International Journal of Space Technology Management and Innovation

July-December 2012, Vol. 2, No. 2

Table of Contents

The *International Journal of Space Technology Management and Innovation* is currently listed or indexed in: Bacon's Media Directory; Cabell's Directories; INSPEC; JournalTOCs; MediaFinder; ProQuest Advanced Technologies & Aerospace Journals; ProQuest Illustrata: Technology; ProQuest SciTech Journals; ProQuest Technology Journals; The Standard Periodical Directory; Ulrich's Periodicals Directory

Collaborative and Educational Crowdsourcing of Spaceflight Software using SPHERES Zero Robotics

Sreeja Nag, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA

Jeffrey A. Hoffman, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA

Olivier L. de Weck, Department of Aeronautics and Astronautics and Engineering Systems Division, Massachusetts Institute of Technology, Cambridge, MA, USA

ABSTRACT

Crowdsourcing is being researched as a problem-solving technique by issuing open calls for solutions to large crowds of people with the incentive of prizes. This paper tackles the dual objectives of building cluster flight software and educating students using collaborative competition, both in virtual simulation environments and on real hardware in space. The concept is demonstrated using the SPHERES Zero Robotics Program, a robotics programming competition where the robots are nano-satellites called SPHERES onboard the International Space Station (ISS), traditionally used as a Guidance, Navigation and Control testbed in microgravity. Zero Robotics allows students to program SPHERES to play a game through a web-based interface and the most robust projects are evaluated on the ISS hardware, supervised by astronauts. The apparatus to investigate the influence of collaboration was developed by (1) building new web infrastructure where intensive inter-participant collaboration is possible, (2) designing a game that incentivizes collaboration with opponents, to solve a relevant formation flight problem and (3) structuring a tournament such that inter-team collaboration is mandated. The web infrastructure was also built using collaborative competitions, to demonstrate feasibility of building space software end-to-end by crowdsourcing.

Keywords: *Crowdsourcing, International Space Station (ISS), Spaceflight Software, SPHERES, Zero Robotics*

DOI: 10.4018/ijstmi.2012070101

1. INTRODUCTION

Crowdsourcing, in the context of this paper, is defined as the methodology by which a well-defined problem is attempted to be solved by announcing it as an open call for solutions to crowds of people with the incentive that the best solutions will be awarded prizes (Howe, 2006). There is no restriction on the methods that the crowds can use to solve the problem, but there may be a time limit given to come up with a solution and constraints on the ways in which the proposed solutions are submitted. Historical applications include John Harrison's longitude determination method and Leblanc's production of soda ash from salt. Recent applications include the Climate Co-Lab to address climate change (Laubacher, Olson, & Malone, 2011), iGEM to engineer biological organisms (Goodman, 2008), Clickworkers (Ishikawa, Gulick, 2012), NASA Tournament Labs – NTL (Boudreau, Lakhani, 2010) and the Mars Crowdsourcing Experiment to annotate semantically rich features of Mars (van 't Woud, 2011).

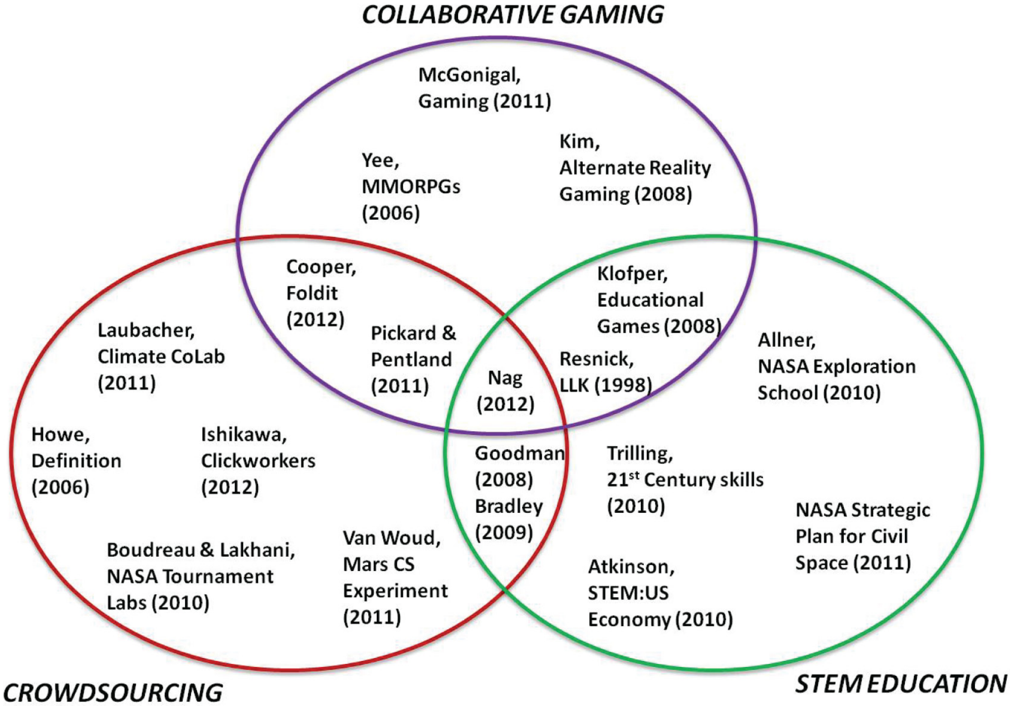
CS-STEM is an acronym for Computer Science (CS), Science, Technology, Engineering and Mathematics. CS-STEM Education refers to efforts invested in bringing students and young professionals, the next generation workforce, up to speed in the fields of CS-STEM and therefore be prepared to address the grand challenges of the 21st century (Atkinson & Mayo, 2010; Trilling, 2010; Resnick, 1998). Two of six goals released as part of NASA's 2011 Strategic Plan have direct relevance to STEM and education ("NASA Strategic Plan 2011," 2012) and earlier educational programs have tried to address them (Allner, et al, 2010).

Collaborative gaming and associated competition refers to the recent gaming phenomenon called 'massively multiplayer online role-playing games' (MMORPGs) (Yee, n.d.). Examples of gaming applications include Guitar Hero, Nintendo's Wii and Alternate Reality Games (ARGs) (Kim, Allen, & Lee, 2008). Literature has shown gaming to have tremendous positive effect as an educational tool: blissful productivity, urgent optimism, working in a collaborative environment and toward something agreed upon as an 'epic win' (McGonigal, 2011).

Satellite formation flight is the concept that multiple satellites (e.g. satellite constellations) can work together in a group to accomplish the objective of or do better than one larger, monolithic satellite (Folta, Newman, & Gardner, 1996). A satellite cluster is a type of constellation where all the modules need to fly within a specific range of each other (communication range, sensing range, data transfer range, etc.) in orbit in order to be functional. This requires solutions to multi-body problems in Earth orbit, precise determination of position, orientation and time, advanced control algorithms, trajectory planning, collision avoidance (Nag & Summerer, 2013) and many other issues. Examples of cluster flight are the DARPA F6 (O'Neill, Yue, Nag, Grogan, & De Weck, 2010), TechSAT-21 and PROBA-3.

This paper provides a proof of concept that crowdsourcing of cluster flight problems as well as CS-STEM education is possible and beneficial using the same program and analyzes the effects of participant collaboration through different mechanisms on both crowdsourcing and CS-STEM education. There have been successful applications of individual and combined topics (graphically represented in Figure 1) exemplified by the Foldit game for demonstrating protein folding to make uotherwise unsolved structures (Cooper et al., 2010), simulation gaming at MIT (Klofper, 2008), spectroscopy game (Bradley et. al, 2009) and the successful solution of the DARPA Red Balloon Challenge (Pickard, & Pentland, 2011). Our application lies at the intersection of crowdsourcing, collaborative gaming and CS-STEM education and the SPHERES Zero Robotics Program (ZR) was developed, observed and analyzed to achieve these goals.

Figure 1. Research Venn diagram for 'filling the gap'



2. SPHERES ZERO ROBOTICS PROGRAM

ZR is an international robotics programming competition where the robots are SPHERES satellites. SPHERES, built in 1999 and operational since 2006, consists of a set of tools and hardware developed for use aboard the ISS and in ground-based tests: three nanosatellites, a custom metrology system (based on infrared and ultrasound time-of-flight measurements), communications hardware, consumables (tanks and batteries), and an astronaut interface (Saenz-Otero, 2005). ZR allows students and amateur enthusiasts to play challenging games first on a high fidelity simulation and then on real SPHERES hardware in microgravity, and therefore demonstrate flight-capable programs. In order to practice programming SPHERES, manage their teams, participate in competitions and submit projects, the students have access to an elaborate website, integrated development environment (IDE) and an online simulator. The program started with a pilot of 2 US high schools in 2009, expanded to 24 in 2010 and 145 US and EU high schools in 2011. Although the program has been/is funded by DARPA, NASA and ESA, our vision is to seek corporate funding in future years (like FIRST Robotics (Melchoir et. al, 2005)) for tournament design and management with NASA and ESA support for astronaut crew time.

The ZR program was modified in 2011 to solicit complex trajectory tracking algorithms, and collaboration was introduced in the previously competition-only tournament structure. In 2011, three different and independent types of collaboration mechanisms were introduced – (1) *in-game collaboration,* (2) *alliance based collaboration* and (3) *forum based collaboration*. First, the game and scoring was designed such that opponent SPHERES (controlled by opponent projects) would be incentivized to collaborate, real-time in a match, to achieve game objectives (i.e. crowdsourcer objectives). Competitions were a round robin of matches and the team with the maximum cumulative points over all matches played won the competition, not the one with the maximum number of wins. Second, halfway through the tournament (after 3D#1 in Figure 4), it was mandatory that selected teams form alliances of 3 teams each and submit one project per alliance for all subsequent competitions. From this point onward, opponent SPHERES in a match would be controlled by projects submitted by opponent alliances, not teams. Third, the 2011 tournament had extensive community forums where teams could exchange ideas, educate each other, challenge each other to informal games and share projects to work on collaboratively. The ZR program was therefore modified in 2011 to be steered in a direction that would help evaluate its impact on both crowdsourcing and CS-STEM Education and assess the impact of collaborative competition on both these objectives.

Furthermore, the entire web interface for ZR (for participants to use the SPHERES simulator) was developed using crowdsourcing contests in collaboration with a commercial company called TopCoder Inc., based on a prototype developed at MIT. TopCoder is a commercial company that uses a mix of competition and collaboration within their online community of over 300,000 developers, who voluntarily register on TopCoder's website, to make scalable, cloud-based software systems. Thousands of developers competed in TopCoder contests for prize money. The intent was to prove that end-to-end crowdsourcing of spaceflight software, i.e. developing the web interface *by* crowdsourcing and then using it *for* crowdsourcing, is possible and beneficial (Nag, Heffan, Saenz-Otero, & Lydon, 2012).

Specific research methods to analyze the achievement the goals of crowdsourcing of cluster flight algorithms, collaborative gaming and CS-STEM education using ZR program include:

1. A case study of the web interface development for the program using TopCoder crowdsourcing contests (Nag et al., 2012);
2. Design of social experiments (Babbie, 2010) based on the observational data collected from the ZR Tournaments;
3. Statistical analysis of tournament data to interpret the educational value of the ZR and the effect of collaborative competition on crowdsourcing and education (Nag, Katz, & Saenz-Otero, 2013);
4. Data analysis of satellite telemetry returned after hardware operations of SPHERES on the ISS based on well-established methods and standards;
5. Systems dynamics modeling to explain the causal effects of the overall framework of crowdsourcing and education (Nag, 2012).

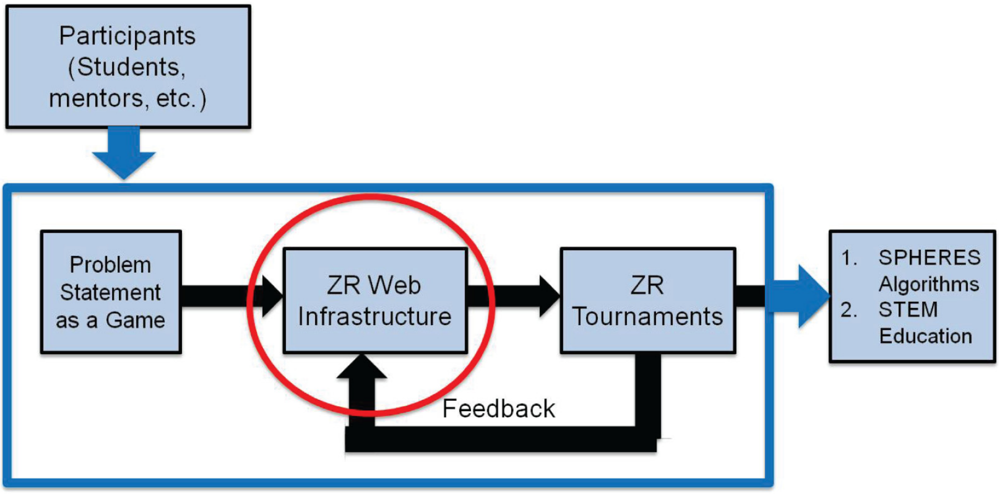
Detailed analysis can be found in previous literature (Nag, 2012). The findings will be summarized in this paper with special focus on (2) design of experiments and (4) satellite telemetric analysis.

2.1. System Representation

Spaceflight software development through Zero Robotics occurs for existing hardware in two stages, as shown in Figure 2: (1) Building the web infrastructure for the programming competitions – circled in red - by leveraging a crowd of thousands of software developers, and (2) the programming competitions themselves – within the blue box - when thousands of students contribute to writing SPHERES software. Both stages are demonstrations of crowdsourcing using different classes of participants and with different objectives.

As depicted in Figure 2, the students who participate in the tournaments are the input into the Zero Robotics 'system' and the output are the dual research objectives, STEM education and satellite software or algorithms. The 'blue

Figure 2. Zero robotics system diagram. Crowdsourcing occurs at two levels represented by the red circle and the blue box.



box' of crowdsourcing therefore has a dual impact of algorithm development *and* education. The 'system' includes a game which is accessible through the ZR Web Infrastructure, to be played during a ZR Tournament. TopCoder crowdsourcing contests led to designing the web infrastructure for these students (crowd creation) and assembly of the software components. This robust web framework allowed writing and testing satellite control programs online (crowd production). The feedback of the students, as they participate in the tournaments, serves to improve the web infrastructure - 'red circle' in Figure 2. Therefore, ZR demonstrates two types of crowdsourcing and together they achieve the dual goals of developing cluster flight software and educating students.

2.2. Web Infrastructure

Users can program the SPHERES using a web-based GUI, which provides a simplified interface to the SPHERES Guest Scientist API functions and enforces constraints that guarantee compatibility with the SPHERES compilers (Nag et al., 2012). Students have access to a text-based editor as well as a graphical editor (for those with little or no prior programming experience) through the Integrated Development Environment (IDE) which allows students to create, save, edit, share, simulate and practice as well as submit computer code for competitions. A distributed computation engine, hosted on Amazon EC2 virtual machines, compiles the user code with the core SPHERES software (game code and SPHERES embedded system code), and performs a full simulation of the program. An Adobe Flash-based front-end visualization creates an animated representation of the results. The projects created by the students via the web interface can be submitted to MIT during formal tournaments, compiled with the core SPHERES software, uploaded on the SPHERES hardware and executed on the ground (nearly frictionless, 3 DOF, Flat Floor Facility) or ISS (6 DOF, microgravity, ISS Facility).

The ZR website also hosts community forums, tutorials, support forums and a variety of management tools for users and administrators. The users have registration, account, team and project management tools for sharing code with each other, informally challenging others, collaborating and formally submitting to competitions. Administrators of the tournaments (MIT) have tournament management tools to collect and run batch simulations of the hundreds of programs submitted from participants and declare the results.

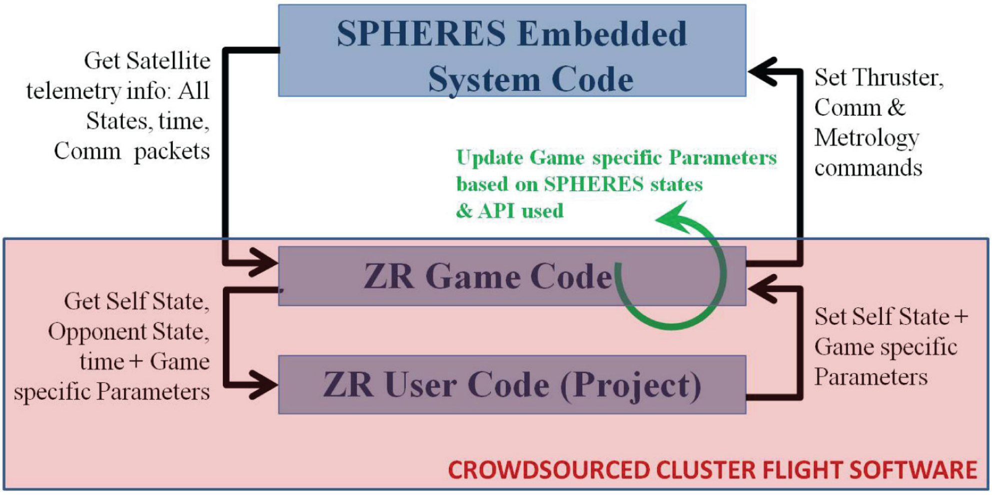
2.3. Zero Robotics Tournaments

ZR is the umbrella program under which multiple *tournaments* are held. A tournament is a series of *competitions* which cater to the same group of participants (e.g. high school students or middle school students) and require one application to be submitted to participate in the tournament. In 2011, all applications had to be from *teams* and not individuals. A *competition* is a bracketed set of matches among the participants (e.g. round robin, double elimination) at the end of which a ranked list can be declared. During a pre-defined period of time between two competitions, participants (teams or alliances) write programs online to play the pre-defined game and submit their final project for the purpose of an automated competition. Each competition is associated with a single game and can be of 3 types: simulation, ground and ISS (Nag et al., 2013). Each competition essentially represents a stage of the software testing process, during its development cycle itself, by subjecting user projects to 3 environments of increasing fidelity. For a simulation competition, MIT runs a batch simulation of all submitted projects within appropriate brackets. Ground competitions require uploading all the submitted projects compiled with the SPHERES Core Code on the SPHERES hardware and running the bracketed matches on the flat floor while ISS competitions entail doing the same on the ISS. A *match* is a head-to-head run between two SPHERES satellites, in simulation or hardware, controlled autonomously by *projects* written by teams or alliances using the ZR web interface. Thus, projects may also be called artificially intelligent *players* or *user code*. Opponent *projects* control one SPHERE each and are (each) given an automatic score at the end of the match.

The ZR Game is the problem statement that participants solve and the game code is the layer between the user code submitted by participants and the SPHERES embedded system (ES) code (Nag et al., 2013) as seen in Figure 3. It relays user commands to the ES layer, informs the user code of the SPHERES states and evaluates the performance of the SPHERE with respect to the game goals. The scoring in the 2011 ZR tournament was designed such that more robust algorithms, as per predefined metrics, scored higher points. Therefore, by designing a hard cluster flight problem as 'game code', inviting participants to play the game by writing 'projects', compiling the projects and game code with the SPHERES embedded system code to test on space hardware, it is possible for amateur crowds to develop new and improved algorithms for complex formation flight maneuvers.

The 2011 game was collaborative in nature (*in-game collaboration*) and called 'Astero-SPHERES' (MIT ZR Team, n.d.-a). The game had 2 versions, one for a 2D playing environment that culminated in a simulation competition and a ground demonstration and another for 3D that culminated in multiple simulation competitions and an ISS competition (Figure 4). The theme was asteroid mining, and it was based on the premise of NASA's future missions to explore near Earth objects (Nag et al., 2013). The game had 3 phases and their objectives, respectively, were: (1) Pick up some items from an available spread by 'docking' to them, (2) melting the virtual ice on a virtual asteroid through precise attitude control and/or mining the asteroids by spinning on or revolving around them, (3) racing and docking to a pre-defined virtual mining station for bonus points. For every time step in a match when 2 players (submitted projects by teams or alliances) collaborated to mine the same asteroid and/or helped each other to reach the mining station, each received double the points that they otherwise would have. The match score of a player (minimum = 0, maximum = 23) was calculated by summing the total number of points accumulated by that player in the 180

Figure 3. Block diagram of the three layers of the software that run the autonomous SPHERES satellites. The red block now represents the formation flight software developed through ZR's crowdsourcing efforts, by combining the game code layer – where the problem was coded – and the user code layer – where in the students code the solution to the game. Together, they command the SPHERES embedded system to achieve formation flight maneuvers.

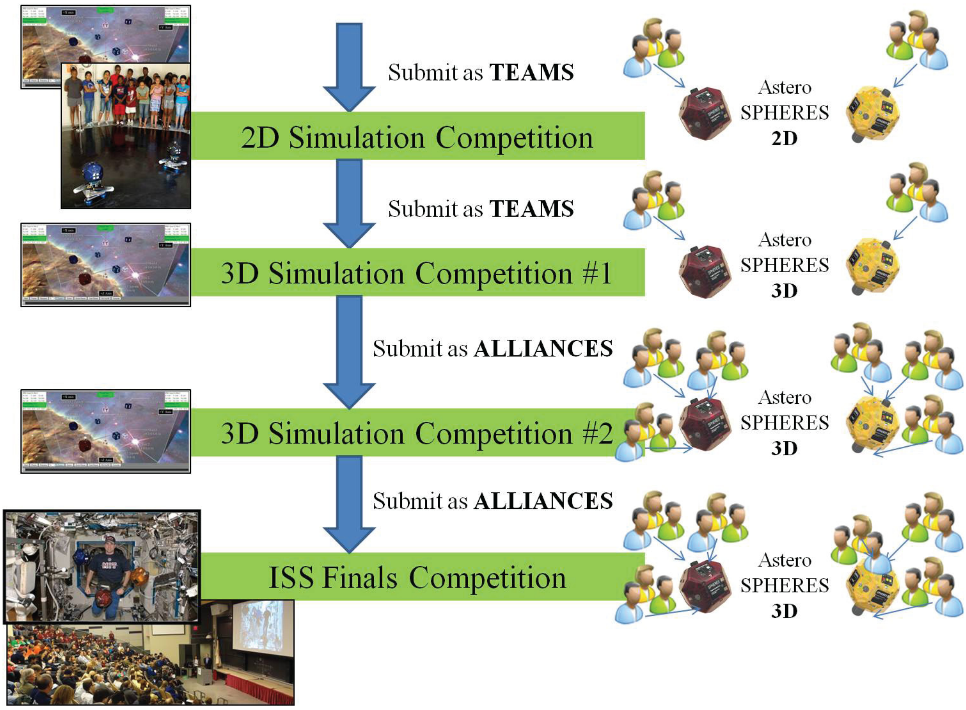


seconds of the match. The competition score for any player was the sum total of the scores over all the matches played by that player. Thus, it was in participants' advantage to make the SPHERES collaborate real-time *within each match* (red vs. yellow in Figure 4 when playing 2D or 3D, as teams or alliances) to maximize the score rather than just beat the opponent. This also implies that the competitions were scored such that the players which could achieve the match/game objectives and maneuvers, averaged over all opponent and environmental situations, emerged higher than those who were not as capable.

The 2011 tournament required that the 54 semi-finalists, chosen from all participating teams after the elimination rounds (after 3D#1 in Figure 4), form 'alliances' of 3 teams and work together as one team to prepare one submission per subsequent competition (*alliance based collaboration*). The project/player submitted by alliances would autonomously control opponent SPHERES in the matches (red vs. yellow in Figure 4) just as they did for previous team-based competitions, thus ensuring the independence of in-game and alliance-based collaborations. Alliances were formed by an automatic algorithm, taking into account preferences of partnering teams and the relative seeding of teams (based on ranks in the previous competition that divided them into tiers).

To aid in-game collaboration, Astero-SPHERES allowed the players to transmit unsigned short typed messages to the opponent player and receive the opponent's messages once every second. The website provided discussion forums for teams to communicate with each other and the game developers on the topic of programming/educational materials, brainstorming for strategies of collaboration within the matches, debating communication protocols within the limited bandwidth of data transmission between the SPHERES satellites and many other competition related interests. Thus, *forum based collaboration* was essential for effective in-game collaboration.

Figure 4. Sequence of competitions within the 2011 HS Tournament with the competition environment shown as thumbnails on the left and the submission format and game on the right. The 2D competition required teams to submit projects to play AsteroSPHERES 2D. The 3D competitions required first teams (in 3D#1) and then alliances (in 3D#2 and ISS Finals) to submit projects to play AsteroSPHERES 3D. There were ~ 3 weeks for teams to play the game associated with the competition and submit their projects via the website for the formal competition.



3. IMPACT OF COLLABORATIVE CROWDSOURCING – RESULTS FROM THE TOURNAMENTS

The primary sources of data to assess the benefits of crowdsourcing are the scores of teams in simulation and hardware competitions which reflected their formation flight performance, the maneuvers they demonstrated and the software they submitted. This paper does not demonstrate solutions to unsolved spaceflight problems, propose better solutions obtained through ZR crowdsourcing compared to existing literature or describe the process of technically integrating the crowdsourced modules with existing cluster flight software. We do, however, claim that students are capable of solving hard formation flight problems, given the ZR tools and access to the SPHERES simulator, in short periods of time. We demonstrate a process by which difficult cluster flight problems can be 'gamed' such that students can contribute to solving them (Figure 3), game scoring designed such that students' solutions reflect the achievement of algorithm objectives and the solutions tested in simulation and on hardware for quantitative evaluation of efficiency. We propose that the same process can be applied to real, cluster flight problems.

The 2010 tournament, in spite of not being a directed crowdsourcing effort or having any incentivized collaboration, taught us many lessons on designing a good collaborative, crowdsourcing tournament (Nag, 2012). In order to achieve crowdsourcing objectives for the development of cluster flight algorithms, it is best if the game is designed to allow both SPHERES to work with each other, like a real cluster, to truly demonstrate formation flight (*in-game collaboration*). This also maximizes the resources available on the ISS as expected for a research-based test than a competitive demonstration where each SPHERE behaves independently. Games require the right mix of competition and collaboration to maintain the excitement of elimination while achieving game objectives. Lastly, we learned that game scoring should be designed such that the scores reflect accurately the quality of the crowdsourced solution so that there is fine and quantitative resolution between the hundreds of solutions submitted, as opposed to only 3 possible scores in 2010. In 2011, the game objectives were designed such that perfect scores were possible only if both SPHERES in a match collaborated to achieve the objectives together, physically and strategically, so participants were incentivized to write collaborative players. Game scores were designed so as to exactly prorate the quality of the solution to the complex maneuver sought. All the maneuvers required in the match, such as position control, attitude control, controlled spinning and controlled rotation have been demonstrated in scientific literature as well as the SPHERES testbed using different methods. The ability of students to program these maneuvers and combine them to solve a complex, formation flight but proxy problem was tested.

The 2010 tournament did not have *alliance-based* and a well-advertised *forum-based* collaboration either, thus could not tap the potential benefits of learning from other teams in the tournament (education benefits) and playing the game better (crowdsourcer benefits). Alliance based collaboration had further educational benefits because three times the number of 2010 teams could experience their projects run on the ISS hardware.

3.1. Using In-Game Collaboration to ‘Solve’ the ZR Game

The 2 main proxy problems to be solved by crowdsourcing, used in the 2011 game, were to write fuel-efficient algorithms for two activities:

1. Spinning a SPHERE at a predefined orientation, angular velocity and position while another SPHERE revolves around it at very close proximity, without colliding;
2. Revolution of a SPHERE about a fixed position spinning SPHERE, in a pre-defined plan, at a predefined velocity and within a predefined close proximity radius without colliding.

These formation flight maneuvers are useful for close proximity inspection by an inspector satellite (the revolving one with controlled attitude) of a target satellite (the spinning one). Also, the spinning satellite may be considered analogous to a tumbling target and the revolving satellite analogous to a satellite demonstrating docking to a tumbling target (Nolet, Kong, & Miller, 2005). The game aspect designed around this proxy problem was themed on mining virtual ‘asteroids’ whose position in the game volume was fixed and known to participants. Maximum points for mining could be obtained only if a team programmed the SPHERE to follow a precise and efficient trajectory around the asteroid location while the other (opponent) SPHERE spun at the asteroid location. ‘Precise trajectory’ meant moving at a specific angular velocity, within a specific annular ring and in a specific plane of revolution. The SPHERES were allocated a finite amount of virtual fuel, which was a predefined fraction of the real fuel, to perform all their tasks in a match, so the maneuver was additionally required to be fuel efficient. The points were prorated depending on which asteroid was mined, at what angular velocity, what orientation and whether alone or collaboratively.

The perfect score in a competition (not only a match) was possible only if a player had perfectly "in-game" collaborating opponents to get the best out of all the available resources, a perfectly optimized strategy of war-gaming and a perfect control algorithm for trajectory tracking of the SPHERE. War-gaming here refers to a robust and autonomous decision making strategy that teams would code within their submitted player such that, during the match, their SPHERE would respond smartly to the opponent's SPHERE and achieve the game objectives, without getting in each other's way. AsteroSPHERES was designed such that every phase in the game presented the players with several choices for their course of action, all of which were not possible in the constraints of time and fuel. Participants were expected to come up with a war-gaming strategy that best supported the technical capabilities of their programmed SPHERE and expectations of their opponents' behavior – as truthfully communicated or otherwise.

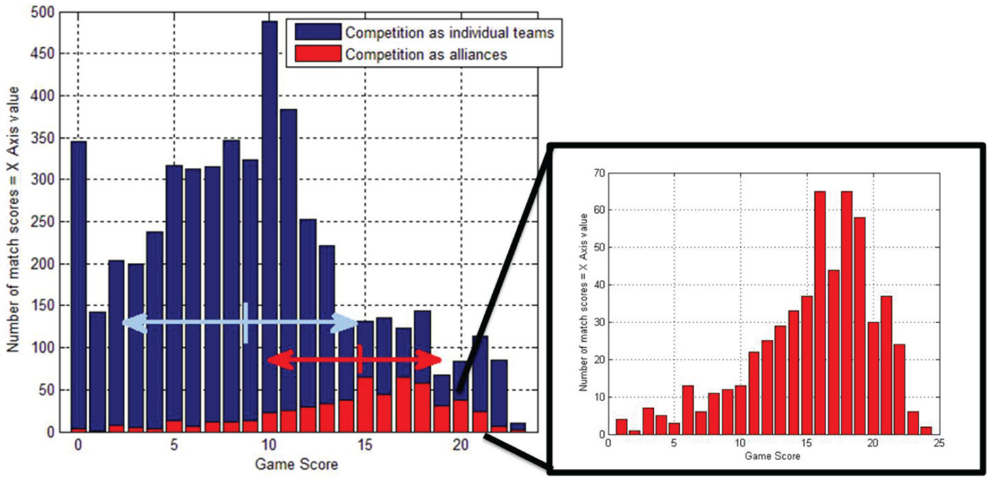
The students were able to come up with efficient trajectory tracking algorithms *and* war-gaming strategies for in-game collaboration. They maintained large spreadsheets of calculations and even pointed out flaws in the game and/or inconsistencies during the tournament, that the ZR Team corrected when moving between one competition to the next, hence improving the problem statement. The teams demonstrated analytical, strategic abilities and collective intelligence. 88 projects (from 88 teams, one project per team was allowed) were received for the 2D competition and 91 projects for the 3D #1 competition. Among the 91 teams that submitted for 3D #1, 4 unique teams in 10 different matches were able to achieve the perfect score of 23 points – within 3 weeks of the game's release and 6 weeks of the IDE's release. The histogram of scores of all the matches in the competition, i.e. each team's project against every other team's project, is shown in blue in Figure 5. These high performing outliers show the impact of crowdsourcing in problem solving. Since both competitions (2D and 3D#1 in 2011) had in-game and forum-based collaboration (neither of which was incentivized in 2010) but not alliance-based, they show the impact of the former in solving harder cluster flight problems than in 2010.

3.2. Effect of Alliance and Forum based Collaboration on Solutions

An alliance (of 3 teams) in 3D#2 controlled a SPHERE in the game in exactly the same way as a team in 3D#1 did and thus were subjected to the same scoring. Effects observed due to alliance-based collaboration (mandatory 3D#2 onward) are, therefore, assumed to be independent of in-game or forum-based collaboration (incentivized in all competitions of the tournament). The results from 3D#1 and 3D#2 competitions in the tournament indicate that alliances of teams showed higher average scores than individual teams, demonstrating the importance of *alliance based collaboration*. As seen in Figure 5, the mean score among all the teams has improved significantly after grouping the teams as alliances. To mitigate the effect of selection bias on experimental validity, only teams that participated as alliances in the 3D #2 Competition were chosen for analysis in the 3D #1. See Figure 4 for the sequence of competitions. The mean score of the team competition (3D#1) was 9.1 (standard deviation of 5.6) and the mean score of the alliance competition (3D#2) was 14.6 (standard deviation of 4.6).

The mean of the alliance scores is more than one standard deviation greater than the mean of the team scores. However, the scores are not normally distributed (by the Kolmogorov Smirnov test), hence a t-test could not be used to calculate the differences. The interpretation of this difference in scores is further complicated by possible learning over the three week interval as well as minor modifications in game rules between the competitions. For instance, while it was entirely possible to have a perfect score in the first competition by programming a perfectly collaborating, strategic revolve maneuver around the "richer" asteroid, getting a perfect

Figure 5. Comparison of score distributions (range = 0 to 23 points) with and without alliance-based collaboration. The blue bars are the scores of teams in 3D#1. The red bars are the scores for alliances in a separate competition, 3D#2. The blue histogram contains 4095 round robin match scores, played between every pair of the 91 teams. The 72 highest teams were formed into 24 alliances, of 3 teams each. Thus, there were 8190 (blue) and 276 (red) matches in 3D#2. The mean and standard deviation of each set is shown in the Figure using the vertical line and arrows.



score in the second competition additionally required a perfectly timed trajectory and a perfect maneuver to dock to the mining station.

The score distributions of the 2D and 3D #1 Competitions were compared to find the effects of the learning period and game rule changes. Both competitions had a 3 week period of preparation/programming (schedule in Figure 4), team participation and modification of game rules. While the competitions received 88 and 91 submissions respectively, only the 70 teams that participated in 3D #1 were chosen for analysis in 2D. The mean 2D score was 6.2, standard deviation 4.78, and the mean 3D score was 7.83, standard deviation 5.6 (Figure 6). From the figure, it is easy to visually interpret that the improvement in the mean between 2D and 3D#1 is far less than the improvement in mean from 3D #1 to 3D #2. It is therefore reasonable to conclude that a larger share of improvement in game scores when alliances were introduced was due to the existence of the alliance variable rather than the learning and game rule modification variables. This conclusion, however, assumes that the combined effect of game rule modifications and participant maturation between the two sets of competitions is equivalent. This assumption is cannot be verified because neither set is quantifiable and both are unrelated changes. No other control was available for this observational study.

While alliances apparently affected the overall scoring of participants, they showed varying effects on the number of perfect solutions obtained i.e. the right tail of the histograms in Figure 5 and Figure 6. Table 1 shows the number of unique 23-scorers in the 2D competition, 3D #1 and 3D #2 and the number of matches they achieved the perfect score of 23 - normalized. The decrease in the number of unique players with the sequence of competitions indicates no change from 3D#2 to 3D#1, over the 3D#1 minus 2D control (Equation 1 where metric = number of unique players). However, the normalized number of matches that achieved a perfect score (as the metric in

Figure 6. Comparison of the 3D #1 with the 2D scores (both played as teams). The score range was 0 to 23 points). The 3D #1 competition is the same as that shown in the left panel of Figure 5, but only those (70) teams that played both 2D and 3D#1 were chosen for analysis. The mean and standard deviation of each set is shown in the Figure using the vertical line and arrows.

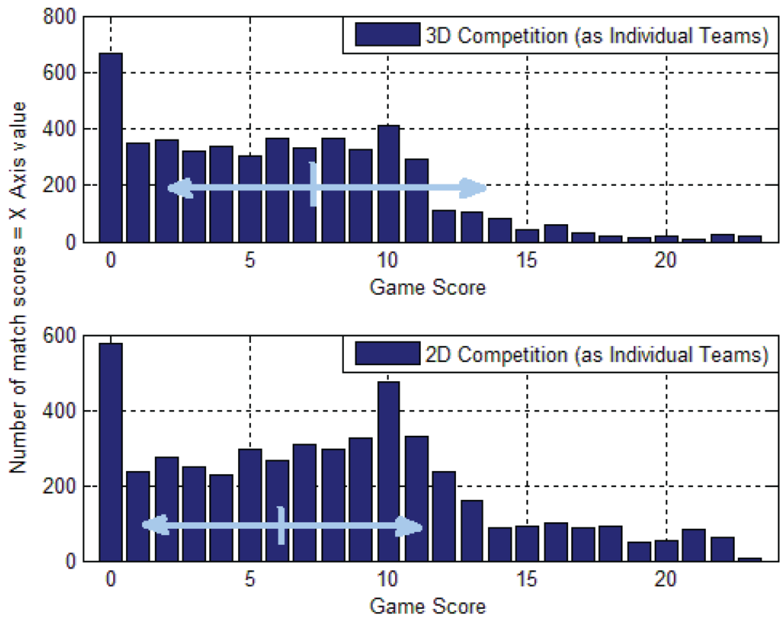


Table 1. Comparison of the perfect solutions obtained through the 3 simulation competitions. The number of matches in the third column has been normalized by the number of matches each player played in the competition. For example, there were 88 submissions of projects by 88 teams in the 2D RR competition, so the number of matches each played was 87.

Competition	Number of Unique perfect players	Normalized number of matches where the perfect score was accomplished
2D as Teams	6	21/87 = 0.27
3D#1 as Teams	4	10/91 = 0.11
3D#2 as Alliances	2	2/23 = 0.09

Equation 1) showed an increase with the introduction of alliances. The decrease in the perfect players between individual competitions seems to indicate that the game changes were harder than what the participants could pick up in a three week learning period.

The tier-based alliance selection process therefore ensured that no perfect solutions could be eliminated in the process due to dilution with each other. The decrease in the number of unique perfect players from 3D#1 to 3D#2 could not have been because teams that made perfect players came together as an alliance because the alliance forming process used a tier-based system. No top performing teams could have joined together and all could continue to further their own strategy.

$$changeDueToAlliances = \left[metric(3D\#2) - metric(3D\#1)\right] - \left[metric(3D\#1) - metric(2D)\right] \quad (1)$$

Teams used the discussion forums on the website to come up with a global communication protocol (*forum based collaboration*) beyond the standard messages that were published for SPHERE-to-SPHERE communication to cooperate more efficiently and play the game better. For example, the most popular protocol called for SPHERE1 to perform revolution as the mining maneuver and SPHERE2 to perform spinning, both on a pre-decided asteroid. The protocol set aside one message, by broadcasting which the SPHERE declared to its opponent that it was following this protocol, so that the opponent could respond accordingly. By the end of the tournament, all teams followed this protocol demonstrating effective diffusion.

3.3. ISS Hardware Demonstration

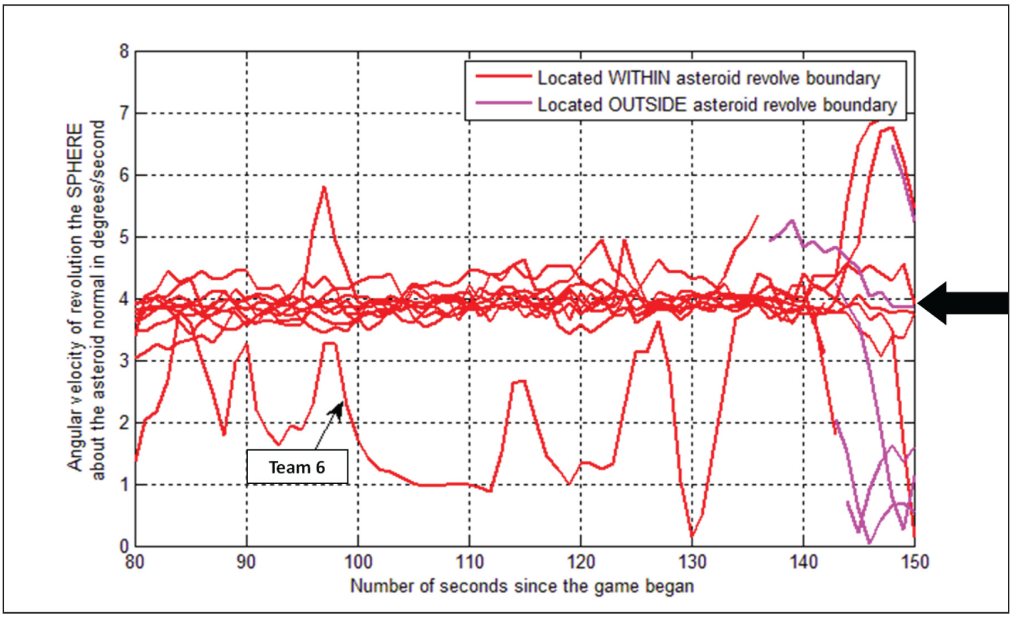
This section demonstrates ZR's success in demonstrating cluster flight solutions, sought online through a game and refined using 3 types of collaboration and 3 levels of competition, in the relevant microgravity environment. The ISS finals are culminating competition of any ZR tournament. Since the ISS projects have passed a sequence of simulation elimination rounds, they are the best players in the tournament i.e. the rightmost tail of the histogram distribution of scores (Figure 5, Figure 6). The best user algorithms developed through online crowdsourcing are finally tested on real satellites to select the most robust and efficient of them.

Each match in the finals started with the primary (1:red) and secondary (2:blue) satellites positioned in the same location after which they autonomously moved around to play the AsteroSPHERES. The satellite telemetry and state of health packets, as returned to the SSC (the JEM module laptop used to communicate with the SPHERES during a test session), are stored for plotting the dynamic states of the satellites in each match.

To analyze the efficiency of the trajectory-tracking algorithms for revolution and spinning about the virtual asteroids on SPHERES ISS hardware, the SPHERES states in the mining phases of each match were analyzed. Mining starts at least 10-20 seconds into the second phase i.e. *at least* 70 seconds after the start of the game. The mining phase may last for as long as the players' strategies permit, i.e. either till the end or until the SPHERES race to the mining stations. Figure 7 shows angular velocity of revolution maneuver of a SPHERE (for this competition, by protocol, always SPHERE1) around the asteroid location while there was a spinning SPHERE at the asteroid location. The instantaneous angular velocity of the SPHERE about the asteroid normal – 'angVel' in Equation 2 – was calculated using SPHERE1's position and velocity at every time step (200 ms). This process was repeated for all the successful matches on the ISS and all curves were plotted together in Figure 7.

$$angVel = \left| < \omega, \hat{n} > \right| * 180 \, / \, pi \quad (2)$$

Figure 7. Plot of the main mining phase (80 to 140 seconds after the start of the match) behavior of SPHERE1 over all the ISS matches, in terms of the angular velocity of revolution around the virtual asteroid on a plane perpendicular to its normal. The resonance angular velocity, for which maximum points were awarded per second, was 4 degrees/second, marked with a thick black arrow. The plot color (red) indicates that SPHERE1 performed the revolve maneuver. The magenta sections indicate the angular velocity when the SPHERE was revolving, but out of the annular shell of point accumulation (outside 10cm-40 cm of asteroid location).



$$\omega = \frac{\left[\hat{r}, vel\right]}{\left|r\right|} \qquad (3)$$

$$\hat{r} = \frac{pos - asteroid}{pos - asteroid} \qquad (4)$$

Where:

pos = instantaneous position of the SPHERE,
vel = instantaneous velocity of the SPHERE,
$\hat{n}$ = unit normal of the asteroid axis,
|| || = norm of a vector,
<,> = dot product of the vectors enclosed,
[,] = cross product of the vectors enclosed,
ω = angular velocity of the SPHERE about the asteroid location,
r = radius vector from the SPHERE to the asteroid location.

Maximum points per second were awarded if *angVel* was 4 degrees per second, prorated as a linear ramp from 0 to 8 degrees per second, *and* if the revolving satellite was within 10 cm to 40 cm of the asteroid center (Nag, 2012). In Figure 7, it can be seen that in the main mining phase in almost ALL the matches, the SPHERE is indeed correctly positioned (red, not pink) and revolving correctly (close to black arrow). The one match where the player on the SPHERE did not perform well has been indicated with an arrow and the team that controlled the revolving satellite mentioned. Team 6 got 9 points for this match, the lowest in the competition among successful tests, but the *only* one where the revolve maneuver was not near resonance.

After 140 seconds of game time, Figure 7 shows that the angular velocities start dropping off and some satellites start leaving the revolution radius (specifically marked in magenta). This is a transition phase where some players chose to stay revolving while others began to leave for the mining station.

Figure 8 shows the spin velocity of the spinning maneuver of a SPHERE (for this competition, by protocol, always SPHERE2) at the asteroid location while there was a SPHERE revolving around it. The instantaneous spin velocity of the SPHERE about the asteroid normal – 'spinVel' (Equation 5) – is calculated from the attitude rate and quaternion of SPHERE2 at every time step (200 ms). This process was repeated for all the successful matches on the ISS and all curves plotted in blue in Figure 8.

$$spinVel = \left| < \bar{\alpha}, \hat{n} > \right| * 180 / pi \tag{5}$$

$$\bar{\alpha} = rotMat * \alpha \tag{6}$$

$$rotMat[1,1] = q4 * q4 + q1 * q1 - q2 * q2 - q3 * q3$$

$$rotMat[1,2] = 2 * (q1 * q2 - q3 * q4)$$

$$rotMat[1,3] = 2 * (q1 * q3 + q2 * q4)$$

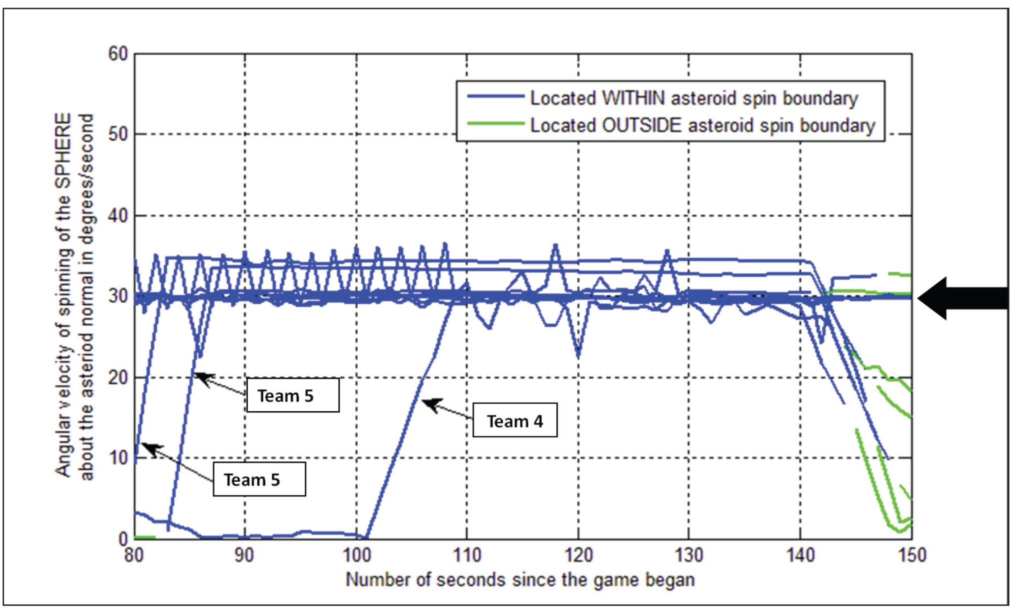
$$rotMat[2,1] = 2 * (q1 * q2 + q3 * q4)$$

$$rotMat[2,2] = q4 * q4 - q1 * q1 + q2 * q2 - q3 * q3$$

$$rotMat[2,3] = 2 * (q2 * q3 - q1 * q4)$$

$$rotMat[3,1] = 2 * (q1 * q3 - q2 * q4)$$

Figure 8. Plot of the main mining phase (80 to 140 seconds after the start of the match) behavior of SPHERE2 over all the ISS matches, in terms of the spin angular velocity about any body axis aligned with the asteroid normal. The resonance angular velocity, for which maximum points were awarded per second, was 30 degrees/second, marked with a thick black arrow. The plot color (blue) indicates that SPHERE2 performed the spin maneuver. The green sections indicate the angular velocity when the SPHERE was revolving, but out of the zone of point accumulation (outside 5 cm of asteroid location).



$$rotMat[3,2] = 2 * (q2 * q3 + q1 * q4)$$

$$rotMat[3,3] = q4 * q4 - q1 * q1 - q2 * q2 + q3 * q3 \quad (7)$$

Where,

α = instantaneous attitude rate of the SPHERE in its body coordinates,
$\bar{\alpha}$ = instantaneous attitude rate of the SPHERE in global/ISS coordinates,
[q1, q2, q3, q4] = instantaneous quaternion of the SPHERE,
$\hat{n}$ = unit normal of the asteroid axis,
|| || = norm of a vector,
<,> = dot product of the vectors enclosed,
[,] = cross product of the vectors enclosed,
$rotMat$ = rotation matrix to transform SPHERE attitude from local to global coordinates using the instantaneous quaternion.

To be awarded maximum points, the spinning SPHERE had to spin at a spin angular velocity (*spinVel*) of 30 degrees/second. For all other values between 0 and 60 degrees/second, the score was linearly prorated. To score, the spinning SPHERE was required to be positioned within 5 cm of the asteroid location. Figure 8 shows that the spin velocity curves bunched up at the resonance spin velocity of 30 degrees/second (black arrow) for a significant amount of the mining time period. After 140 seconds, some players continue to spin while others drop off their spin and move away toward the mining station – green curves instead of blue. Three players started their spin maneuver late into the mining phase in spite of being correctly positioned (blue curves at time < 100s). Team 5 scored 8 and 9 points and Team 4 scored 6 points in the match indicated. Team 5 lost points since they controlled their spin velocity, very accurately, at an angular rate ~ 4 degrees/sec higher than the resonance velocity. Team 5 reported later that this was due to a bug in their project, and evidently an outlier among all teams. The match between Team 6 (revolve) and Team 4 (spin) was the lowest scoring match in the competition among all successful tests, since Team 6 did not do well either (Figure 7). However, the match was certainly an outlier among the 12 successful matches.

The proxy problem of precise revolution or spinning of a SPHERE, at exact positions, orientations and angular velocities, gamed in the form of mining asteroids, therefore yielded very robust algorithms from high school students that could perform efficiently even on space hardware and in microgravity – demonstrating the value of crowdsourcing through ZR. Equation 2 to Equation 7 are the same equations used in the game layer (Figure 3) to determine the dynamic scores of the satellites due to revolving or spinning from their state vector telemetry during the mining phase – showing the efficiency of the game scoring mechanism to determine the 'best' crowdsourced solution.

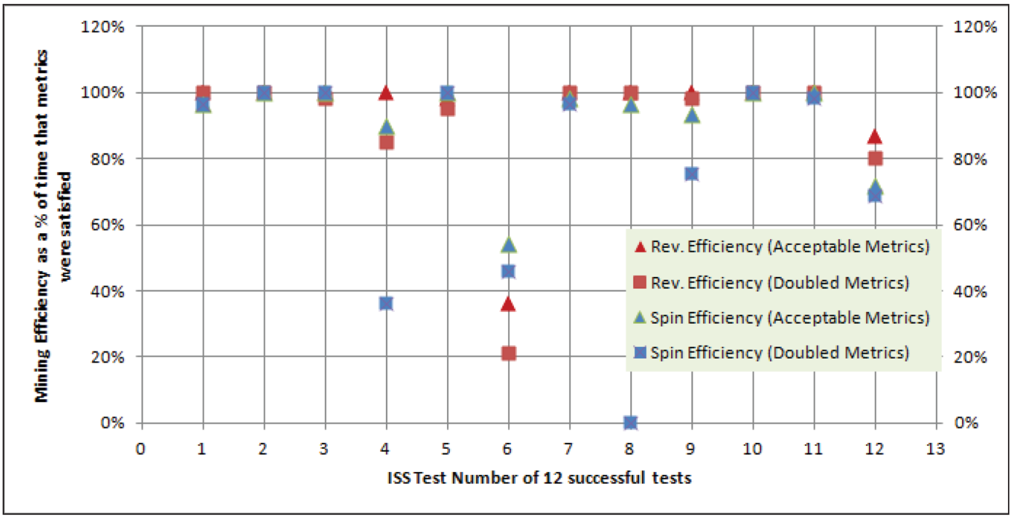
Efficiency of both mining maneuvers, revolution and spinning, has been calculated as the percentage of time that the players spent within acceptable error levels of angular and spin velocity respectively, for each test/match of the ISS test session (see Equation 8 in Box 1). For every test, only the time between 80 – 140 seconds since the match started is used to calculate efficiency, since that period is the main mining phase as described earlier, i.e. the summation in Equation 8 are for tStep ϵ [80, 140]. The acceptable error level for angular velocity (Equation 2) is assumed to be 1 cm/s about the asteroid location and that for spin velocity (Equation 5) to be 6 degrees/s. These values are chosen as per successful test definitions within SPHERES research framework.

The average efficiency of revolving and spinning players over the main mining phase of all successful ISS tests is 93.5% and 91.8% respectively when acceptable error levels are used (triangles in Figure 9). Seven of ten tests show 100% efficiency in revolution and the main outlier is Test 6 due to Team 6's imperfect control, also seen in Figure 7. Test 6 is also a spinning maneuver outlier, due to Team 4's late start, also seen in Figure 8. As mentioned before, this was the lowest scoring and exceptionally

Box 1. Equation 8

$$efficiency\left(test\right) = \frac{\sum_{tStep}\left[resonanceVel - errVel \leq ang \,/\, spinVel(tStep) \leq resonanceVel + errVel\right]}{\sum_{tStep} 1} \quad (8)$$

Figure 9. Efficiency of the revolve (blue) and spin (red) mining maneuvers with respect to regular metrics (triangles) and stricter metrics (boxes), over the time span of the main mining phase i.e. between 80 and 140 seconds since the match begins (X axis of Figure 7 and Figure 8)



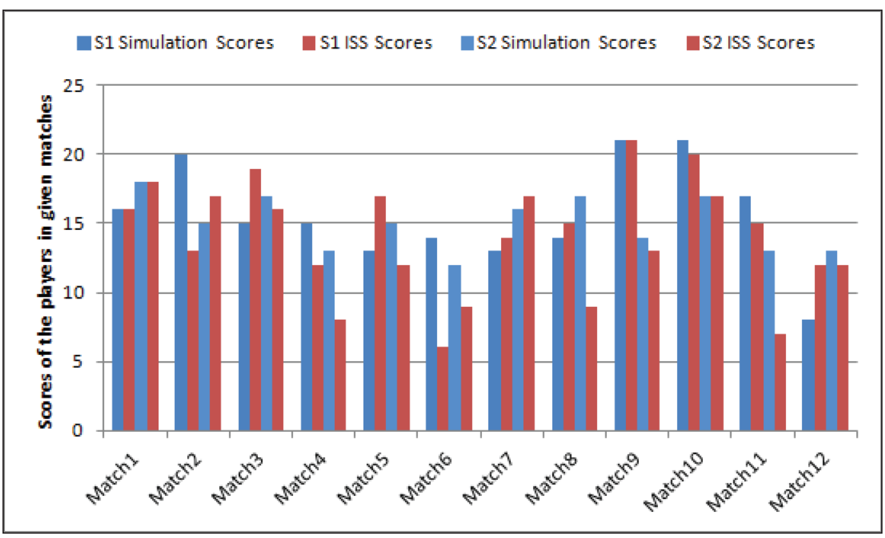
underperforming test among the ISS tests. The above analysis repeated for acceptable error levels *halved,* i.e. within 5 mm/s (about the asteroid) of resonance for revolution and within 3 degrees of resonance for spinning, the average efficiency is 90% and 76.5%. The main outlier that causes the drop of spin efficiency is Test 8, due to the imperfect control of Team 5. Since they apparently controlled their attitude rate at 34 instead of 30 degrees/s, their control was very efficient but at the wrong resonance velocity, hence 100% efficiency at acceptable error levels but 0% at stricter levels. The average efficiency of spinning at stricter error levels, excluding Team 5, is 83.5%. Student teams, therefore, demonstrated that they were capable of writing very efficient control algorithms for hardware demonstration in space which not only met research acceptable levels (91-93%) but also doubled standards (76-90%).

All the matches that were declared successful on the ISS were run in simulation with higher noise levels in the SPHERES simulator (e.g. with asymmetric thruster firing, random noise in thrust levels) to compare the ISS performance of the players with respect to what the teams programmed them to achieve. The results of the simulated matches were compared to the ISS results in terms of 3 major metrics: Station docking results for each team in each match, mining robustness of teams and Match scores for both players in each match (Nag, 2012), to establish that the conclusions drawn in the last 2 sections is also applicable user programs are tested on space hardware. While our analysis showed that the submitted programs were able to achieve station docking as efficiently as mining - 80% robust (Nag, 2012) - only the last metric has been discussed in this paper in more detail.

Figure 10 shows the comparison of all match scores in 12 successful matches conducted on the ISS test session; 10 of the 12 US matches and 2 of 3 EU matches were successful. For the unsuccessful matches, the simulation results were used to determine the match winners for the championship award. The average difference between the simulation and ISS scores is 2.75 (with a standard deviation of 2.52), which is less than a sixth of the mining points possible. Also, the bonus for a racing to the station first was 4-6 points and winning the game was 2 points, so the difference in total scores could be because a different team won the station race and/or match by a hair's margin on the ISS vs. in simulation. Since the disparity between ISS and simulation is not significant, conclusions on the effects of collaboration on crowdsourcing of spaceflight software algorithms, based on the analysis of simulation scores, are valid.

During the highest scoring match of the ISS competition, which resulted in scores of 21 and 13 respectively, both SPHERES had ~23% of their virtual fuel remaining at the end. Furthermore, in the next best match of the competition, with scores of 20 and 17 per SPHERE, the fuel remaining was 41% and 61% respectively. Both these matches were played by 4 unique players i.e. 4 alliances of 12 teams each. This clearly indicates that the participants were capable of more challenging and resource-constraining scenarios which were not measured using the 2011 scores. If the score had been a function of fuel usage, the best players in the tournament could be sorted further for fuel-efficiency. For the middle school version of AsteroSPHERES, MIT undergraduates had written functions for the spin and revolve maneuvers as part of the game API library which MS students used to play the game. Analysis of the MS fuel usage *on simulation* reveals that the best performing revolver had at most 7% fuel remaining over all the matches that it played. Since the MS ISS event typically used about 4% extra fuel for SPHERES maneuvers com-

Figure 10. Comparison of the scores of both SPHERES in all the successful matches on the ISS with the scores when the corresponding matches were simulated on the SPHERES



pared to simulation, a *qualitative* comparison with the HS ISS results shows that the HS students showed more fuel-optimal performance compared to randomly selected MIT undergraduate students. This finding strengthens the theory that crowdsourcing done in an appropriate way can be educationally powerful and produce resource-efficient algorithms.

3.4. Dedicated Crowdsourcing Tournaments

ZR has also launched dedicated crowdsourcing tournaments as open registration events for anyone in the world to play even more challenging ZR games aimed to solve harder space control systems problems. For example, a recently concluded tournament within the program was the ZR Autonomous Space Capture Challenge (MIT ZR Team, n.d.-b), whose goal was to develop an algorithm for the recently announced DARPA Phoenix Mission to demonstrate technologies for cooperatively harvesting and re-using valuable components from retired, nonworking communications satellites in geosynchronous orbit. The Phoenix spacecraft is required maneuver itself into position and synchronize with a tumbling object such that tools can be extended to remove or attach necessary parts. In keeping with this goal, the objective of the ZR game was to write a computer program to control a satellite (called a "Tender") to enable it to dock with a space object (or POD) that may be tumbling through space. The best algorithm submissions from simulation competitions, conducted every week for four weeks, were tested in zero gravity on real SPHERES satellites aboard the ISS. The tournament was open to all age groups and all nationalities, unlike the HS tournaments which were for US and EU students only. In spite of these constraints and large turnout, the 3 finalists were high school teams(DARPA, 2012) – demonstrating ZR's ability to achieve dual crowdsourcer and educational objectives even with unsolved, open-registration problems.

Another use of ZR Tournaments for dedicated crowdsourcing is verification and validation (V&V) of the newly developed algorithms by introducing them as part of the ZR game code[1]. Since tens or even hundreds of thousands of simulations are run for each tournament, this will be an opportunity to test the algorithms on the SPHERES simulator by subjecting it to hundreds of programs written by a random sample set of people. ZR Tournaments are thus great platforms where new algorithms can be developed by designing a game around specific problems and inviting crowds to play it *as well as* developed algorithms can be tested by subjecting them to thousands of human-designed simulations.

4. SUMMARY OF EDUCATIONAL IMPACT

The results highlighted in the previous sections are important to gauge the benefits to STEM Education too, since performance of the teams in the ZR competitions was a metric for the value delivered by STEM Education. As published separately (Nag et al., 2013), surveys reported that student users found games and competition exciting and therefore learned math, science, strategy and programming while solving real-world problems by playing games The program saw participation grow by 241% between 2010 and 2011. Above 85% mentors and students have reported significantly positive improvement in CS-STEM and leadership skills and the predicted retention rate is approximately 89%. Average performance of teams through the tournament increased by 3.4 points on a 0-23 point scale. Participants attributed positive educational influence to all the collaborative features in ZR. Intra-team collaborative features were better received than inter-team features as indicated by their differential preferences, although at varying degrees of statistical significance. The feedback has shown us ways in which the collaboration implementation within

ZR can be improved to deliver better quality education and we have a framework in place for measuring the effects on our objectives.

5. DISCUSSION

Our studies indicate that there is a case to be made for combining collaborative competition, crowdsourcing and STEM education. The objective is to find a balance between the needs of the involved stakeholders: the scientific community, and educators and their students. The scientific community would want crowdsourcing contests to target specific skills. From their perspective, everyone in the crowd need not solve the problem. In fact, it is sufficient if a large group of people are interested enough to try and solve the multiple sub-problems till an overall solution emerges. This allows for selection of talent from a large pool but retention of only the truly motivated and capable. The education community would want to introduce students to broader skills, something exciting and dynamic that would maximize retention rate in STEM. They would certainly find the scientific approach too utilitarian.

Therefore, the game needs to be designed in a way that achieves optimal ground between stakeholder interests. We found that introducing collaboration helped these dual objectives by improving overall performance of teams (Figure 5) and number of perfect demonstrations in matches (Table 1), and reported educational benefits and increased ability to solve difficult formation flight problems in simulation and hardware.

5.1. Can Students Solve Problems that Scientists Cannot Solve?

This paper does not demonstrate the solution of an unsolved or unsolvable problem as DARPA Phoenix demonstration in Section 3.4 aims to do (not analyzed because recently concluded). Instead, it highlights the possible utility of crowdsourcing for solving cluster flight problems than scientists cannot solve, not necessarily due to scientific difficulty. Crowdsourcing may be useful if the potential crowdsourcer does not currently have resources within her organization to solve a problem or a subset of the problem or needs help with solving a problem – all for no additional cost over that of running an educational program.

HS Students have demonstrated, through ZR, that when mentored appropriately and with the right software tools available, they can outperform even MIT undergraduate students. Their solutions have demonstrated efficiency (>90% of submitted players, Figure 9) and robustness of control (>80% of players (Nag et al., 2013)) for precise formation flight even in random noise levels aboard the ISS. In fact, the top solutions achieved the cluster flight game objectives using less than a quarter of the fuel allocated to them (~23%) and 80-90% of the players were calculated as efficient even when using acceptable error levels twice as strict as usually acceptable in SPHERES research (Figure 7, Figure 8, Figure 9). This indicated that a much tougher problem could have been solved by students within this year's program. Finally, student feedback for alliance-based collaboration indicated that lack of enough work to distribute among all three collaborating teams was an important reason why high performers dominated the project finalization phase and lower performers felt abandoned. While this teaches us lessons on alliance formulation and games for future years, the case in point is that many teams felt they could have contributed *much more* than what they did. Hence, in the presence of alliance-based collaboration, more difficult problems should be welcomed as a challenge.

In spite of admirable student capabilities, does the introduction of crowdsourced components introduce added risk to mission avionics? First, restrictiveness (enough to prevent innovation) or openness (enough to break the core software) of crodsourced software depends entirely the game developer/crowdsourcer as a function of the software components required, modularity of the embedded system (ES) and availability of hardware to test on. The game layer in Figure 3 may be flexibly coded to filter user commands and evaluate the SPHERE performance appropriately. Second, crowdsourced software will be subjected to the same, rigorous testing process that professional software is. The onus to insert the modular crowdsourced components in the correct location and to make sure that the integrated product works is on the crowdsourcer. While the competitions of 2011 (Figure 4) demonstrated 3 stages of the software testing process by subjecting user projects to 3 environments of increasing fidelity (simulation, ground, microgravity), we acknowledge that open-sourcing deeper layers of the software for operation outside the ISS will entail more testing before final flight.

5.2. Does Crowdsourcing not Entail a Waste of Resources if only One Solution is Used?

The ethics of crowdsourcing is an important concern raised in literature (Ranade & Varshney, 2012) because crowds invest time and effort into submitting solutions to problems from which only a small subset (sometimes only one) is finally used. However, effort is not wasted if crowdsourcing programs are additionally used for educational purposes, as with ZR. All participants educationally gain from the experience of solving real-world problems. Introducing collaboration into the framework further reduces wasted effort because it entails combining many good solutions into an integrated one. Therefore, the concept of collaborative crowdsourcing and education for cluster flight algorithm development mitigates one of the chief concerns associated with standalone crowdsourcing.

The same logic applies to explain why effort be put into making the crowdsourcer interface, programming games and managing the tournaments for those problems where the same effort could be spent programming the algorithms themselves. Once the capital cost of setting up the ZR infrastructure within a cluster flight laboratory (e.g. MIT SSL) is accomplished, both crowdsourcing and educational value is gained at the maintenance cost of one, and moreover, at a cheaper maintenance cost than hiring traditional crowdsourcing companies such as TopCoder Inc.

6. CONCLUSION

Crowdsourcing and STEM education goals may be conflicting. In crowdsourcing, one cares only about the very best of solutions, i.e. the rightmost tail of the histogram distribution of performances in any competition (Figure 5, Figure 6). The purpose of sourcing solutions from dozens, hundreds or thousands of people is to identify the outliers that are most novel and high performing. In CS-STEM on the other hand, one cares to get maximum number of students involved and educated i.e. shift the average of the histogram distribution for any competition toward the right or raise the average score (vertical lines in Figure 5, Figure 6). The ZR program has proven that it is successfully able to achieve both simultaneously, apart from efficient and robust hardware test runs as well as positive user reviews of satisfaction and STEM inclination.

As a concluding note, we would like to stress on the importance of iterative evaluation in the development of any such program. The 2011 surveys combined with performance trends and participation statistics were invaluable in devising modifications to the program to make it more effective in the coming years. The scientific and education community are equal stakeholders in the process and hence pre-program input and post-program feedback from both is vital.

ACKNOWLEDGMENTS

The authors are grateful to Alvar Saenz-Otero, Jacob G. Katz, Sonny Thai and Prof. David W. Miller at MIT for their indispensible support during the 2011 tournament, whose results have been presented in this paper. This work was performed primarily under contract “NNH11CC26C: InSPIRE” with the Defense Advanced Research Projects Agency (DARPA).

The subcontractors to MIT are Aurora Flight Sciences and TopCoder. The SPHERES program is run by the NASA SPHERES team at Ames Research Center. The original concept for Zero Robotics was conceived in cooperation with astronaut Gregory Chamitoff.

REFERENCES

Allner, M., McKay, C., Coe, L., Rask, J., Paradise, J., & Wynne, J. J. (2010). NASA's explorer school and spaceward bound programs: Insights into two education programs designed to heighten public support for space science initiatives. *Acta Astronautica*, *66*, 1280–1284. doi:10.1016/j.actaastro.2009.09.019

Atkinson, R., & Mayo, M. (2010). Refueling the US innovation economy: Fresh approaches to science, technology, engineering and mathematics (STEM) education. *The Information Technology & Innovation Foundation, Forthcoming*. Retrieved from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1722822

Babbie, E. (2010). *The science of social research* (12th ed.). Stamford, CT: Wadsworth Cengage Learning.

Boudreau, K. J., & Lakhani, K. R. (2010). The confederacy of software production: Field experimental evidence on heterogeneous developers, tastes for institutions and effort. In *Proceedings of the NBER 50th Anniversary Conference on the Rate and Direction of Inventive Activity*.

Bradley, J. C., Lancashire, R. J., Lang, A. S., & Williams, A. J. (2009). The spectral game: Leveraging Open Data and crowdsourcing for education. *Journal of Cheminformatics*, *1*(9). doi:doi:10.1186/1758-2946-1-9

Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., & Beenen, M. (2010). Predicting protein structures with a multiplayer online game. *Nature*, *466*(7307), 756–760. doi:10.1038/nature09304

DARPA. (2012, August 29). *Synchronized tumbling: How to catch a retired satellite*. Defense Advanced Research Projects Agency. Retrieved December 14, 2012, from http://www.darpa.mil/NewsEvents/Releases/2012/08/29.aspx

Folta, D., Newman, L., & Gardner, T. (1996). Foundations of formation flying for mission to planet earth and new millennium. In *AIAA/AAS Astrodynamics Specialists Conference*. Retrieved from http://www.aric.or.kr/treatise/journal/content.asp?idx=25499

Goodman, C. (2008). Engineering ingenuity at iGEM. *Nature Chemical Biology*, *4*(13). doi:doi:10.1038/nchembio0108-13

Howe, J. (2006, June). The rise of crowdsourcing. *Wired*.

Ishikawa, S. T., & Gulick, V. C. (2012). Clickworkers interactive: Towards a robust crowdsourcing tool for collecting scientific data. In *Proceedings of the 43rd Lunar and Planetary Science Conference*, The Woodlands, TX.

Kim, J., Allen, J. P., & Lee, E. (2008). Alternate reality gaming. *Communications of the ACM*, *51*(2), 36–42. doi:10.1145/1314215.1314222

Klofper, E. (2008). *Augmented learning: Research and design of mobile educational games*. Cambridge, MA: MIT Press.

Laubacher, R., Olson, G., & Malone, T. (2011). The climate CoLab: Large scale model-based collaborative planning. In *Proceedings of the International Conference on Collaboration Technologies and Systems (CTS)*.

McGonigal, J. (2011). *Reality is broken: Why games make us better and how they can change the world*. London, UK: Penguin Books.

Melchoir, A., Cohen, F., Cutter, T., & Leavitt, T. (2005). *More than robots: An evaluation of the FIRST robotics competition participant and institutional impacts*. Waltham, MA: Heller School for Social Policy and Management, Brandeis University.

Nag, S., Heffan, I., Saenz-Otero, A., & Lydon, M. (2012). SPHERES Zero Robotics software development: Lessons on crowdsourcing and collaborative competition. In *Aerospace Conference, 2012 IEEE* (pp. 1–17). Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6187452

Nag, S., Katz, J. G., & Saenz-Otero, A. (2013). Collaborative gaming and competition for CS-STEM education using SPHERES Zero Robotics. *Acta Astronautica*, *83*(0), 145–174. doi:10.1016/j.actaastro.2012.09.006

Nag, S., & Summerer, L. (2013). Behaviour based, autonomous and distributed scatter manoeuvres for satellite swarms. *Acta Astronautica*, *82*(1), 95–109. doi:10.1016/j.actaastro.2012.04.030

Nag. (2012, June). *Collaborative competition for crowdsourcing spaceflight software and STEM education using SPHERES zero robotics* (Dual S.M.). Cambridge, MA: Massachusetts Institute of Technology.

NASA Strategic Plan 2011. (2012, April 24). Retrieved from http://www.nasa.gov/pdf/516579main_NASA2011StrategicPlan.pdf

Nolet, Kong, & Miller. (2005). Design of an algorithm for autonomous docking with a freely tumbling target. In *Proceedings of the SPIE Defense and Security Symposium 2005,* Orlando, FA (Vol. 5799–16).

O'Neill, M. G., Yue, H., Nag, S., Grogan, P., & De Weck, O. (2010). Comparing and optimizing the DARPA system F6 program value-centric design methodologies. In *Proceedings of the AIAA Space Conference*, Anaheim, California.

Pickard, G., & Pentland, A. (2011). Time-critical social mobilization. *Science*, 334.

Ranade & Varshney. (2012). To crowdsource or not to crowdsource? In *Proceedings of the Collective Intelligence Conference*, Cambridge, MA.

Resnick, M. (1998). Technologies for lifelong kindergarten. *Educational Technology Research and Development*, *46*(4), 43–55. doi:10.1007/BF02299672

Saenz-Otero, A. (2005, June). *Design principles for the development of space technology maturation laboratories aboard the international space station* (PhD). Cambridge, MA: Massachusetts Institute of Technology.

Team, M. I. T. Z. R. (n.d.-a). Zero robotics 2011 game manual. *Zero Robotics*. Retrieved April 28, 2012, from http://www.zerorobotics.org/web/zero-robotics/tournament-details?tournamentId=1

Team, M. I. T. Z. R. (n.d.-b). Zero robotics autonomous space capture challenge webpage. *Zero Robotics*. Retrieved April 27, 2012, from http://www.zerorobotics.org/web/zero-robotics/tournament-details?tournamentId=6

Trilling, B. (2010). From libraries to learning "libratories" : The new ABC's of 21st-Century school libraries. *School Library Monthly*, *27*(1), 43–46.

van 't Woud, J. S. S., Sandberg, J. A. C., & Wielinga, B. J. (2011). The Mars crowdsourcing experiment. In *Proceedings of the 16 th International Conference on Computer Games*, Wolverhampton, UK: IEEE Xplore.

Yee, N. (n.d.). The psychology of MMORPGs: Emotional investment, motivations, relationship formation, and problematic usage. In *Avatars at work and play: Collaboration and interaction in shared virtual environments* (Vol. *34*, pp. 187–207). London, UK: Springer-Verlag. doi:10.1007/1-4020-3898-4_9

ENDNOTES

1 Analysis described in this paragraph has been conducted by Jacob G. Katz, PhD candidate at MIT, and the conclusions drawn presented at an internal MIT seminar.

CALL FOR ARTICLES

International Journal of Space Technology Management and Innovation

An official publication of the Information Resources Management Association

The Editor-in-Chief of the *International Journal of Space Technology Management and Innovation* (IJSTMI) would like to invite you to consider submitting a manuscript for inclusion in this scholarly journal.

ISSN 2155-6350
eISSN 2155-6369
Published semi-annually

MISSION:

The mission of the *International Journal of Space Technology Management and Innovation* (IJSTMI) is to provide a forum to discuss the benefits of space technology for society, national economies, and industries and to investigate industrialization (commercialization) of space technologies, resulting in the development of new markets and applications (i.e. navigation, space tourism, solar power satellites etc.). This journal addresses space agencies management programs, space economics and technology innovation developments, and the impact of new space programs on employment and economic growth (i.e. direct benefits, etc.). IJSTMI examines the impact (i.e. space budgets reductions, etc.) of the global financial crisis on the space industry and space agencies' space exploration programs.

COVERAGE:

Topics to be discussed in this journal include (but are not limited to) the following:

- Institutional management and innovation
  - IPR rights
  - Management initiatives
  - Public private partnerships, mergers, and acquisitions
  - Small medium enterprise (SME) programs
  - Space agencies industrial applications programs
  - Space agencies industrial policies
- Space economics and management
  - Employment
  - Space budgets
  - Technology innovation and its contribution to economic growth
  - Trade export
- Space industry market trends
  - Earth observation
  - Interplanetary space exploration
  - Navigation
  - Telecommunications
- Space markets and applications
  - Medical devices and new drug development
  - Osteoporosis therapies and drugs
  - Solar energy satellites
  - Space debris
  - Space food
  - Space ports
  - Space tourism
  - Sub-orbital transportation vehicles

All submissions should be emailed to:
Dr. Stella Tkatchova, Editor-in-Chief
tkatch@yahoo.com

Ideas for special theme issues may be submitted to the Editor-in-Chief.