## Plans

* Elastic Ground collision on ball
* Non-elastic ground collision on ball
* Introduce elasticity attribute
* Determine if two objects are colliding
* Ball collisions
* Wall collisions
* <http://digitalerr0r.wordpress.com/tutorials/>
* Shadows for point light
* Depth of field
* Bloom
* Transmittance
* Refraction
* Better version control

## 20140103a

* Drawing the camera frustum doesn’t work because you need to draw it as a world object, not from the Camera.Draw function

## 20140107a

* If Depth buffer problems: defined in RenderTarget constructor, is turned off by SpriteBatch, and check if depth buffer is turned on before each draw.
* Button for Shadow target (light perspective), cycle through perspectives
* If Billboarding bug: due to translating to world in vertex shader, created a separate vs

## 20140108a

* Specular color is cool! What if color was based on normals? Could get an oil slick effect
  + Find reflection angle
  + If angle between 0 and 45, scale 255,0,0 to 255,255,0
  + If angle between 45 and 90, scale 255, 255,0 to 0,255,0
  + If angle between 90 and 135, scale 0,255,0 to 0,255,255
  + If angle between 135 and 180, scale 0,255,255 to 0,0,255
* Fog factor, see BasicEffect.fx
* Can I make a world with a final image that has transparency?

## 20140111a

* Regarding half pixels: see <http://drilian.com/2008/11/25/understanding-half-pixel-and-half-texel-offsets/>
  + Note, that when I say “subtract a half-pixel from position,” what I mean is “move the position of the quad one half-pixel towards the upper-left.  To do this, you actually add (-1/width, +1/height).  The reason for the signs (-, +) is that you’re moving left and up.  In post-projection space, -x is left, and +y is up.  The reason that it’s 1/width instead of 0.5/width is because, in clip space, the coordinates range from -1 to 1 (width 2), and not from 0 to 1 (width 1) like they do in textures, so you need to double the movement to account for that.
  + half pixel offset is specific to D3D9 as D3D10/OpenGL/consoles do not need to do this.
  + Of course, if you’re drawing world geometry and using its screen coordinates to index into a texture map (like, for instance, if you’re using [light pre-pass rendering](http://diaryofagraphicsprogrammer.blogspot.com/2008/03/light-pre-pass-renderer.html)), you’ll still want to add the half-texel instead.  In fact, [here is a great reference on how to handle this case](http://diaryofagraphicsprogrammer.blogspot.com/2008/09/calculating-screen-space-texture.html).
* Create a function that when the window is resized, it resizes halfpixel and all of the rendertargets

## 20140112a

* To render non-aliased shadows, perhaps: From the light’s perspective, clip the rendered world to the camera’s perspective and increase the resolution of the rendered world from the light’s perspective, proportional to the render of the world from the camera’s perspective. This should produce shadows that are mapped to the resolution of the final image.
* Create camera in wings 3d
* Create view frustum objects
* Create visual camera objects that hold a view
* <http://stevehazen.wordpress.com/2010/02/15/matrix-basics-how-to-step-away-from-storing-an-orientation-as-3-angles/>

## 20140113a

* Depth must be a single, otherwise it will be clamped and cause weird problems with spotlight
* Quake cam control objects
  + Move cursor when Ctrl is held down
  + Move selected object with wasdqe when Ctrl is held down
  + When left click is held down, move object with cursor, wheel, rotate with wasdqe
* Pointlight shadows
* Directional light shadows
* Geometry light
  + Start with point light that uses sphere.x
* Shaped light
* Real mirror
* Real reflection specular

## 20140115a

* <http://channel9.msdn.com/coding4fun/blog/Adding-awesome-lighting-effects-to-your-XNA-game>
* <http://jcoluna.wordpress.com/2011/07/06/xna-light-pre-pass-cascade-shadow-maps/>
* <http://msdn.microsoft.com/en-us/library/ee416307%28v=vs.85%29.aspx>

## 20140118a

* Include a reflection map, that either reflects the scene or reflects by default a pre-rendered scene, like a well lit busy office. Multiply this with specularColor.
* coneDirection = normalize(facingDirection – position)

## 20140124a

* Association web
  + Create node objects that contain information
  + When I mouse over a node, it displays a tooltip with some of its information
  + Nodes exist as glow balls
  + Nodes can be associated to other nodes, forming a line between them
  + Lines pull the objects toward each other, but repel each other when they are close enough
  + Some nodes can be topic nodes, other nodes can be data nodes
  + Put one reference paper in each node
  + Papers that reference other papers will have lines to them in red
  + Papers that deal with certain topics or keywords will have lines pointing to those topics
  + A paper node can have different visual attributes, such as transparency, size, color, brightness, based on factors like: how good I think it is, how relevant it is to the selected topics, how many papers reference it, how many papers relevant to the selected topics reference it
* Fog
* Depth of field
* Select 3D objects
  + Project a silhouette of the object from the camera’s perspective
  + Enlarge the silhouette by a few pixels and draw the silhouette behind the object
    - The silhouette should be part of the object draw code so that it is always draw immediately before the model with depthstate off
  + Increase the Material.ambient of the selected object while it is selected
  + Move the object relative to mouse movement, relative to the camera, while respecting world collision detection
* Implement LUA library for UI development
* True specular mirror shader

## 20140125a

* Better asset management and state management
* Save
* IsFixedTimeStep = false will cause physics to desync

## 20140126a

* Instead of using states, I could use events and dynamic Action<> methods
* Turbo causes physics to desync

## 20140128a

* Textured Balls
* Scale camera by simultaneously scaling the viewport and fieldOfView

## 20140205a

* Bring back the smooth lerpy camera
* Replays can be done by recording the input in every frame to file, then read back the file

## 20140206b

* Fix Camera with DirectedObject
* Fix Lights with DirectedObject, particularly spotlights

## 20140207a

* To achieve perfect ball to surface collisions:
  + If the ball has moved past the boundry in this frame, Find the point that intersects with the path taken by the ball, then find the distance from that point to the ball, then reflect this vector 90° toward the wall pivoted on the intersection point. Multiply this reflection vector with the magnitude of the position and velocity vectors.
  + Much simpler than calculating the time of collision and running the collision formula
  + Include a threshold of energy absorption that modifies the returning reflection vector from 1.0f (full elastic collision) to 0.0f (no elasticity, no bounce)
* Have a choice between a number of collisions increasing in complexity
  + Dumb collisions: move object to edge and reflect
  + Good collisions: reflect position and velocity on point of intersection where path and edge meet

## 20140208a

* Draw camera in overlay with transparency
* The Mind
  + Run mind processes in the game using states, such that tasks can only have a certain amount of complexity per sub-frame, and there is a limit to how many sub-frames can happen in a game frame.
  + Allow the user to control the number of sub-frames.
* For pan, transition, I need absolute and relative versions
* For Place, redesign for absolute and relative
* Draw camera model
* Frustums

## 20140215a

* Implement an association web using the Netsky Engine; Refer to the notes in Plans.docx
* For a particular paper I am looking through, I may want to note properties about the paper.
  + Domain: Aerospace
  + Topics: Ad-hoc mobile networks; IDS;
  + Dataset Attributes
    - Domain: Aerospace
    - Private
    - Size
    - Range
    - Design
    - Attack type
    - Audit material
    - Detection technique
    - Size
    - Features
    - Records
    - Design (Simulated//Anonymized/honeypot)
* A paper shall exist in the world as a node
* Nodes are represented as spheres
* When a node is selected, the associations are drawn, represented as arrows to other nodes
* Nodes can have different types
  + Papers
    - Papers will be scaled to represent their contribution, which is determined based on how many associations it has.
  + Paper topics (whose parents are the paper and orbit the paper node)
    - Paper topics will be scaled to represent their portion of the paper
  + Static Topics
    - Static topics are general topics not associated with a specific paper.
    - Paper topics will associate to Static topics
    - Static topic will be scaled to represent their importance, which is determined based on how many associations it has.
  + Datasets
    - These shall be independent of papers, but papers will associate with them.
  + Keywords
* When a node is selected, the camera shall focus the node. Unrelated nodes will be pushed away. Related nodes will be pulled in close.
* Nodes naturally repel each other with a force based on mass and distance.
  + Nodes all have the same density, scale mass with size.
* Selecting a nodes causes an attractive force on associated nodes and a stronger and wider repulsion force on unassociated nodes.
* Selecting a node pulls up relevant information on the node.
* Need a simple graphics option for shitty systems
  + Only a very basic one-pass render with one light.
* Need a GUI
  + Check out <http://neoforce.codeplex.com/>
  + <http://www.digitalrune.com/>
* Better Skybox with dynamic elements
* Increase visible depth to infinity
* Don’t render objects that have a radius smaller than what would render as a few pixels; Render according to a function of depth and radius

## 20140616a

* Perfect shadows – FFXIV does them with character models… It appears that the detail of the shadows are a function of the camera position
  + Shadow volumes? What is this?
  + Current implementation: Render the scene from the perspective of the light to a shadow map
* Need to implement a new game mode that runs replays, where there is no physics or regular game interaction, similar to pause, but in each frame, a line of a replay script is read in which object positions and behaviors are dictated.
  + To record a replay, each object shall have its state adequately captured for each update. These replay frames shall have a datetime associated with the capture. These captures happen on updates, not on draws, which means there are magnitudes more captures than draw frames.
  + It should be possible to rewind and fast forward
  + It should be possible to change the mode of the camera to freely move during a replay while the replay is in any state (rw, play, ff, pause), whereas the default mode of the camera shall be the recorded behavior of the camera
* Need to implement a process management architecture that allows for processing of tasks in fixed frames of time and these processes can be paused and continued between frames. An implementation of this is needed for an implementation of a cognitive architecture that is synced to a rate of time of the game world. The game world rate of time does not need to be in sync with real world time.
  + Limiting the processing of a task to a certain amount of time can be done by using threading and sending an interrupt signal to the thread when time expires. The process will have to check for this interrupt signal to exit the process. When the process is interrupted, it needs to save progress for possible later continuation.
  + For example, a goal can split itself into subgoals then do each of those goals. The goal can check for interrupts between each subgoal. Interrupts are cascaded into subgoals.

## 20150101a

* The Mind
  + The mind will NOT run asynchronously. It must be bound to the same flow of time present within the game world.
    - Need to keep track of number of ticks that have passed.
    - Consistently check how much time was spent processing.
    - Force update completion after number of ticks even if the mind is not ready.
    - A process taking place in the mind may not be complete in time for a forced update, therefore it needs to be handled by deciding on how to retain progress of this process for the next update.
  + I will need a graphical display of what is going on.
    - What processes are running and their statistics.
    - Where the processes lie in hierarchical order.
  + Need performance visualization for functions in my code.
    - How many ticks each function takes
    - How much memory is allocated each cycle

## 20150105a

* The Mind
  + The cognitive architecture shall be designed solely using the unique language of processes so that the architecture itself can be modifiable by the mind.
  + A process is literally an allocation of resources. Whether it be computational space, time, or energy.
    - To maintain the contract for the given resources, the process must follow a resource-management pattern wherein it is always possible to determine the state of the resource allocation. Possible tactics may be the following:
      * Check periodically on the remaining resources. If the resources are spent, the process ends.
      * As a fail-safe, if a process exceeds its allocated resources by a percentage, the process will be considered unsafe and discarded