18/6/2018

* Started to work on vectors. 3 Dimensional to work between particles to decide if they would interact between each other.
* Need to start a random number generator that has random and programmable varieties.

19/6/18

* Random Number generator Class with a true random number generator and one that uses a seed number. They both only produce random numbers between 0 and 360
* The model cannot know about the implementation and therefore for the random number generators cannot be hard coded in and must be changed in the console for testing.
* The RNG return an array of the numbers [2] which is the angle between x and y and the angle between this plane and the z plane.
* Using Interfaces allows the user to decided on the random number generator required on runtime.
* Need to now work on how the ejection vector and the movement vector relate.

24/7/18

* Test method set up to test the movement and ejection
* Only one test included so far to test the programmable random number generator and it has passed
* Created a collision Class which has an annihilation collision in (Particle + anti-particle = 2 photons)
* The collision uses to methods to determent the total energy of the two particles using rest mass and velocity. This property would probably be better for the particle themselves to calculate and know themselves to avoid complication
* Need to implement a particle tree as particles are being created when they are needed in the vectors class and this is completely unworkable
* Photon class needs to be located in a suitable area
* Created a plan for the development of the project in layers (Like a cake) so parts of the program do not depend on other parts that they should not.

26/7/18

* Started the particle layer for the main project
* Need to find the actual values of the rest masses for the hadrons
* Created the Quarks but only the Up and Down flavours have hold any data
* Meson class set up but I need a list of all the mesons and their data to be able to add them as there are a lot
* Implemented a test to check if the particles are completed and the test will fail if they are not. This is a checklist for me to work through when I can find out the specifics of each particle

1/8/18

- Added a spin property to all particles, it is currently only a Boolean data type but needs to be changed to Left and Right somehow (New class?)

Plan for final development structure

Layer 1: Particle

* Where the particle diagram is used and that relation is coded into the system. All particles are also provided with a full array propitiates and methods they may require (eg Mass to Energy)
* This is the first layer to be implemented as it will allow the particles to be used throughout the development of the rest of the program with ease
* Uses separate program to the collision, vectors and UI programs
* Start with the Quarks and then move onto the bigger particles

Layer 2: Collision

* Where all the calculations for the collisions takes place and outputs what particles are created
* Will need different collisions for different types (eg Electron capture, annihilation, Proton-proton)
* Should not require any of the vectors layer in coding

Layer 3: Vector

* Where the vector calculations and positioning takes place
* Calls upon both the particles and collisions layer to determine weather any new collisions take place after the primary collision

Layer 4: UI

* The terminal screen UI should be developed alongside the rest of the program but should not be integrated into it as it will be removed for a cleaner graphical UI later.
* The graphical UI will be the last part of the program to be developed.