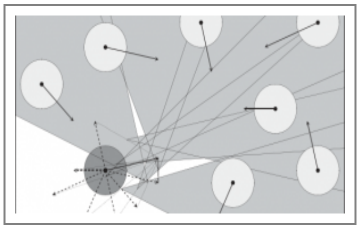
OPEN TEASER



Velocity Obstacle Algorithms and Locomotion Integration

Alex J. Champandard on April 30, 2012

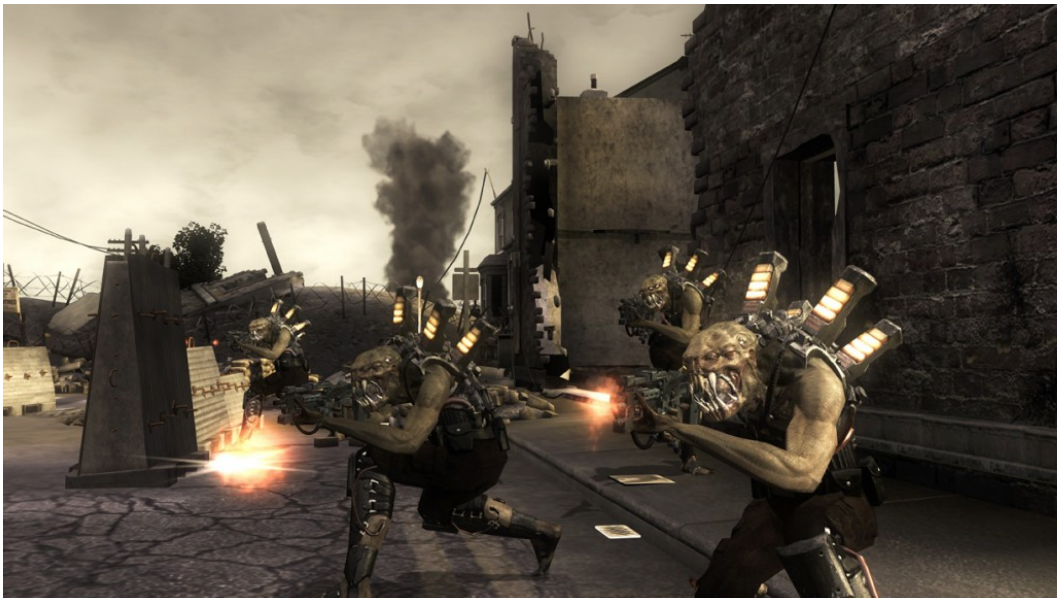
Share This!

13 Like 5 Tweet 4 +1

submit reddit

Over the past 12 to 18 months, collision avoidance algorithms based on velocity obstacles such as ORCA and RVO have seen a dramatic increase in adoption — in particular thanks to titles like CRYSIS 2, SPACE MARINE, and RESISTANCE 3 leading the way. This move has increased the quality of in-game characters, but many questions still remain open...

Yesterday, a live masterclass broadcast via AiGameDev.com with one of the field's leading researcher, Jamie Snape, digged into the topic further. Jamie has been heavily involved with the development of the RVO2 libraries and the HRVO and ORCA algorithms.



Screenshot 1: RESISTANCE 3 uses an RVO-style implementation for collision avoidance. The enemies navigate around each other, but rely on physics to deal with collision with walls and navigation mesh boundaries. (See below.)

Locomotion Integration

One of the big challenges of such algorithms is the integration with existing locomotion and animation systems.

> *"The question here is the significance of the collision avoidance vs. everything else. Are you going to let your algorithm have the last*

> *word? Some developers take the output of ORCA as a prefered guidance velocity for their animation system, whereas other developers try to follow the algorithm closely and use the actual velocity."*

Snape points out that the decision depends on the density and complexity of the scene. If you want characters to avoid collisions in such instances, you'll need to follow the output of the algorithm very closely. Such questions are the kinds of issues that AI programmers have been facing for decades now, even while using simpler algorithms such as steering behaviors for instance.

Snape also draws from his experience consulting on the recent SPACE MARINE game developed by Relic, which was an iterative process.

> *"That's the impression I got. [...] It's something that takes time to discover, from working with the algorithm for about a year. It's always a compromise, for example SPACE MARINE had about 50-100 characters, and sometimes they're close together and at other times they are not. In general, it requires an adaptive approach to consider the collision avoidance more or less depending on the situation."*

Screenshot 2: SPACE MARINE uses an RVO library to provide a fast implementation of collision avoidance for its enemies, which can end up in large dense clusters during parts of the game.

Navigation Boundaries

Snape points out from his experience on SPACE MARINE that there are further challenges with the navigation integration too:

> *"One of the things we found out from this example is the relative cost of dealing with static obstacles vs. moving agents. Maybe it's a little counter intuitive, but dealing with a static line obstacle is harder than dealing with a dynamic obstacle! You can simulate 10 agents for a single static obstacle, how you chose to deal with that is very important."*

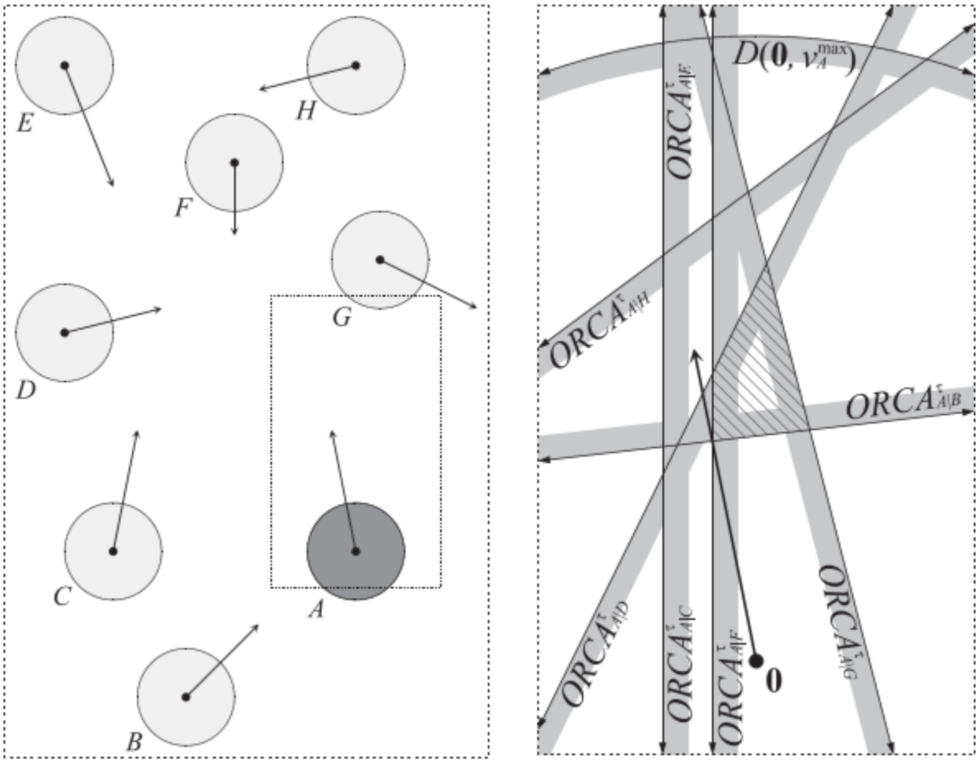
Snape continues by explaining some common solutions to these issues.

> *"If you can avoid using these algorithms to deal with static obstacles, then that's probably the best thing to do because they really were designed for moving obstacles and that's the thing we really worked*

> *on. [...] In the RVO2 library in particular, there's four times more code for static obstacles!”*

RESISTANCE 3 in particular does not integrate navigation boundaries with its RVO implementation, and instead relies of the collision and physics simulation to push the agents away from static obstacles.

Snape concludes by pointing out there's a large scope for research there, in particular unifying the two into a very efficient algorithm!



(a) (b)

Figure 3: CRYSIS 2 in particular uses the ORCA approach to collision avoidance, not least for performance reasons.

Further Reading

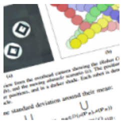
If you'd like to dig into futher details about RVO-style technology and its application, here are good places to start:

1. Watch the replay and slides of the masterclass with Jamie, available with PLUS subscriptions and above.
2. See the interview with Jesse Cluff about the AI in SPACE MARINE, including the RVO part. (PREMIUM)
3. Watch the interview about RESISTANCE 3, also inculding a section about collision avoidance. (PREMIUM)
4. Grab your copy of the Paris Shooter Symposium 2011, and find out how ORCA was applied to CRYSIS 2.

You can access the developer interviews and other masterclasses with a PREMIUM or ULTIMATE membership. The Shooter Symposium recordings are sold standalone or with studio memberships of PLATINUM and above; see the bottom of this post for details.

Content Index

Related Articles


Collision Avoidance for Game Characters with Velocity Obstacles

Problem Representation for Planning Algorithms 101

Are Evolutionary Algorithms Suitable for Building AI in Games?