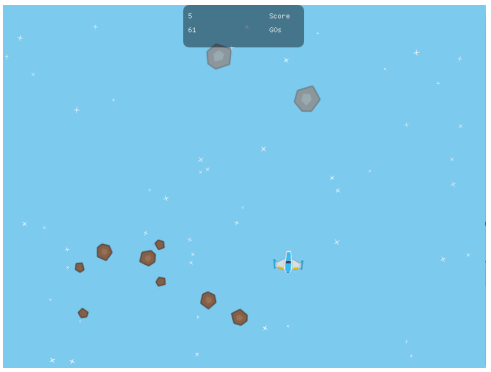
Game Engines: Exercise 4: Sprites

First mandatory exercise!

<table>
<tr>
<td>Learning objectives
  </td>
<td>

Create a sprite atlas
Using translate, rotate, scale of a sprite
Use the pivot point for rotations
Use GLM and vector math


The game uses the following controls:

Arrow keys: Control ship
Space: Shoot laser / restart game
d: Debug circle colliders

Handing in: Create a zip-version of source-files, header-files and resources (CMakeFileLists.txt, .json and png). We will build your project using CMake, so make sure it works before hand-in (Note: If you make sure to keep all files in the same directory it should work without you needing to change the CMakeFileLists.txt).
Do not submit SimpleRenderEngineProject files.
Changes to SimpleRenderEngineProject are not allowed.
</td>
</tr>
<tr>
<td>4-1
  </td>
<td>Create a sprite sheet using www.codeandweb.com/texturepacker (Free) using the sprites found in the Sprites directory. You must include sprites for SpaceShip, Asteroids/Meteors (3 sizes), Laser, and a bang.

You must use the settings described in include/sre/SpriteAtlas.hpp:
https://github.com/mortennobel/SimpleRenderEngine/blob/master/include/sre/SpriteAtlas.hpp

You must create a single sprite sheet (asteroids.png + asteroids.json) which contains all sprites used in the game. This should replace the current sprite sheet.

After updating the file, run CMake again.</td>
</tr>
<tr>
<td>4-2
  </td>
<td>Asteroid class

Must have a constant rotation and a constant velocity. The rotation and velocity should have a random value upon creation.
Asteroids should exist in 3 sizes
When the game starts then add 5 large asteroids on a random position.

</td>
</tr>
</table>

4-3 	**Laser class** • Should move slow (to be visible) • Should fire when space is pressed in the direction that the spaceship is facing from the position of the spaceship. • Laser should be destroyed after 1 second
4-4 	**Collision detection** • Make sure the Asteroids and Laser class inherits from the Collidable class and have an appropriate radius. (Press ‘d’ in-game to see size). • In the beginning of every frame, all Collidable objects should check for collisions with other Collidable objects. • Implement the following behavior: o If a laser hit anything it should be destroyed o If an asteroid is hit by a laser it should be replaced by two smaller asteroids. If the asteroid is the smallest size, then it should disappear. Whenever the asteroid is hit, the score should be incremented by one. o If an asteroid hits a spaceship the player dies and the Bang sign should be shown instead of the spaceship. The game should now restart when space-key is pressed.
4-5 	**Prevent memory leaks** • Make sure that the game does not leak memory • When the player has died, then make sure that space-key will trigger the game to restart. • Make sure the number of GameObjects shown in the GUI (GOs) corresponds to the number of game objects rendered.
4-6 	**Optional: Add final touches** • Add stars as background to the game using a star sprite. • Add win-state • Improve rendering when wrapping (such that you can see the same object clipped both in top/bottom or left/right) • Add multiplayer (control other ship using WASD) • Add enemies