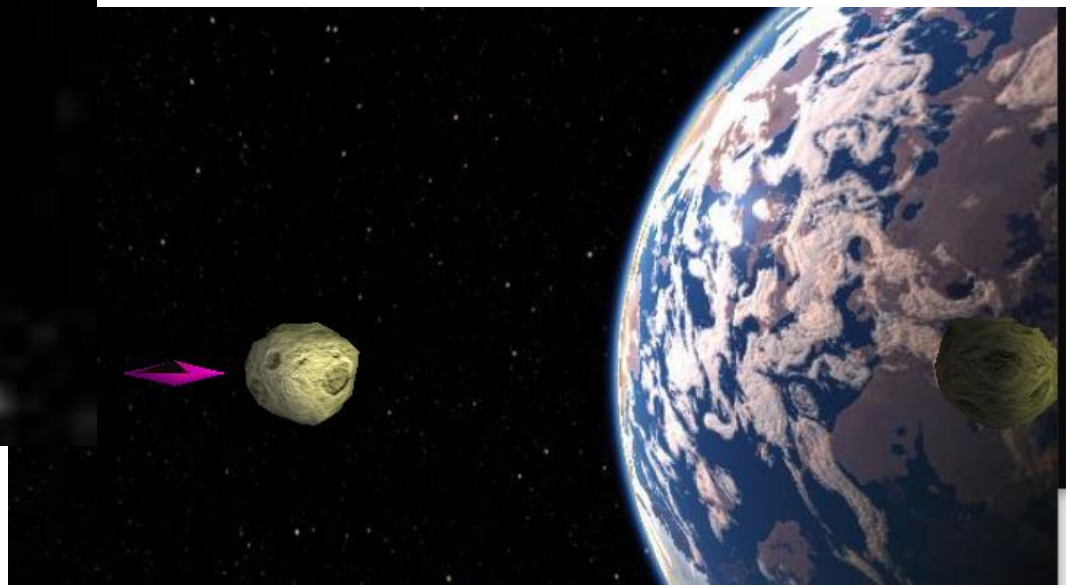
Cvičení I

Kreslení objektů složených
z grafických primitiv

Hra Asteroidy



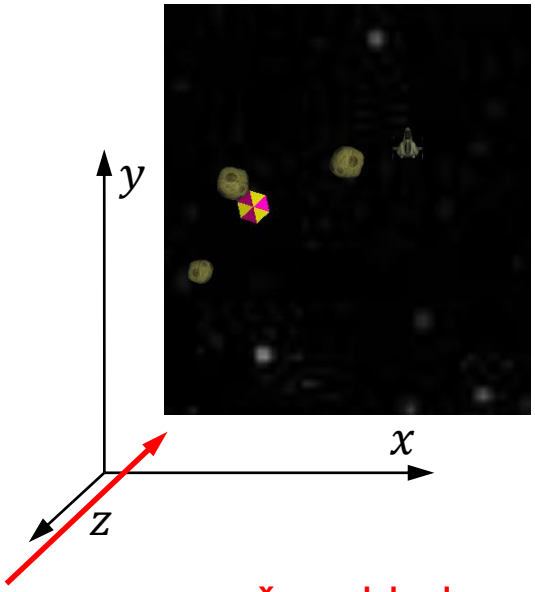
3D střílecí hra s cílem zničit všechny asteroidy a ufa



Hra Asteroidy

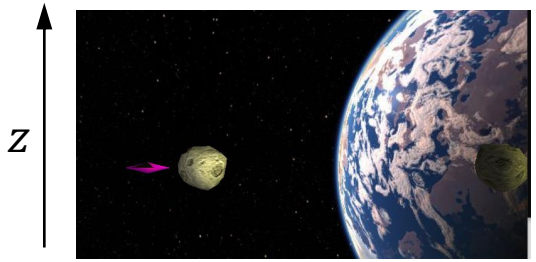
- všechny objekty scény se pohybují v rovině xy

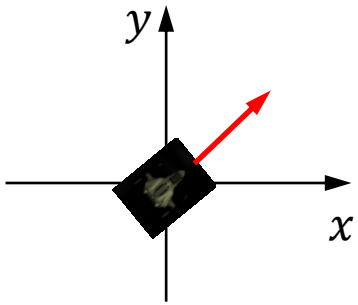
statický pohled na celou scénu



směr pohledu
osa -z

dynamický pohled z rakety (budeme dělat příště)





směr pohledu
=
směr natočení rakety v rovině xy (z=0)

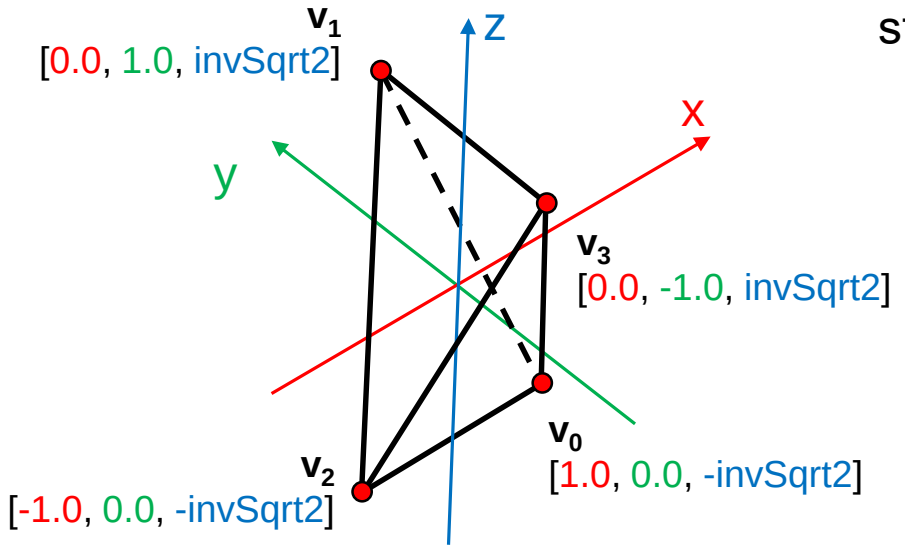
Hra Asteroidy

- objekty ve scéně

raketa asteroid ufo výbuch střela

- každý objekt má vlastní (viz soubor render_stuff.cpp):
  - kreslící funkci - např. *drawSpaceShip()*
  - inicializační funkci – vytvoření vao, vbo a ebo - např. *initMissileGeometry()* či *loadSingleMesh()*
  - strukturu *MeshGeometry* popisující geometrii objektu (vao, vbo, ebo a počet trojúhelníků) – např. instance pro ufo se jmenuje ufoGeometry
- každá instance objektu ve scéně má vlastní stavovou strukturu, např. *AsteroidObject* (viz render_stuff.h)
  - odvozena ze struktury Object (obsahuje pozici, rychlost, směr pohybu)
  - instance všech objektů jsou uloženy v seznamech ve struktuře GameObjects definované na začátku asteroids.cpp

Střela – vzorový příklad (TASK 1_1)



střela = jehlan (rovnostranné trojúhelníky)

missileGeometry->numTriangles = 4;

všechny trojúhelníky při pohledu zvenčí vrcholy uspořádány v protisměru hodinových ručiček

(pole missileVertices)
vbo

souřadnice vrcholů ▲ barvy vrcholů ▲

v_3	v_0	v_1	v_2	v_3	v_1	v_0	v_2	v_1	v_2	v_0	v_3	c_3	c_0	c_1	c_2	c_3	c_1	c_0	c_2	c_1	c_2	c_0	c_3

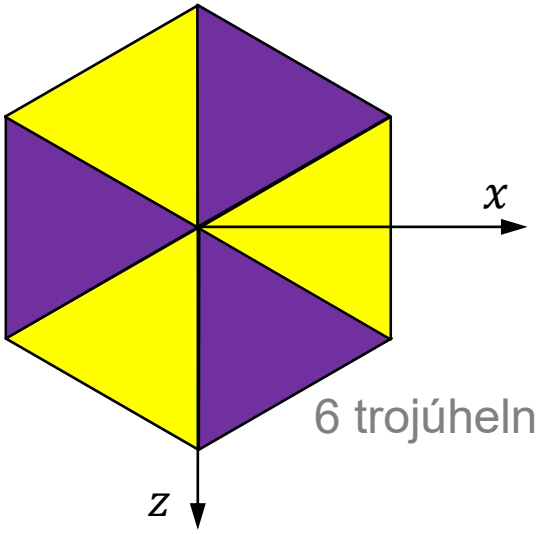
index prvního vrcholu

počet vrcholů

glDrawArrays(GL_TRIANGLES, **0**, missileGeometry->numTriangles*3);

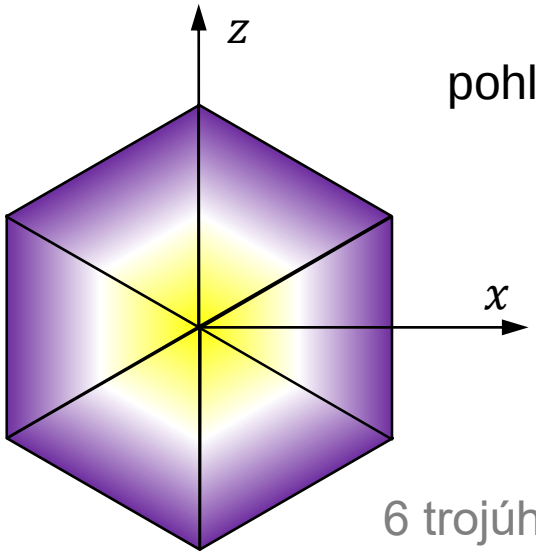
Ufo - geometrie objektu

pohled shora

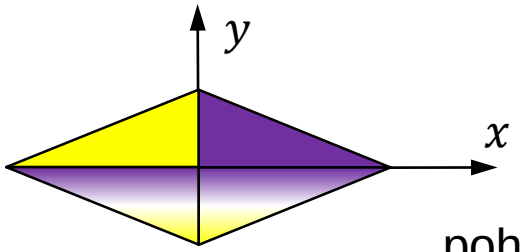


6 trojúhelníků

pohled zdola

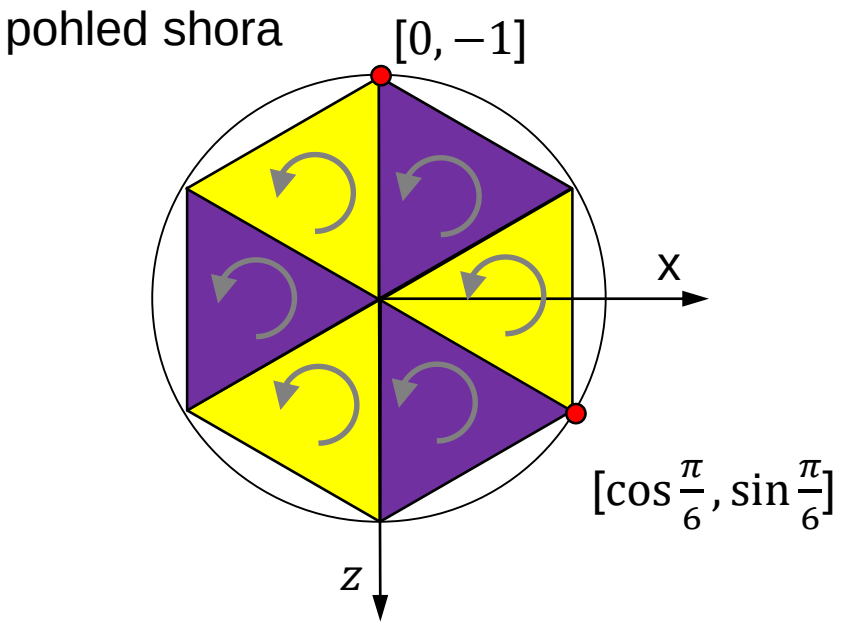


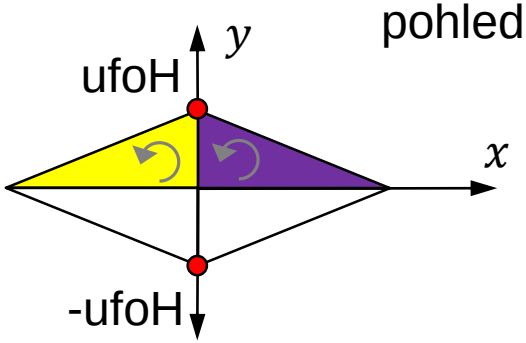
6 trojúhelníků



pohled z boku

1 úloha – horní část ufa (TASK 1_2)



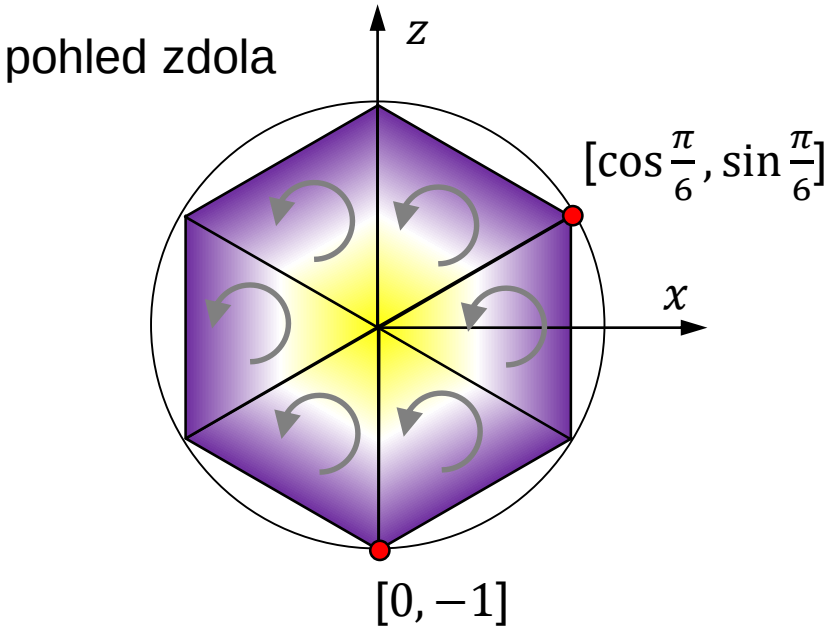


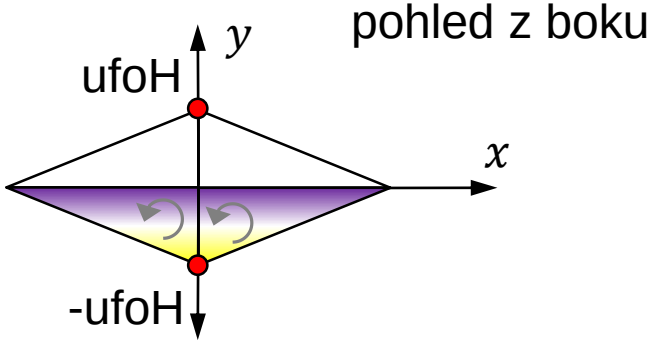
I. naplnit pole **ufoVertices** (*prokládané pole – pozice a barva*) – nejprve žluté a pak fialové trojúhelníky (6*3 vrcholů, vrcholy ▲ se opakují)

II. vytvořit vao a vbo, naplnit vbo z pole ufoVertices + napojit buffery na shadery pomocí glVertexAttribPointer()

III. vykreslit trojúhelníky pomocí **glDrawArrays()**

2 úloha – spodní část ufa (TASK 1_3)





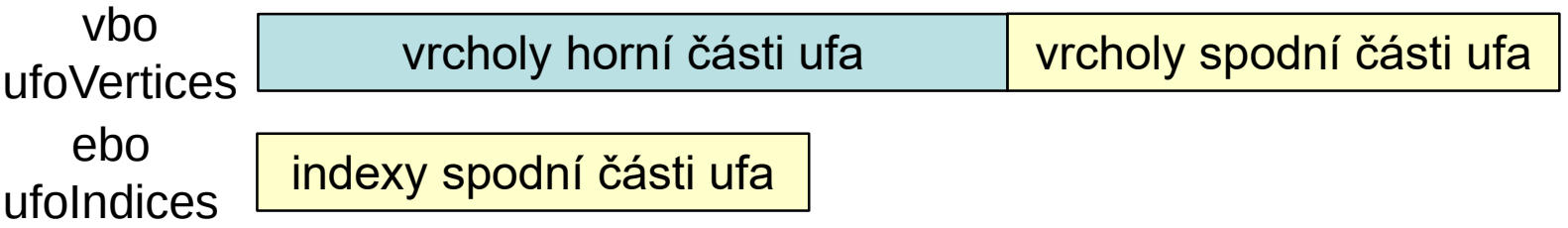
I. doplnit na konec pole **ufoVertices** nové vrcholy (6+1) spodní části a vyplnit pole indexů **ufoIndices** (vrcholy se neopakují, indexy ano →18x; pozor na začátek indexů spodní části, aby ukazovaly na nové vrcholy)

II. vytvořit a naplnit ebo z pole ufoIndices, vao a vbo sdílené s horní částí

III. vykreslit trojúhelníky pomocí **glDrawElements()**

Kam doplňovat kód?

- část I – naplnění polí
  - pole ufoVertices a ufoIndices → uloženy v souboru data.h



- část II – vytvoření vao, vbo, ebo a napojení na shadery
  - funkce initUfoGeometry() → soubor render_stuff.cpp
- část III – vykreslení
  - funkce drawUfo() → soubor render_stuff.cpp

soubor **README.txt**
- seznam čísel řádků kam dopisovat řešení úloh
- blok pro doplnění kódu označen dvojicí komentářů (X označuje číslo úlohy)

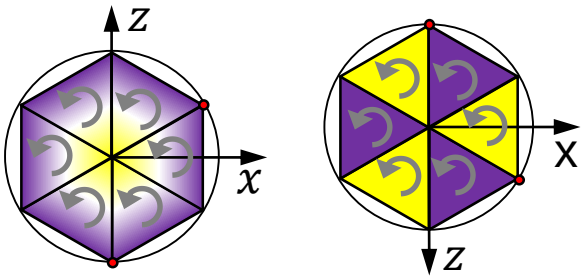
```
// ======== BEGIN OF SOLUTION - TASK 1_X-Y ======== //
// ========  END OF SOLUTION - TASK 1_X-Y  ========  //
```

Užitečné rady

- pro definici pozic vrcholů lze využít předdefinované konstanty ufoH, cos30d, sin30d (*deklarované v souboru data.h před polem ufoVertices*)
- začněte nejprve definicí pouze jednoho trojúhelníku, poté vytvořte vao a vbo, doplňte kreslící funkci a teprve pokud vám bude vše fungovat, jak má (*tj. vykreslí se první trojúhelník*), tak pokračujte doplňováním zbývajících trojúhelníků
- statická kamera kouká na scénu ve směru osy $-z$ → pro kreslení spodní části ufa je nutné zakomentovat vykreslovací příkaz pro horní část (*jinak spodní část nebude vidět*)
- inspirace pro vyplnění pole ufoVertices → pole missileVertices definující geometrii střely (jehlan), ale pozor, pole není prokládané (*v první polovině obsahuje pozice všech vrcholů a v druhé polovině všechny barvy*)
- stiskem klávesy “r” dojde k resetu hry → náhodnému vygenerování pozic objektů

Užitečné rady

- bývá zvykem, že všechny trojúhelníky při pohledu zvenčí mají vrcholy uspořádány v protisměru hodinových ručiček, tj. jsou natočeny ke kameře přední stranou → důležitá je konzistence orientací



- inspirace pro vytvoření vao, vbo a ebo → funkce initMissileGeometry()
- *pokud se nezobrazují žádné objekty (asteroidy, raketa) a máte grafickou kartu AMD/ATI:*
  - *problém způsobuje příkaz glVertexAttrib3f(shader.colorLocation, …); ve funkci loadSingleMesh() v souboru render_stuff.cpp*
  - *řešením je napevno ve vertex shaderu colorVertexShaderSrc (soubor data.h) nastavit barvu: theColor = vec4(1.0, 1.0, 1.0, 1.0);*
  - *pro správné vykreslování ufo je ale potřeba vrátit původní řádek theColor = vec4(color, 1.0);*