









## Game Engines: Exercise 4: Sprites

<p>Learning objectives</p> 	<ul style="list-style-type: none"> <li>• Create a sprite atlas</li> <li>• Using translate, rotate, scale of a sprite</li> <li>• Use the pivot point for rotations</li> <li>• Use GLM and vector math</li> </ul> <p>An online version of the final game can be found here:  <a href="http://www.itu.dk/~mnob/asteroids/Asteroids.html">http://www.itu.dk/~mnob/asteroids/Asteroids.html</a></p> <p>The game uses the following controls:</p> <ul style="list-style-type: none"> <li>• Arrow keys: Control ship</li> <li>• Space: Shoot laser / restart game</li> <li>• d: Debug circle colliders</li> </ul> <p><b>Handing in:</b> 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 need to change the CMakeFileLists.txt). Do not submit SimpleRenderEngineProject files. Changes to SimpleRenderEngineProject are not allowed.</p> 
<p>4-1</p> 	<p>Create sprite sheet using <a href="http://www.codeandweb.com/texturepacker">www.codeandweb.com/texturepacker</a> (Free) using the sprites found in the Sprites directory. You must include sprites for SpaceShip, Asteroids/Meteors (3 sizes), Laser, and a bang.</p> <p>You must use the settings described in include/sre/SpriteAtlas.hpp:  <a href="https://github.com/mortennobel/SimpleRenderEngine/blob/master/include/sre/SpriteAtlas.hpp">https://github.com/mortennobel/SimpleRenderEngine/blob/master/include/sre/SpriteAtlas.hpp</a></p> <p>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.</p> <p>After updating the file, run CMake again.</p>
<p>4-2</p> 	<p><b>Asteroid class</b></p> <ul style="list-style-type: none"> <li>• Must have a constant rotation and a constant velocity. The rotation and velocity should have a random value upon creation.</li> <li>• Asteroids should exist in 3 sizes</li> <li>• When the game starts then add 5 large asteroids on a random position.</li> </ul>
<p>4-3</p> 	<p><b>Laser class</b></p> <ul style="list-style-type: none"> <li>• Should move slow (to be visible)</li> <li>• Should fire when space is pressed in the direction that the spaceship is facing from the position of the spaceship.</li> <li>• Laser should be destroyed after 1 second</li> </ul>

<p><b>4-4</b></p> 	<p><b>Collision detection</b></p> <ul style="list-style-type: none"> <li>• Make sure the Asteroids and Laser class inherits from the Collidable class and have an appropriate radius. (Press 'd' in-game to see size).</li> <li>• In the beginning of every frame, all Collidable objects should check for collisions with other Collidable objects.</li> <li>• Implement the following behavior: <ul style="list-style-type: none"> <li>○ If a laser hit anything it should be destroyed</li> <li>○ 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 increment by one.</li> <li>○ If an asteroid hit 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.</li> </ul> </li> </ul>
<p><b>4-5</b></p> 	<p><b>Prevent memory leaks</b></p> <ul style="list-style-type: none"> <li>• Make sure that the game does not leak memory</li> <li>• When the player has died, then make sure that space-key will trigger the game to restart.</li> <li>• Make sure the number of GameObjects showed in the GUI (GOs) corresponds to the number of game objects rendered.</li> </ul>
<p><b>4-6</b></p> 	<p><b>Optional: Add final touches</b></p> <ul style="list-style-type: none"> <li>• Add stars as background to the game using a star sprite.</li> <li>• Add win-state</li> <li>• Improve rendering when wrapping (such that you can see the same object clipped both in top/bottom or left/right)</li> <li>• Add multiplayer (control other ship using AWSD)</li> <li>• Add enemies</li> </ul>