QUADOTS
Design Document

Souren Papazian (ssp2155)
Surashree Kulkarni (ssk2197)
Srilakshmi Chintala (sc3772)

Error handling resource management

Feature set:

1. The Quadots library provides Point and Dot objects and helper functions to manipulate the behaviour of these objects in 2D space. The library makes use of the Quadtree data structure on which the helper functions are built.

The user can define the behaviour of these objects to produce interesting simulations like:

- Boid's implementation(or other physical simulations)
- Collision detection in two dimensions
- Spatial Indexing
- Conway's Game of Life simulation program
- Image analysis
- State Estimation

A user on the more granular level can make use of the library to :

- Create new point or dot objects
- Update the state/position of objects
- Define the behaviour(rules) of these objects
- Run simulations for n objects
- Generate n random objects and simulate behaviour

2.The Quadot library inherently makes use of a Quadtree structure on which the helper functions are built. This gives the advantage over just using a brute force method to access the point objects.
Quadtree is a tree data structure in which each internal node has exactly four children. Quadtrees are most often used to partition a two-dimensional space by recursively subdividing it into four quadrants or regions. The regions may be square or rectangular, or may have arbitrary shapes.
Essential features of a Quadtree are:

- They decompose space into adaptable cells.
- Each cell (or bucket) has a maximum capacity. When maximum capacity is reached, the bucket splits.

The Quadtree structure provides functionality for:

- Inserting objects into the quadtree
- Deleting objects from the tree
- Getting the nearest neighbours of an object (logic for collision detection)

Interface design:

Optimizations

Features for the convenience of users

Ideas for release 1.2

The Quadots library version 1.0 lets the user perform only basic manipulations on a Point/Dot's coordinates, velocity and angle. Version 1.2 should have the following features:

- Added features for Dot/Point.
- Integration with the C++ Graphics so a user has added flexibility to use any kind of shapes he wants as elements, and is not restricted to simply Points/Dots.
- Concurrency- manipulate clusters of points parallelly to improve execution speed.
-