Explanation

**Implementing Gravity**

http://gamedev.stackexchange.com/questions/15708/how-can-i-implement-gravity

**Collision Detection**

**Collision Response**

1. Book Game Physics

Page 248

2. Chris Hecker's

http://chrishecker.com/Homepage

3. Collision with ground

Geometric Series - Bouncing Ball

https://www.youtube.com/watch?v=ZWEYeT43W4U

Infinite Bounce

http://gamedev.stackexchange.com/questions/49616/why-do-restitution-values-less-than-one-still-cause-infinite-bouncing-in-box2d

http://lolengine.net/blog/2011/12/14/understanding-motion-in-games

**4. Resting Contact**

page 116 - 117

Second we need to recognize when an object has velocity that could only have

arisen from its forces acting for one frame. After frame 1, the velocity of the particle is

caused solely by the force of gravity acting on it for one frame.We can work out what

the velocity would be if only the force acted on it, by simply multiplying the force by

the frame duration. If the actual velocity of the object is less than or equal to this value

(or even slightly above it, if we acknowledge that rounding errors can creep in), we know that the particle was stationary at the previous frame. In this case the contact

is likely to be a resting contact rather than a colliding contact.

**5. Codeblocks compiling stupidity**

- when I switch between "Explosion Generator projection" and the "Tutorial" (Cplusplusguy), due to compilation issues, I can't get it to work:

"Explosion Generator" requires "-std-c++0x" option

but this option won't work with printf, sprintf

To fix this, go to Codeblocks->Compiler-> other Options, and add. That should be compatible to both

-std=gnu++0x

Reference:

https://gcc.gnu.org/bugzilla/show\_bug.cgi?id=40278

**- Using GLSL with SDL**

make sure you define NO\_SDL\_GLEXT before including SDL\SDL\_openGL.h

http://rainwarrior.ca/dragon/sdl\_glsl.html

http://www.evl.uic.edu/arao/cs594/sdlglsl.html

**6. Lighting**

**per-fragment lighting**

1. Specular Light

- Specular Light is the bright spot that occurs when light hits an object surface and reflects back towards the camera.

- Falls of more rapidly across the object surface

**7. Shader**

**Specifying Version Number**

- if a #version directive does not appear at the top, then it assumes 1.10,

**Different Types of Variable (Qualifers)**

- varying: is deprecated

**-** uniform: for per pixel draw call

- in: is for input from the previous pipeline stage,

- out: output to next stage

- attribute: effectively equivalent to an input qualifier in vertex shaders. It cannot be used in any other shader stage. It cannot be used in interface blocks.

- http://stackoverflow.com/questions/6196007/opengl-shading-language-different-types-of-variable-qualifiers

**- http://www.opengl.org/wiki/Type\_Qualifier\_(GLSL)**

**Learning Modern 3D Graphics Programming**

- http://www.arcsynthesis.org/gltut/

**8. OpenGL Transformation and OpenGL Matrices**

- http://www.songho.ca/opengl/gl\_transform.html

- At June 14th, I finally am starting to realize what's going on

- After projection to clip space, the points are no longer normalized, we restored by dividing all components of the vertex by w, so we then it the vertex are in (-1,1,-1,1,-1,1) range

gl\_Normal

- Inverse Transpose of gl\_ModelView matrix

- https://www.cs.uaf.edu/2007/spring/cs481/lecture/01\_23\_matrices.html

- http://www.lighthouse3d.com/tutorials/glsl-tutorial/the-normal-matrix/

Coordinate systems in OpenGL

- http://www.matrix44.net/cms/notes/opengl-3d-graphics/coordinate-systems-in-opengl

**9. Matrix Stacks**

**-** 3 stacks of matrices, one stack for each matrix type (Texture, Projection and ModelView) **-** each stack top is automatically applied to every vertex

- http://vis.cs.pitt.edu/teaching/cs1566/lectures/L09\_ViewingOpenGL.pdf

- Anything You don't understand about tutorial 8, built-in matrices by the CplusplusGuy, just look at this pdf and you'll get it

**Normal Vector and ModelViewMatrix**

- http://www.lighthouse3d.com/tutorials/glsl-tutorial/the-normal-matrix/