Synopsis

I will provide to jME **a clean implementation of an A.I.** in order **to control the locomotion** of “characters”. Implementing this A.I will can reduce drastically the time needed to develop a game, especially a rpg or racing game, since for almost every game we need to set how these “characters”, that can be a group of monsters or a car, will be moving around the scene.

I'm a passionate of video-games, 3d art and programming since I was a little boy and **I really enjoy spend all the day coding**. I'm specially motivated to this project because I would love to see the beauty of the finished algorithm, producing a 3d boids simulation.

Deliverables

When this project is finished users will be able to **provide to a vehicle** (We will call vehicle to any model that uses locomotion. It can be a character, a spaceship, etc.) the ability to:

1. **Seek** a point / Arrive to a point:
2. **Flee** away from a point or several points
3. **Pursuit** another vehicle
4. **Evade** another vehicle or a group of vehicles
5. **Avoid** obstacles
6. **Mix** several of the abilities mentioned before

Project Details

Every vehicle should be able to do a list of **basics behaviors** by separately.

It will be possible to **merge** several of the **basic behaviors** and obtain almost any wanted behavior. We have to **take in mind since the first moment that the basic behaviors should be compatible** in order to merge them without bugged stuff.

An in depth explanation how these behaviors work , with included example apps, can be found in these websites:

- http://www.red3d.com/cwr/steer/
- http://gamedevelopment.tutsplus.com/series/understanding-steering-behaviors--gamedev-12732

An illustrative example of merging these basic behaviors can be found in this video:

- https://www.youtube.com/watch?v=86iQiV3-3IA

Finally, each behavior should include an amount of **configurable settings** in order to provide the user a way to adjust these vehicles to his own project.

We don’t have to reinvent the wheel because there is a massive amount of **information and code** about these behaviors available on the **internet**. So I will be constantly reviewing already coded algorithms and porting it to our java project.

In particular it is needed to **check the library already developed** that can be found here:

- https://code.google.com/p/jmonkeyplatform-contributions/source/browse/#svn/trunk/jme3-artificial-intelligence/release/libs

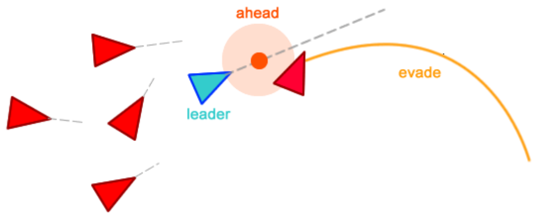
One of the sources that I will be reviewing the OpenSteer c++ library (http://opensteer.sourceforge.net/) too, and seeing what improvements we can include (making the needed adaptations) in the project. T**his java library** attach the agents with **JME Control class.**

It would be **possible** to make **native bindings with the c++** library, **but** because of the simplicity of the basic steer behavior **transferring the code to Java worth it**.

Groundwork:

These weeks I have been reading and seeing examples of jME. Then I started checking the java steer behaviors library:

First of all you can see that t**he pursuit function do not work well,** some **pursuers stay in front of the pursued.** Instead, they should evade this situation like is shown on the picture.



Furthermore, **the "avoid" behavior is really bugged**, the vehicles are trying to change the trajectory but "other forces" impede it. This should be further investigated in detail.

In this video you can see the results *after fixing* some problems: http://youtu.be/8ZboS5Mc8m8

Project Schedule

The project will require **13 weeks** to be completed and will need **6 hours of work per week**. Additionally, I have all the summer free so I can spend more time if needed.

1 weeks: May 19th – May 25th :

Gather information, what solutions can be implemented in our project **and how**.

4 weeks: May 26th – June 22th :

Check the java library **updating and fixing "seek", "flee", "pursuit" and "evade"** behaviors.

3 weeks: June 23th – July 13th :

Implement the **"avoid"** (including separation) behavior.

4 weeks: July 14th – August 10st :

Implement the possibility of **mix several behaviors**. Identify **bugs and fix them**.

1 week: August 11nd – August 17th :

Make **tutorials** and **demos** showing the power of the features. **Review** all the code, documentation, tests and examples.