

SE101 Ideas Clinic Spaceship Activity

Collaborate & Converge:

- The simulation should always be able to run. Maybe sometimes the results won't be ideal.
- Use feature flags. Don't delete past iterations of a subsystem: use a feature flag to control which iteration gets used at runtime.

Iteration	Sensors	Navigation	Propulsion	Defence
0	Determine Interfaces	Determine Interfaces	Determine Interfaces	Determine Interfaces
<i>Rotate Pairs/Squads</i>				
1	Report fake objects	Go to first visible warpgate	Use UFO drive to reach desired destination	Demonstrate firing, aiming, burst fire, and controlled/sequential fire in multiple directions
<i>Rotate Pairs/Squads</i>				
2	Filter and clean data, pass to peer subsystems	Hard-code Wikipedia solution to Dijkstra graph	Implement PD control for directed thrusters	Repeatedly destroy nearest obstacle as fast as possible
<i>Rotate Pairs/Squads</i>				
3		Implement Dijkstra's algorithm	Minimize fuel costs (i.e. parking on top of warp gates before triggering jump)	Intelligently destroy obstacles while conserving ammo (i.e. fire only as often as necessary, avoid needless destruction)

Why evolutionary + horizontal prototyping? <ul style="list-style-type: none"> - integration is a key challenge - algorithms are known - not safety-critical - optimal performance requires experimentation 	Learning Objectives <i>Multi-Person Development</i> <ul style="list-style-type: none"> - Pair programming - SE Team Organization 	Maximize Growth Opportunities <ul style="list-style-type: none"> - Go outside your comfort zone - Work on a weakness - Go beyond current strengths 	No Marks This is a large, open-ended, team-based activity. It is too hard to assess marks. All your other schoolwork is typically small, closed-form, individual exercises -- which are easy to assess. You will get out of this activity what you put in to it.
	<i>Multi-Version Software</i> <ul style="list-style-type: none"> - Version control (Git) - Feature Flags - Prototyping Theory 		