Repurposing Game Play Mechanics as a Technique for Designing Game-Based Virtual Worlds

Walt Scacchi
University of California, Irvine

1. Overview
2. Related Research on the What and How of Repurposing Game Mechanics
3. Case Studies in Repurposing Game Mechanics
   3.1 Case 1 – Producing functionally similar games
   3.2 Case 2 – modding an existing game via new game play levels, characters, and play objectives
   3.3 Case 3 – Replacing multi-character dialogs and adding Rashomon-style role-play
   3.4 Case 4 – Recognizing resource allocation challenges with uncertainty in problem domain
   3.5 Case 5 – Choosing meta-problem solving domains for game development, extension, and play
4. Comparative case analysis: The emerging technique of repurposing game play mechanics for game design
5. Conclusions
6. Acknowledgements
7. References

1. Overview

This paper examines the concept and practice of repurposing game play mechanics as a game design technique. It begins in the next section with a review of related research that informs this view of repurposing as a game design technique that may be adopted and practiced by new game developers or game players who want to engage in end-user game software development. Next, the paper describes five game design case studies where different forms of repurposing were employed. These case studies include identification of the example games and play mechanics that served to inform the game designs that follow from each. Each case study also categorizes the kinds of repurposing methods that were employed in reusing and adapting the source game mechanics into those employed in the new game design. Finally, the paper includes a simple comparative analysis and summary of these repurposing methods, as a basis for generalizing the method as a reusable and adaptive game design technique.

2. Related Research on the *What* and *How* of Repurposing Game Mechanics

Designing computer games or game-based virtual worlds is a challenging problem [Bar04]. Such a problem has the complexity of both software systems design, and of designing fun interactive play experiences [FSH06, How08, Sch08]. For the software engineering specialist new to game development, clean-sheet game design can be daunting challenge. So developers are wise to first start by examining existing games and game play experiences as a basis for designing future games. Such an approach relies on some form of informal domain analysis [PrA91] to identify recognizable game features and capabilities, and also game objects, functional operations, game play rules, and relationships between objects, operations, rules, features and capabilities, as game play mechanics [Sic08].

A formal domain analysis might have the potential to give rise to a persistent set of reusable game components, play mechanics, or eventually a meta-design theory for games [cf. KRF11] that could enable the (semi-) automated generation of playable computer games [Sca12b]. However, we are not yet at that point. Instead we first seek a simpler contribution that is to explore how observation and informal analysis of existing games can lead to identification and conceptual reuse of game play mechanics [cf. FrK06]. Such an approach is already available in part through game modding tools and techniques that allow end-users (game players or aspiring game developers) to learn how to modify an existing game product [ElS06] to create functionally equivalent game variants, functionally similar game versions, or total conversions of the game into an entirely new game [Sca12a]. Such forms of game modding are a widespread form of end-user software engineering [cf. KAB11] that rely on user interface customization, use of domain-specific scripting languages, or game-specific software development kits [Sca12a]. Modding as an approach to game development can however also be extended across games through informal *mashups* [cf. CDP11, NeC06] that compose game play mechanics from multiple games, or from other online applications, into a new game conversion.

In this paper, we utilize a collection of case studies of game-based virtual worlds that we have produced across a number of game development projects. Our goal is to describe our experience in recognizing and reusing game play mechanics found in existing games (including those we have developed) as the basis for articulating a new approach to game design. Five cases are presented, where each identifies the game (or games) that preceded or precipitated the design of a new game. In many of these cases, explicit reuse of game play mechanics was not our goal, but in review, it appeared, so we use this paper to highlight our experience and findings. We then provide a comparative case analysis to identify an emerging set of generalizations that arise from our experience. Such generalizations are intended to serve as a starting point for a more fully articulated approach to design of computer games or game-based virtual worlds through either opportunistic repurposing or systematic reuse of game play mechanisms.

3. Case Studies in Repurposing Game Play Mechanics

In this section five case studies are presented that identify and describe different techniques or strategies for repurposing game play mechanics. The case studies arise from observation of practices that gave rise to different computer game development projects produced or directed by the author. Each is described in turn.

3.1 Case 1 – Producing functionally similar games

One of the first game R&D projects focused on producing games for young students in grades K-6 that focused on informal life science education [SNA08]. These games were envisioned as part of a science learning game environment called *DinoQuest Online* for free-to-play, Web-based deployment. It was designed to complement and interoperate with an on-site interactive, hands-on science exhibit called *DinoQuest,* located at the Discovery Science Center (a family-friendly science museum) in Santa Ana, CA [SNA08]. Our starting point was to develop games for learning life sciences appropriate for primary/elementary school students, focusing on dinosaurs as prehistoric living creatures. Such creatures like other living animals or humans, all have basic body systems (e.g., skeletal, nervous, digestive, reproductive, cardio-pulmonary), and understanding the what these system are, what they do and how they do it, and interrelationships of these body systems is fundamental, as is understanding the ecologies in which creatures thrive, survive, or die. The ecologies include relationships among creatures of different species, such as relationships indicating prey-predator, food chains, and "circle of life," all of which are foundational to contemporary life science in the worlds in which people and animals live. So how might we design a game or play mechanic that would highlight these ecological relationships? Our inspiration came from the core play mechanics in the games *Tetris* and *Dr. Mario*, displayed in Figure 1.

Tetris and Dr. Mario are tile-matching puzzle games where sets of geometric colored tile patterns (sometimes called, "tetrominoes" in homage to dominoes) arrive/fall into the game play space (the playing field), and must be matched to fit in with other tile patterns already in place (Tetris), or form a repeated pattern like three-in-a row (Dr. Mario). The arriving tile patterns may be rotated or slid across the playing field in order to align and match the in-place patterns. Game play is motivated by time, as the patterns fall into and progress across the playing field at a certain pace, so that rapid recognition matching patterns and geometric spatial reasoning are helpful cognitive skills for effective game play.

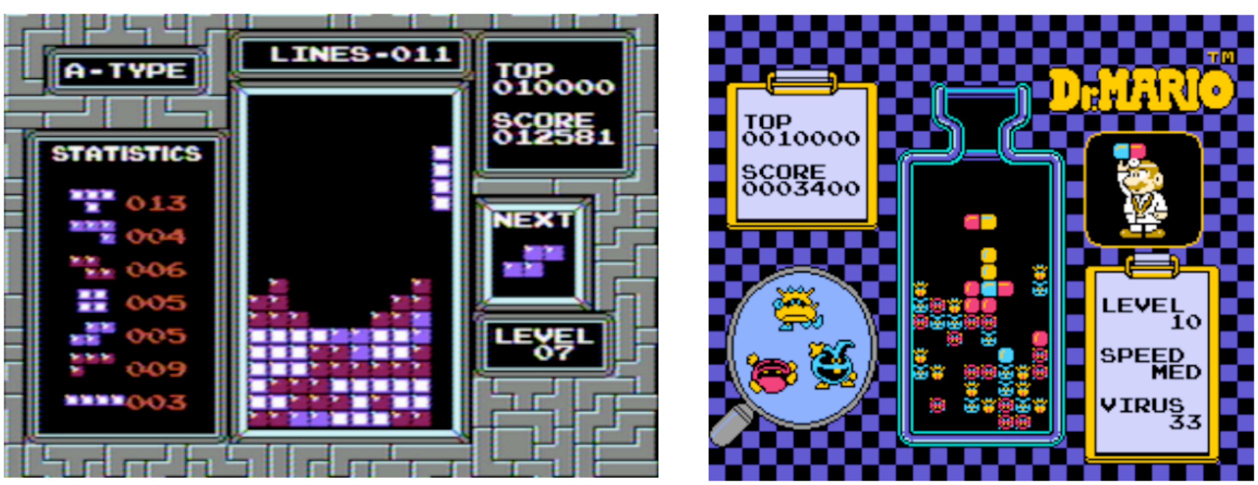

Figure 1. Screenshot of *Tetris* (left) and *Dr. Mario* (right) games (source images: Wikipedia.com)

This game encourages and relies on such cognitive skills as game play increases in complexity and repetition. As the few people involved in the game project team were all experienced in Tetris game play, we quickly began to see how the basic Tetris play mechanic could be repurposed by substituting a carnivore, herbivore, plant, and mulch as the four (color) kinds of tiles. The resulting prey-predator food chain game is seen in Figure 2. However, we elected to rotate the Tetris/food chain game board 90 degrees, so prey-predator tetronimoes emerge and flow left to right, rather than top to bottom.

When carnivores (predators) are aligned with herbivores (prey), they form a simple food chain, allowing the survival of the carnivore and the consumption of the prey. Similarly, when herbivores align with plant, the herbivore survives, as the plant is consumed. When plants align with mulch they thrive, and the mulch is consumed. When one or more carnivores are aligned with plants or mulch, they will eventually die and turn into mulch. So our game enacts a prey-predator food chain relationship where mulch contributes to the plants, plants to herbivores, herbivores to carnivores, and carnivores to mulch as the circle of life (or food web), and game play enables players to score points as they recognize and re-enact the prey-predator and food chain relationships through their game play moves. As such, we were fortunate to recognize that Tetris and Dr. Mario could serve as functionally similar games and game play, whose game play mechanics could be abstracted, reused, and tailored as part of our game design.

Figure 2. A screenshot of the prey-predator food chain games in *DinoQuest Online* [SNA08] that repurpose familiar *Tetris* play mechanics, from left-to-right instead of top-to-bottom.

3.2 Case 2 – Modding an existing game via new game play levels, characters, and play objectives

Another project arose in response to our presentation of DinoQuest Online at the Game Developers Conference in 2007. We had a chance encounter with people from Intel Corporation who inquired of us if or how games might be employed to provide an interactive game-based training environment targeted to adult technicians as the player/trainee. These technicians operate, diagnose, and service semiconductor fabrication operation facilities in advanced manufacturing systems factories [Sca10]. We took up this challenge. At the time, the then new first-person shooter game, *Unreal Tournament 2007*, was being demonstrated at the GDC (prior to its release later that year), as a new version succeeding *Unreal Tournament 2004*. Both UT2K4 and UT2K7 were among the early generation of FPS games that could be modded using a game editor (UnrealEd), which we were already familiar with (cf. [Sca02]). Figure 3 displays UnrealEd.

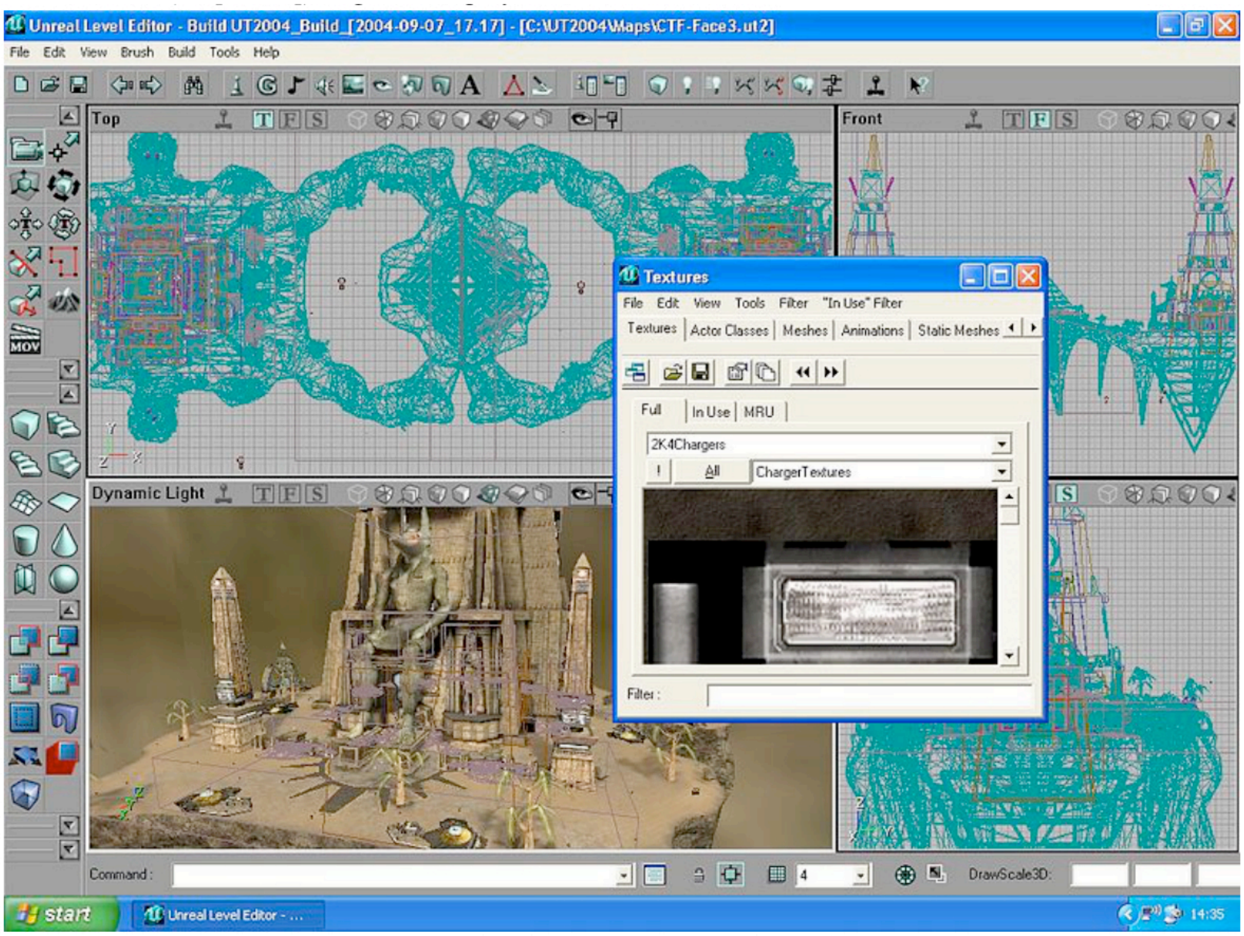

Figure 3. A screenshot of UnrealEd software development (game modding) kit operating with *Unreal Tournament 2004* game environment (source image: Wikipedia.com)

Our challenge entailed understanding what a semiconductor fabrication facility looks like, what kinds of fabrication machines are present, their basic operation, along with technician procedures and practices for operating, diagnosing, and servicing the machines, including dealing with material spills that "contaminate" a fabrication clean room. Much of this information could be found in information acquired through copious Web searches, along with visits to a local semiconductor and nanotechnology systems fabrication facility located at UCI (the UCI Integrated Nanosystems Research Facility). We then constructed object, character, and fabrication machine models and associated textures (skins) using tools like Autodesk 3DStudio and Photoshop. Visualizing and simulating materials spills were a priority interest of our sponsors, as training technicians in diagnosing and cleaning-up such spills was a critical challenge of interest to our sponsors. Similarly, we recognized that fabrication technicians were attired in full-body suits ("bunny suits"), and modeling and simulating how technicians put on such suits ("gowning"), became a separate follow-on game-based training simulation project with our sponsor. So once we acquired knowledge about semiconductor fabrication operations and diagnostic servicing, we were able to rapidly design and prototype a new game called *FabLab* through a mod of UT2K4. A sample of images is shown in Figure 4.

Among the noteworthy details of the modding-based game development effort was that it was performed by a single undergraduate student researcher (directed by the author), and initially released within three months of the start of the project. Similarly, feedback from the project sponsor included additional challenge tasks to develop other game prototypes for different manufacturing and business operations processes. Nonetheless, the networked multi-player game play capabilities provided by UT2K4 and UT2K7 were inherently available within the FabLab game mod, and thus could be reused as-is at no additional software development effort.

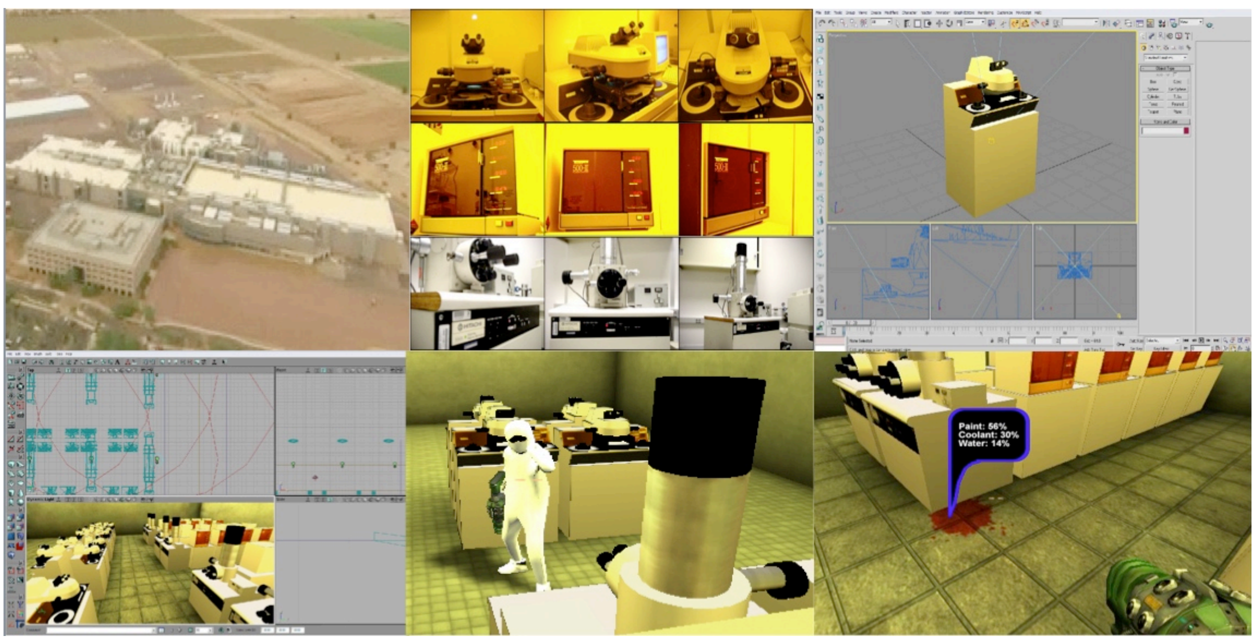

Figure 4. Images for the development and use of the FabLab game mod [Sca10], developed with UnrealEd and *Unreal Tournament 2004*.

3.3 Case 3 – Replacing multi-character dialogs and adding Rashomon-style role-play

Another of the game development challenges posed by Intel was for our group to rapidly develop a new game from scratch using low-cost tools and a total development schedule including global deployment and end-user play evaluation in less than two months. This would be a training game, rather than a game-base operations simulator like FabLab, and it would need to be deployable on low-cost personal computers or mobile devices (e.g., smartphones). Part of the sponsor's goal was for us to articulate and document the game design and development process in ways that might be reused by corporate eLearning specialists who develop on-the-job training materials for newly hired business process personnel. Also, our development budget was limited to be comparable to that available for these specialists to develop conventional training materials (e.g., PowerPoint presentation slide decks). The selected business process was "order management" which focuses on the activities of business specialists who handle in-coming customer orders for products that are manufactured either "built to external order" or "built to fill internal inventory." The result of this internal-only game development and deployment (due to non-disclosure agreement at the time), was a simple, single-player 2D role-playing game called *Customer Business Analyst* (CBA) that covered 3-4 weeks of simulated order management problem-solving tasks in a cubicle office environment, from simple to problematic (with nagging customers making phone calls to complain, communicating with customers and manufacturing operations via email, accessing in-house corporate enterprise resource planning systems (the "Hub"), and unreliable order status information). These features are highlighted in Figure 5—the two figures on the left provide "birds eye view" of the office space, while the two on the right provide an analyst's "desktop view" of their order management work activities.

After delivery and retirement of the CBA game, about two years later, we received a new challenge from a large California statewide government agency responsible for deploying State-mandated annual training for State employees in recognizing, avoiding, or reporting sexual harassment in public workplaces. Once again, our game prototyping budget and demonstration schedule was extremely limited, so this called for a game design and demonstration that could be done "quick and dirty."

Our choice was to repurpose the role-playing CBA game through substitution of in-game characters dialog while also providing support for the *Rashomon* effect [Hei88], but through in-game characters and non-player characters. The substitution of character dialog and situations as the basis for a new/adapted story is a common practice found literature, cinema, and game-based machinima. For example, the Shakespearian story of *The Tempest* was repurposed into the cinematic science-fiction story of *Forbidden Planet*, while the popular *Red vs. Blue* machinima series represent the substitution of human voices over in-game character dialogs/taunts from various *Halo* FPS games. The Rashomon effect refers to the situation where multiple actors in a common situation or interaction provide differing, sometimes conflicting, interpretations of what they perceive (cf. the allegory of the blind men feeling the elephant). This exploration of characters acting in different roles encountering a common situation also appears in interactive game-based worlds, where players can switch between characters to re-interpret and re-experience a recent game play situation, as a play mechanic, appears in games such as *The Residents: Bad Day at the Midway* released in 1996 [Res14]. Our new game play experience was thus designed to demonstrate how to provide scenario-based character encounters that feature innocent, ambiguous, or abusive verbal interactions different/competing perspectives of in-game characters regarding whether or not they perceived multi-character dialog or interactions as engendering sexual harassment activities. A screenshot of the repurposed CBA-into-sexual harassment game is shown in Figure 6.

To be clear, this game is not envisioned as a game to play to win, but instead to provide a sufficiently engaging experience so that State employees would play through dozens of different possible sexual harassment scenarios to fulfill the State-mandated training requirement. The State's harassment liability insurance provider eventually argued against the development and deployment of the game in part because it was "a game" though the Rashomon effect was highlighted as very original and likely well-suited for sexual harassment training (though a few other reviewers objected to its value for this subject, as they preferred clear and unambiguous interpretations of what is or is not harassment).

Figure 5. Four screenshots from the *Customer Business Analyst* game for training order management business analysts.

3.4 Case 4 – Recognizing resource allocation challenges with uncertainty in problem domain

A different repurposing opportunity emerged during a research project engagement with a different corporate sponsor, and then later with a government sponsor. The challenge was to design and prototype a multi-player virtual world where different users could engage in "mission planning" problem-solving tasks found in military and civilian command and control systems. Mission planning refers to the activity of identifying and allocating available resources to remote parties who need the resources to complete their assigned mission (e.g., search and rescue of victims in a remote location following a natural disaster). Mission planning is difficult in situations that are dynamic and constantly changing, as resource availability is characterized by uncertainty, as is information about what is the most pressing conditions or mission variables to manage at a given time.

Mission planning is a kind of multi-party resource allocation with uncertainty problem in classical game theory. Few people are particularly adept at mission planning, and the consequences of allocation decisions may be very high. Subsequently, there is widespread R&D effort focused on identifying "optimal" solutions to such allocation decisions, but little in the

Figure 6. A screenshot from the repurposed CBA game now including substituted in-game character dialogs (here highlighting a scenario where one character uses possibly ambiguous terms of endearment or sexual innuendo, drawn from a current sexual harassment training scenario).

way of training simulators that situate new players in planning contexts where they must make resource allocation decisions in the presence of private/shared information that is dynamic and uncertain.

We developed and demonstrated a game-based virtual world for conducting experiments in multi-player game-based mission planning we called *DECENT* [SBN12]. Our world building efforts were based on the user-extensible open source software (OSS) virtual world server and *Second Life* work-alike platform, *OpenSim* (http://www.opensim.org). OpenSim operates on a scalable (MMO) and federated wide-area network environment linking multiple interoperable virtual worlds via the *Hypergrid* [Lop11, Sca12b]. Once again, as in Case 2 for the FabLab game, we start by investigating and analyzing information about mission planning activities and facilities, as the basis for modeling and simulating the world of mission planning problem-solving. Figure 7 displays a collage showing a large physical facility for mission planning; adjacent to scenes of the virtual world we modeled to simulate it (in-world agents not shown in these scenes, but they can be seen elsewhere [SBN12]). The virtual world for experimental studies for mission planning in a decentralized command and control setting is called, DECENT.

Figure 7. Scenes from a physical facility for mission planning, and mirror virtual world for mission planning in DECENT [SBN12].

In further brainstorming with our project sponsors based on early demonstrations of the DECENT world, our sponsors asked what kind of mission planning game play would be possible in such a virtual world. OpenSim, much like Second Life, is not a game development environment, but is an open-ended, user extensible virtual world environment. So it lacks game play primitives, score keeping, and domain-specific scripting facilities designed for game play. However, it is open to user extension. After considering and debating a number of alternatives, we decided that the popular seven-card poker tournament game, *Texas Hold'em*, would be our candidate for supporting mission planning game play. Accordingly, we extended our DECENT world to support this game, but utilizing the in-world user display screens to show both each user's card holdings, along with the table cards on a large virtual wall display visible to all. We also repurposed the poker cards to now be denominated in four types of military services (land, sea, air, cyber), with card point values expressing a corresponding quantity of military objects (e.g., tanks, ships, planes, staff). We also added meta-game non-player cards drawn at random that serve to modify the value of cards held/shown, as another way to introduce uncertainty into resource allocation game play. In multi-player games, only the winning hand gets its resource allocation effected, based on the cards in the final round, while losing hands do not get allocations. Otherwise, game play follows the rules and procedures of the Texas Hold'em card game. Figure 8 displays a scene from in-world game play, and a photograph of multiple students playing the game.

Finally, we have taken the multi-player virtual world of DECENT and have also repurposed its characters and objects for non-military applications. For instance, we have repurposed from the virtual world for mission planning to adult teachers and children students in an elementary classroom setting with desks, tabletop computers, and large-screen projector to support new experimental studies of classroom-based science and health-care education.

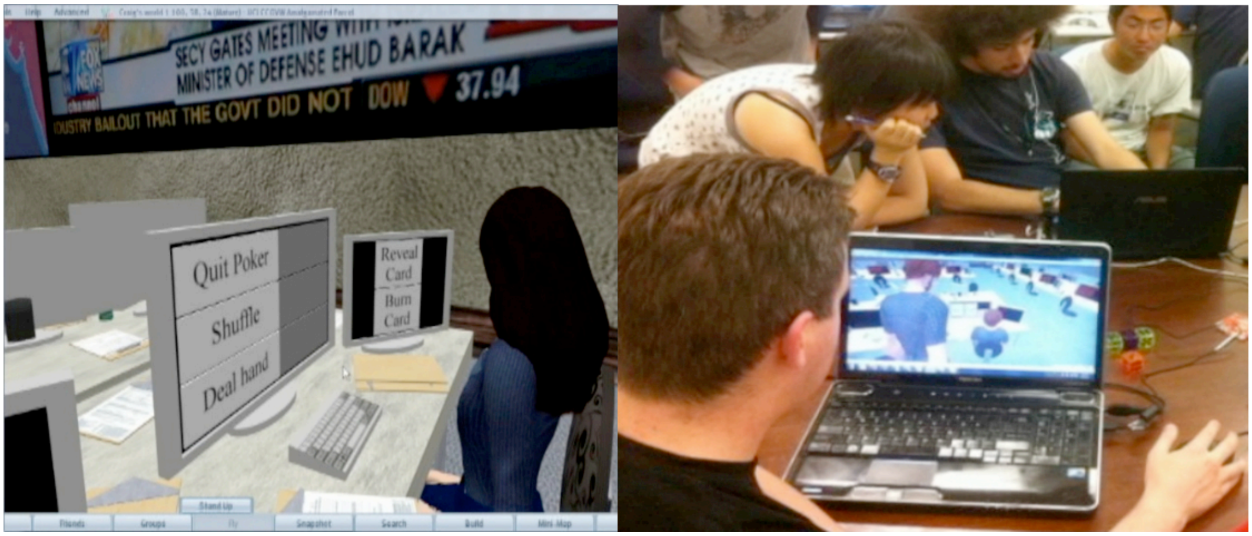

Figure 8. Scenes of game play in the DECENT virtual world for mission planning [SBN12].

3.5 Case 5 – Choosing meta-problem solving domains for game development, extension, and play

The last case study examines an ongoing research project that focuses on developing a reusable framework for developing games about contemporary science mission research projects. Mission oriented science projects are generally large-scale quests involving multiple characters in different roles (cf. a role-playing game) pursing both local and shared goals that include accomplishing a previously unattained destination [cf. How08].

Our interest is on mission quests that focus on new research studies that seek to travel to difficult sites, observe/sense, and retrieve objects/material results of great scientific interest. Recent NASA/space science projects include space-borne robotic travel, landing, instrumented sensing, and resource harvesting from nearby planets, moons, asteroids, or comets. Earth-centered satellite observation missions are similar, as are exploratory voyages to the bottom of the Earth's oceanic abyssal plane or deep-sea destinations. They are also missions to develop, launch, navigation, and capture/delivery of Nano-scale objects of interest within living systems, like the human body. We also are examining other mission-like quest games, such as *Buzz Aldrin's Race Into Space* (which recreates the competition between the US and Russia to land a man on the moon), along with cinematic science-fiction stories like *Fantastic Voyage* (a microscopic scale vehicle navigation and treatment of a human body) for mechanics for repurposing. A growing number of scientific missions like these have broader implications for both anticipating technology commercialization efforts, as well as providing awareness and public interest for future science students. Subsequently, this effort leverages and repurposes what we had accomplished with our mission planning world from Case 4, as well as what we have learned from our other repurposing case study efforts.

What is therefore new in this case is our focus on meta-problem solving domains: multiple independent domains of scientific exploration where large-scale missions are focal, as are modeling and simulation of diverse behavioral objects and multiple character roles that collectively articulate mission processes, procedures, and puzzles. In this case, the challenge for designing a reusable science game development framework through repurposing to rapidly develop new, scalable, and potentially interoperable games that are associated with scientific research missions in different science domains. Figure 9 displays screenshots of game play mechanics we are currently investigating for space science missions that focus on travel to, capture, and retrieval of near-earth objects or small planetary objects, such as near-earth asteroids [HLR12, Sca12b, Wal13].

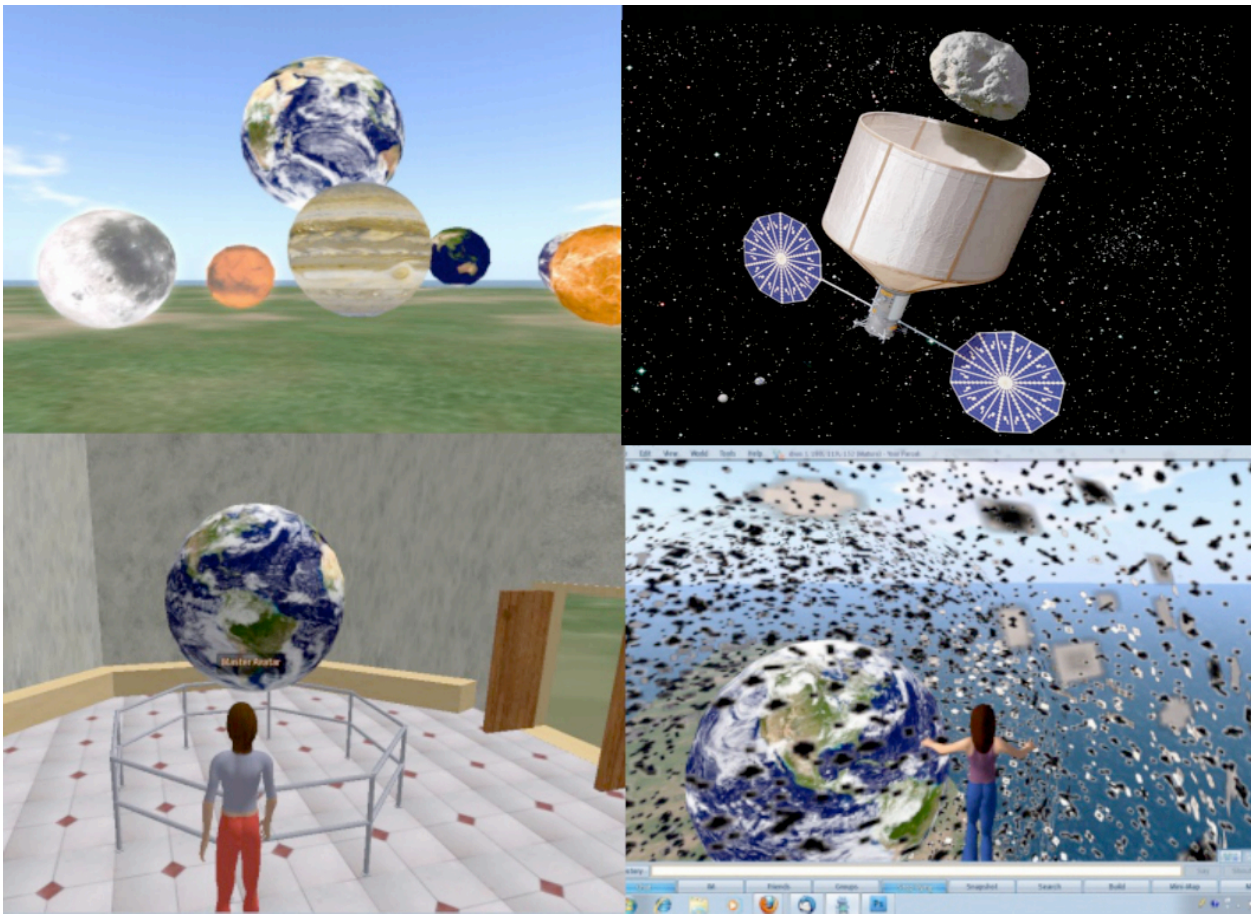

Figure 9. Example game objects and affordances for near-earth space exploration games for prototyping future space science missions and educational outreach [cf. Sca12b].

4. Comparative case analysis: The emerging technique of repurposing game play mechanics for game design

In this paper, we presented five case studies of game-based virtual worlds that we have produced across a number of game development projects. Our goal has been to describe our experience in recognizing and reusing game play mechanics found in existing games, including those we have developed, as the basis for articulating a new approach to game design. Along the way we identified five ways for repurposing game play mechanics: via (a) appropriating play mechanics from functional similar games; (b) modding game play levels, characters, and weapons through a game-specific software development kit; (c) substituting in-game character/non-player character dialogs along with adopting multi-player role-play scenarios; (d) employing play mechanisms for resource allocation with uncertainty that re-enact classic game theory problems; and (e) identifying game design patterns (quests, multiplayer role-playing) that can be used to develop families of games across science research problem-solving domains. All of these games repurposing mechanics are reusable, though the five cases presented find them used idiosyncratically. However, in presenting these cases, it becomes clear that it is possible to systematically observe, identify, conceptually extract, generalize and then specialize/tailor game play mechanics that are provided capabilities [AlS13] in existing games.

While these five repurposing cases each denote a different type of repurposing, we would not say that a repurposing taxonomy or process meta-model would only include these five classes. In contrast, we hold that repurposing is an interesting and productive family of design heuristics that can inform game design in ways that can be readily mastered by software engineers who focus on software reuse tools and techniques [FrK06, Nei84, PrA91]. Identification, classification and taxonomization of repurposing heuristics would however appear as a promising line of research and practice in game design that is informed by modern software engineering theory and practice. It is also fair to say that repurposing is not an end-user software engineering practice, at least not at this time, as repurposing relies on insights and practices arising in analysis and modeling of games or game play mechanics as a problem domain [PrA91]. But determining which repurposing heuristics might be useful for end-users seeking to engage during game modding, also appears as a promising line of inquiry and practice. Thus existing games can serve as a rich base from which it is possible to analyze as an application design domain that utilizes diverse game play mechanics and associated objects, operations, and play experiences. Consequently, repurposing is a new and different method for identifying, analyzing, generalizing reusable game-based software system designs that are available to game designers familiar with existing games, or experienced in game development.

Last, it is also worth noting that repurposing is not merely copying and editing, or merely creating derivative variants of known games. Repurposing is not the same as modding, though modding practices can inform or afford the repurposing of game play mechanics [Sca12a], and as just noted, repurposing may inform game modding. Repurposing is targeted at the design level, rather than at the implementation level, of software. Reuse of source code (or open source software) is a different problem and opportunity compared to reuse of software system designs, or reuse of game play mechanics. Reusing game play mechanics or the design of play mechanisms, is desirable and non-infringing: it does not necessarily produce or give rise to derivative works which might infringe on the copyright of an established game property. Once again, repurposing of game play mechanics is an emerging technique for the design of computer games and game-based virtual worlds, as demonstrated in this paper.

5. Conclusions

This paper examines the concept and practice of repurposing game play mechanics as a game design technique. In addition to a review of related research, the paper briefly describes five game design case studies where different forms of repurposing were employed. These case studies include identification of the source game and play mechanics, as well as categorizing what kinds of repurposing methods were employed in reusing and adapting the source game mechanics into those employed in the new game design. The paper also includes a simple comparative analysis of these repurposing methods, as a basis for generalizing the method as a reusable and adaptive game design technique. Repurposing appears to be a promising new heuristic approach to identifying, rapidly designing, and making computer games or game-based virtual worlds in ways the employ informal domain analysis techniques and related methods from reusable software and end-user software engineering. It also suggests new lines of research and practice that leverage computer games and software engineer research and practice, which therefore merit further investigation.

6. Acknowledgements

Support for this research was provided by grants #0808783, #1041918, and #1256593 from the U.S. National Science Foundation. No review, approval, or endorsement is implied. Only the author of this chapter is responsible for statements and recommendations made herein.

7. References

[AlS13] Alspaugh, T.A., and Scacchi, W. (2013). Ongoing Software Development Without Classical Functional Requirements, *Proc. 21st. IEEE Intern. Conf. Requirements Engineering*, Rio de Janeiro, Brazil, 165-174.

[Bar04] Bartle, R.A. (2004). *Designing Virtual Worlds*, New Riders, Indianapolis, IA.

[CDP11] Cappiello, C., Daniel, F., Matera, M., Picozzi, M. and Weiss, M. (2011). Enabling End User Development through Mashups: Requirements, Abstractions and Innovation Toolkits, in *End-User Development*, Lecture Notes in Computer Science, Vol. 6654, 9-14.

[CND10] Cypher, A., Nichols, J. Dontcheva, M, and Lau, T. (2010). *No Code Required: Giving Users Tools to Transform the Web*, Morgan Kaufmann.

[ElS06] El-Nasr, M.S. and Smith, B.K. (2006). Learning Through Game Modding, *ACM Computers in Entertainment,* 4(1). Article 3B

[FrK06] Frakes, W.B. and Kang, K. (2006). Software Reuse Research: Status and Future, *IEEE Trans. Software Engineering*, 31(7), 529-536.

[FSH04] Fullerton, T., Swain, C., Hoffman, S. (2004). *Game Design Workshop: Designing, Prototyping and Playtesting Games*. CMP Books, February 2004.

[HLR12] Hasnain, Z., Lamb, C.A., and Ross, S.D. (2012). Capturing near-Earth asteroids around Earth. *Acta Astronautica*, 81(2), 523-531, December.

[Hei88] Heider, K.G. (1988). The Rashomon Effect: When Ethnographers Disagree, *American Anthropologist*, 90(1), 73-81. Also see *Rashomon* at http://en.wikipedia.org/wiki/Rashomon_(film)

[How08] Howard, J. (2008). *Quests: Design, Theory, and History in Games and Narratives*, A.K. Peters, Wellesley, MA.

[KAB11] Ko, A.J., Abraham, R., Beckwith, L., *et al.* (2011). The State of the Art in End-User Software Engineering, *ACM Computing Surveys*, 43(3), Article No. 21.

[KRF11] Koehne, B. Redmiles, D. and Fisher, G. (2011). Extending the Meta-Design Theory: Engaging Participants as Active Contributors in Virtual Worlds, in End-User Development, *Lecture Notes in Computer Science*, Vol. 6654, 264-269.

[Lop11] Lopes, C. (2011). Hypergrid: Architecture and Protocol for Virtual World Interoperability, *IEEE Internet Computing*, 15(5), 22-29.

[Nei84] Neighbors, J.M. (1984). The Draco Approach to Constructing Software from Reusable Components. *IEEE Transactions of Software Engineering,* SE-10(5), 564-574, September 1984.

[NeC06] Nelson, L. and Churchill, E., (2006). Repurposing: Techniques for Reuse and Integration of Interactive Systems, *Proc. 2006 IEEE Conf. Information Reuse and Integration*. 490-495.

[PrA91] Prieto-Diaz, R. and Arango, G. (1991). *Domain Analysis and Software Systems Modeling,* IEEE Computer Society, Los Alamitos, CA.

[Res14] Residents (2014). *Bad Day at the Midway*, Wikipedia.com, http://en.wikipedia.org/wiki/Bad_Day_on_the_Midway, accessed April 2014.

[Sca02] Scacchi, W. (2002). Understanding the Requirements for Open Source Software Development, IEE Proceedings-Software, 149(1), 24-39, February 2002. Revised and expanded version appears in K, Lyytinen, P. Loucopoulos, J. Mylopoulos, and W. Robinson (eds.), *Design Requirements Engineering: A Ten-Year Perspective,* LNBIP 14, Springer-Verlag, 467-494, 2009.

[Sca10] Scacchi, W. (2010). Game-Based Virtual Worlds as Decentralized Virtual Activity Systems, in W.S. Bainbridge (Ed.), *Online Worlds: Convergence of the Real and the Virtual*, Springer, New York, 225-236.

[Sca12a] Scacchi, W. (2012). Modding as an Approach to Extending Computer Game Systems, *Intern. J. Open Source Software and Processes*, 3(3), 36-47. Reprinted in S. Koch (Ed.), *Open Source Software Dynamics, Processes, and Applications*, 177-188, Information Science Reference, IGI Global, 2013.

[Sca12b] Scacchi, W. (Ed.), *The Future of Research in Computer Games and Virtual Worlds: NSF Workshop Report,* Technical Report ISR-12-8, Institute for Software Research, University of California, Irvine, August 2012.
http://www.isr.uci.edu/tech_reports/UCI-ISR-12-8.pdf

[SBN12] Scacchi, W., Brown, C., and Nies, K. (2012). Exploring the Potential of Computer Games for Decentralized Command and Control, *Proc. 17th. Intern. Command and Control Research and Technology Symposium* (ICCRTS), Paper104, Fairfax, VA, June 2012.

[SNA08] Scacchi, W., Nideffer, R. and Adams, J. (2008). A Collaborative Science Learning Game Environment for Informal Science Education: DinoQuest Online, in IFIP International

Federation for Information Processing, Volume 279; *New Frontiers for Entertainment Computing;* P. Ciancarini, R. Nakatsu, M. Rauterberg, M. Roccetti (Eds.); Boston: Springer, 71–82.

[Sch08] Schell, J. (2008). *The Art of Game Design: A Book of Lenses*, Morgan Kauffman.

[Sic08] Sicart, M. (2008). Defining Game Mechanics, *Game Studies*, 8(2), December.

[Wal13] Wall, M. (2013). Capturing an Asteroid: How NASA Could Do It, *Space.com*, 10 April 2013. http://www.space.com/20591-nasa-asteroid-capture-mission-feasibility.html