Selectable:

On start, adds itself to the selection controller selectable master list.

public bool selected; tracks if object is selected

Selection Controller:

// desc

public bool onEnemy; tracks if cursor is over enemy

List of all selectables and list of currently selected selectables

public List<Selectable> masterSelect = new List<Selectable>();

public List<Selectable> currentSelect = new List<Selectable>();

These vectors keep track of the box drawn for selection

[SerializeField] private GameObject selectionAreaTransform;

These vectors keep track of the box drawn for selection

private Vector3 startPosition;

Vector3 lowerLeft;

Vector3 upperRight;

Vector3 lowerRight;

Vector3 upperLeft;

Update( )

If game is not paused, checks for input to select or command

Mouse to world and get mouseworldposition3d are helper functions

Entity:

Class controls boids movement. Alignment moves towards target, cohesion is supposed to keep speed and direction with other boids (hard to tell if its working currently), and separation keeps them from clipping into each other. Bounds keeps them within a box, perching stops them

As of now, Entitys calculate their own path, but report to a EntityManager class in their parent object. This allows for the center of the entites to be neatly calculated, and allows the entitys to have a list of their sibling entities, and allows for settings to apply to a whole group of entities

Entity Manager:

Calculates the center (avg pos), and rotates in the direction of avg motion. Settings here will apply to every child entity

Entity Controller:  
need to see where code from the manager and the controller differ.

Game Controller:

Static main designed to connect other controllers, a hub

Input Controller:

Route all input checking through here.

RTS Camera:

RTS view, moves forward in direction of z axis of camera