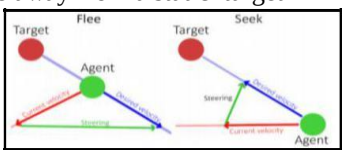
Final Project Title: Steering Behaviors for Autonomous characters:

1) **Introduction**

In this Project, steering behaviors for autonomous characters are implemented. Different combinations of steering behaviors are used to achieve higher level goals such as avoiding obstacles while characters are in motion and making the characters follow the leader. We implemented various steering behaviors described below:

i. **Seek:** steers towards a static target

ii. **Flee:** steers away from a static target



iii. **Pursuit:** Pursuit is similar to seek except that the target is in motion.

iv. **Evasion:** In Evasion, the target moves away from a moving pursuer.

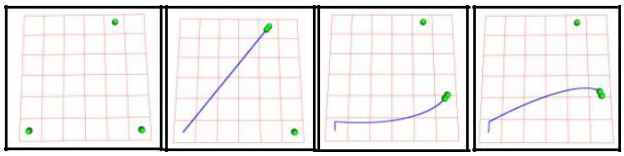
v. **Collision** detection and avoidance: to avoid collision between characters and other.

2) **Design and Methodology:**

Based on above-mentioned steering behaviors, we used 3 different scenarios for simulation:

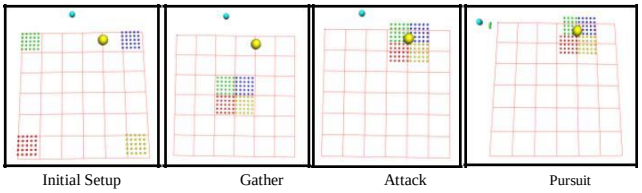
i. **Scenario-1:**

The intended goal is to implement basic seek, flee, pursuit and evade for one character and single target. Two algorithms for pursuit are implements. See below figure for reference.



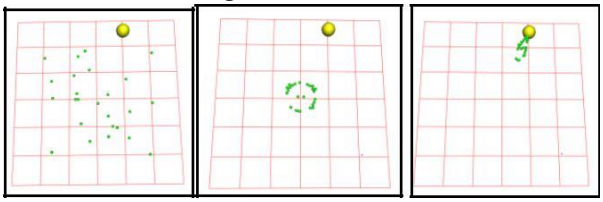
ii. **Scenario-2:**

In the second scenario, a scene to simulate an army following a leader to attack a kingdom is implemented using the steering behaviors. They assemble at a point and attack. When the kingdom is attacked the king evades and the army's top five agents pursue to capture.



iii. **Scenario-3:**

In the third scenario, troops assemble at a point from random positions with random velocities. They have to form a circle at the assembly point. Collision detection and avoidance enabled a circle to be formed. The troops attack the kingdom, forming a line to attack. Figure shows this scene.



3) **Implementation challenges:**

The challenges that we faced were mostly collision detection and avoidance between characters. The algorithm used was effective in detecting obstacles/ characters towards the character's velocity.

4) **Performance Evaluation:**

Scenario 1:

i. Two pursuit algorithms were evaluated; it was found that the algorithm that approaches the evader from its predicted future position caught the evader in an earlier time.

Scenario 2:

i. Different velocities were used to pursue the king. As expected the higher the velocity of the pursuer the king is captured. However, when the king's velocity is greater than agent's velocity, the king is not captured.

Scenario 3:

i. Adjusting the number of character: As the number of characters is increased, it becomes difficult to handle collision detection and form a perfect circle at the center. Some characters remain behind other particles as shown in . As the number of character go above 30.

ii. With a higher velocity the line formed to attack the kingdom is also not perfect. Collision detection is not perfect..

iii. When the agents are moving with different velocities, it is difficult to detect collision between them as some of them will come closer and some will go far apart (Unlike in Scenario-2).

5) **Conclusion:** In this project, various new topics were studied and evaluated. Different functions used by autonomous characters have been implemented such as seek, flee, pursue, evade, collision detection and avoidance. The algorithm used for collision avoidance can be improved so pursuing and attacking can be effectively done, especially when velocities are random or high.