

Lab 6: Particles & Steering Behaviours

Objective

- Understand how particle emitters and steering behaviours work

Goals

- Implement a simple particle emitter.
- Implement the seeking steering behaviour.
- Get an understanding of how to use the imgui library to create a simple graphical user interface (gui)

Changes to Framework

- Integrated imgui for UI: <https://github.com/ocornut/imgui>
 - ImGui has a lot functions, the best documentation is the provided sample code which you can find in: include/imgui/imgui_demo.cpp
 - Note: keyboard input in imgui is currently broken, if you figure out how to fix it please let the TA know :)
- Integrated FreeImage for texture loading. FreeImage is wrapped up by the TTK::Texture2D class. <http://freeimage.sourceforge.net>
- Integrated GLEW to get modern GL function pointers. This is needed to render the gui generated by imgui. The rendering functions provided by TTK::Graphics still use GL 1.0
- Added a class to manage key frames. Take a look at KeyframeController.h, you should be able to make sense of everything in this class

You may work individually or in groups of 2 - 3. Only one person needs to submit, make sure everyone's names are in all of the submitted source files

Itinerary

- This week's lab is a little different from the prior labs. Consider this lab a stepping stone for your second assignment. Instead of giving you step by step instructions, this document will give you some ideas and insights into implementing a particle editor. Your task will be to explore and experiment to achieve the following tasks:
 1. Add options in the imgui interface to control the particle life time range
 2. Implement the ability to control the emission rate of particles from the particle emitter and add an option in the imgui interface to control the emission rate.
 3. Implement the seeking steering behaviour and have each particle steer towards a target point (see applyForcesToParticleSystem() in main.cpp)

4. Implement a box emission area and add a UI toggle for box emission or point emission. You should also be able to control the width and height of the box from the UI

Here are some tips:

- To add UI elements look at the ImGui examples in the DisplayCallback function located in main.cpp
- For task (2) emission rate means how quickly the particles spawn. The units for this is arbitrary but a good way to implement it might be to think of it as “particles to spawn per second”. The higher the value, the more frequent particles spawn. Think of a water faucet, the more open it is the more water flows through it.
- For task (3) you need to implement the seeking steering behaviour for the particles, this means each particle should seek towards some target point. Seeking is implemented as follows:
 - $\text{seekVector} = \text{targetPosition} - \text{currentPosition}$
 - $\text{seekDirection} = \text{normalize}(\text{seekVector})$
 - $\text{seekForce} = \text{seekDirection} * \text{seekInfluence}$
 - You then need to pass this seekForce to the particle, see the `applyForceToParticle` function in the ParticleEmitter class
- Task (4) asks you to implement a box emission area. “Point emission” means that all particles spawn from the same position, which is the emitter’s position. To implement “Box emission” you need to create a box around the emitter’s position and randomly select a point in the box to spawn the particle.
 - In other words, implement the “Emit Area” attribute from this example:
<http://particle2dx.com>

Submission

Submit a zip file the following:

1. Any source files you modify

Make sure you put your name and student number in a comment at the top of each file!

Failure to follow these submission guidelines will result in a **zero**!