1. Implementation

When the agent is build, we use “state” to clarify what scenario we choose.

In SocialForcesAgent::updateAI(), pursueEvade(dt) and leaderFollow(dt) are used to calculate the social force.

1. Pursue and Evade:

In Pursue and Evade, red agents are pursue agents.

Util::Vector SocialForcesAgent::pursueEvade(float dt) calculates result for pursue type agents and evade type agents.

The pursue agent needs a force to pull it to the position, so we use result += agent->position() + agent->velocity() \* dt - position(), so that agent has a force with a direction to the position.

The evade agent needs to go away from pursue agent, so we use result -= agent->position() + agent->velocity() \* dt - position().

1. Leader Following:

In Leader Following, the leader acts the same as the default agents. The followers are almost the same as pursue agents, but a separation is added so that followers won’t crash into the leader.