

Competitive Game Development: Software Engineering as a Team Sport*

Walt Scacchi

Institute for Software Research

and

Center for Computer Games and Virtual Worlds

University of California, Irvine

*Presented at the *2nd. Games and Software Engineering Workshop*,
34th. Intern. Conf. Software Engineering, Zurich, 9 June 2012

Overview

- The what, why, how and outcomes of game software development competitions
- Related efforts in competitive software development
 - software development competitions
 - software engineering education and research
- Observations, lessons learned, and conclusions

The *What* of game software development competitions

- Competition affords the opportunity for alternative interpretations of common game software requirements.
 - independent selection of game topic
- “Green field” game software development versus game modding [Scacchi 2011]
- *Goal*: present observational results from multi-round field studies of computer game software development competitions hosted at UC Irvine, starting in 2010.

Video game development club game demos

Video Game Development Club @UCI Log In Sign Up

VGDC UCI
Video Game Developers of UC Irvine, CA

THE BEST PLACE AT UC IRVINE TO LEVEL UP YOUR PORTFOLIO AND CAREER!

HOME ACTIVITY FORUMS GROUPS MEMBERS PHOTOS WIKI INFO **DEMOS** GAME COMPANIES

Demos

Angry Hand of God

The Angry Hand of God Trailer

Like Share More info



VGDC UCI VGDC UCI
Like 136

Follow @vgdcuci 16 followers

TAGS

3d aif artist blizzard business contest demo field trip first meeting game design game idea game industry leaders game industry professionals game jam GDC guest speaker help howto meeting mobile new room no meeting obsidian oc weekly opportunity pitch poll producer projects research rookies slides summer svn tips top this tour trivia night tutorial tutorials uci alum vgdc in the news video winner xna

The *Why* of game software development competitions

- There is growing interest in conducting and facilitating such competitions for reasons including:
 - starting up a local culture of game software development,
 - building entries into student resumes in preparation for job placement [cf. Scacchi 2002, 2004], as well as
 - having extracurricular fun outside of coursework that can utilize knowledge gained inside coursework [Hamilton 2011].
 - exercising user-led innovation using tool-kits [von Hippel, 2002, 2005, Franke and von Hippel 2003]
 - gaining SE experience in rapid prototyping, agile development, or accelerated time-compressed product development
- But do the participating developers learn software engineering, or what do they learn about SE practices, techniques, or tools?

Published: April 13, 2011 Updated: 7:59 a.m.

UCI students build video games in week

Tweet 15

Recommend

Sign Up to see what your friends recommend.

Email

Share

By IAN HAMILTON / THE ORANGE COUNTY REGISTER

After a week spent building a computer game from scratch, Justin Britch is sitting behind a laptop Monday night just five hours from an 11 p.m. deadline.

"We couldn't get our levels to work for a very long time," Britch tells Reza Ghassemi, president of UC Irvine's Video Game Development Club.



Jesus Quezada, a member of team Rainbow Dice Games works on coding a video game at UC Irvine on Monday.

CHRISTINE COTTER, FOR THE ORANGE COUNTY REGISTER

"For how long?" asks Ghassemi.

"Are they working now?" Britch, asks his teammates.

No, half the levels still don't work.

Britch, 19 and a resident of Mission Viejo, is participating in his second "Game Jam," a week-long marathon competition put on by the university's gaming club pitting teams of computer science, art and informatics majors working around the clock against one another to build computer games.

Though the games are simple, mostly two dimensional with flat maps and relatively simple multicolored animated characters, the third "Game Jam" competition held by the club is an opportunity for budding artists, game designers and programmers to add a finished product to their portfolio. And they do it in just a week while experiencing what it's like to work and collaborate on a project.

With video game giant **Blizzard Entertainment** nearby, UC Irvine is seeing an increase in the

Text:

Next Article »

TODAY'S DEAL

SEE IT

Tacos & Co

\$10 for \$20 Worth of Food and Drink at Tacos & Co

OCRegister.com



ADVERTISEMENT

Sprint

TRADE UP & SAVE \$100

For a limited time, bring us an iPhone from any carrier and get at least \$100 off an iPhone 4S.

LEARN MORE

Restrictions apply.



iPhone

More from Technology

- E3 shows companies looking beyond games
- TheStreet: iPad Mini, iOS 6, MacBook Pro
- O.C.-based iPad restaurant touted as pioneer

Recommended for You

The *How* of game software development competitions

- Game software engineering process: issues, constraints, and caveats
 - Requirements
 - Reuse
 - Design
 - Code sprint
 - Testing and post mortem
- Collaborative game software development tools
- Balancing game SE team competition

Game software requirements

- primary emphasis on creating and satisfying *non-functional requirements* for the game as product,
 - examples:
 - *game must be playable in one week!*
 - provide online video (YouTube) of game demo
 - provide external testers all installable game run-time resources
- game software functional requirements are tacit and undocumented.
- game developers as *end-users* [Scacchi 2010]
 - elicitation of functional requirements can often be much less complicated than compared to situations where developers ("us") and users ("them") are distinct groups.

Game software reuse

- *What gets reused?*
 - game development components (e.g., closed/open source software game engines) and libraries
 - game play mechanics, design of play sequences, and play experience
 - game content assets, but not misappropriated media assets subject to copyright.
 - knowledge and experience from earlier game development competitions
- *Modding as a reuse strategy*
 - *modifying* existing games via extension mechanisms like domain-specific scripting (modding) languages which reuse, modify, or create new game play mechanics and play experiences [Scacchi 2011]
- Game development tool frameworks (discussed later)

Game software design

- Game design principles [Fullerton et al 2004, Rogers 2010, Schell 2008] are different from those for software design.
 - Game design focuses attention on:
 - how to address non-functional requirements for game characters
 - choice of game play mechanics well-suited for the game's genre,
 - the look and feel of game level or world design,
 - user interface design and overlay, etc.
 - Little/no focus on game's software functional requirements.
- Collaborative design of game software arises through *shared online artifacts* and *persistent online chat* records [cf. Elliott, Ackerman, and Scacchi 2007, Scacchi 2002].

Design General

Home > Forums > Game Jam Winter 2012 > Expedient Interactive > Design General

Tagged: [designers](#), [Logo](#)

This topic has 2 voices, contains 1 reply, and was last updated by   RyanPatrick 75 days ago.

Viewing 2 posts - 1 through 2 (of 2 total)

Author	Posts
 <p>npetitti</p>	<p>January 16, 2012 at 10:03 pm #43031</p> <p>Hey everyone, this is where we will be posting anything design related. So if you have any questions/comments/concerns and you can't get on skype or in the game lab to talk about it then this is the place.</p> <p>This is a link to the GDD. It will be changing constantly but it will always be updated with the most current ideas. If you have any questions or any of it doesn't make any sense post it on here so we can clarify it for everyone. Its in a google doc so that means everyone can view it and change it. Please don't mess around with it too much unless you know that it is going to be official.</p> <p>https://docs.google.com/document/d/1bCUv1w9cpJgXILgqW1mF8mnwM4QULqz8kpNMqVx-kFo/edit?hl=en_US</p> <p>Designers: We need to start working on level design ASAP and I want everyone to be on the same page. I have a Legend of symbols that I want everyone to use so we're all on the same page. I don't have a scanner so I can't upload an image of it here but be sure to talk to me in the game lab tomorrow so I can show you. Oh this is Nathan btw, lead designer.</p>

January 16, 2012 at 11:37 pm #43036

 <p> RvanPatrick</p>	<p>I think you need to work on permissions there, as I can't get access to it.</p> <p>Also, if you could post a copy to the documents file in the dropbox that'd be great.</p>
---	--

August 9, 2011 at 3:40 am

#35928



JustinBritch

Yeah, those are all good ideas. And things that would defiantly be put in to a real game. As of now though, they need to sit in the "if time permits" area, because that skill feature needs a huge amount of resources devoted to it. So we will see if we can have time to start putting those in, but I really like your suggestions.

August 10, 2011 at 11:55 pm

#40195



pollend

how will AI behave for each type of monster

My Website:<http://gitastudents.com/~pollindm/index.html#>

Michael Pollind

high school 😊

August 11, 2011 at 1:47 am

#40196



IanMerion

Hey all. I'm currently working on stage 4 and will work on getting them all on tiled soon. A couple of questions:

What do you guys think about stat/xp tomes? I was thinking of hiding them in places the player might not think to look.

How about doors? It would make area transfers more interesting. I was thinking of adding a few here and there that lead to more enemies/secret tomes.

I was also thinking about adding breakable walls.

Also how about spike floors to represent impassable terrain? And spike traps to challenge the player to move through parts of the stage quickly?

That's all for now but I might have more queries later.

EDIT: Health pickups? Otherwise the player will probably die often. Unless you're making health regeneration crazy fast which I don't approve of.

August 11, 2011 at 4:17 pm

#40216



JustinBritch

Micheal:

The A.I. we are going to have right now should be as simple as possible. The melee units are going to move towards the player and try and hit him until he dies. They should be slower than the player, but should advance toward the player if they are on screen together. They should be able to walk down any thing, but they won't have the ability to jump up things (at least not yet). If there are other enemies in front of melee units, then they will just wait patiently for the other enemy to "move" or be killed. The snake will only be able to attack from 1/2 a screen away, so they will advance to the player until then and then start firing their slow moving projectile at what the players position was when they started the attack.

The boss is a totally different situation, but we will deal with that after these first enemies are done.

Game code sprint

- a game *code sprint* or *hackathon* [Wikipedia 2011] or *indie game jam* [Wikipedia 2012]
- Emphasize production of *useable game* within a pre-specified period of time, compared to other requirements.
 - buildable game source code
 - all game content assets provided
 - complete run-time executable installation

Game software testing and *post mortem*

- Game software testing
 - Little developer-oriented verification
 - if no functional requirements, then testing focuses on addressing non-functional requirements
 - Mostly independent end-user playtesting [Fullerton, et al. 2004].
 - Game competition judges act as non-aligned end-user play testers
- End-user demonstration and game showcase
 - Not "demo or die," but shared developer experience
- Post mortem [Grossman 2003]
 - common for game developers

Collaborative game software development kits (SDKs), libraries and components

- Commercial game development frameworks: *Microsoft XNA*, *GameMaker: Studio*, *Unreal Development Kit*, or *Unity 3D*, [Wu and Wang 2011]
- Free/open source software components for game development like *Blender* (3D modeling and animation), *OGRE* (graphics run-time environment), game engines like *Crystal Space*, *Delta 3D*, and dozens of others [Game Engines 2012].
- Current SDKs and frameworks tend to reinforce *one style (or genre) of game and game development*
 - domain-specificity does have its advantages for reuse and development process familiarization.

Game software development team management

- Teams not interested in financial incentives or cash rewards for their efforts
 - they want friendly competition, not cut-throat
- They do welcome opportunity to acquire and employ new, unfamiliar game SDKs in their project work.
- Emphasis on “winning” the competition is in shared experience, local “geek fame,” and similar forms of social capital.

Balancing game software engineering competition

- Team *skill and role-set balancing* that seeks to plausibly equalize the size, composition, and expertise of each game development team.
- *Experienced student game producers* help to organize the game design and development effort.
- Team composition is determined by event organizers (students) via semi-random assignment of participants to a team, so participants do not choose which team they join.
- Equalized team role-set composition enables the competition to resemble a *role-playing game*.

The Outcomes of game software development competitions

- Game day: teams showcase their game development results
 - External game publishing can follow after competition, for example, on Microsoft's *XBox Live Indie Game* marketplace
- Participants enact career contingencies as accomplished, upcoming game software developers ready for (entry-level) placement in "industry."
 - Game industry versus other non-game industries
- Role-based development efforts good for:
 - learning teamwork
 - individual contribution
 - shared responsibility, and
 - technical skill acquisition and demonstration.
- Participants learn how to confront and deal with team members who do not fulfill or honor their commitment to the team's effort, schedule, and product goals.

More Outcomes

- UCI VGDC game video demos at <http://www.clubs.uci.edu/vgdc/blog/showcase>
- What doesn't get addressed during game development competitions:
 - security
 - anti-cheating
 - commerce and payment systems (e.g., micro-transactions)
 - external user-centered requirements elicitation or market-driven focus group feedback
- Discovering the challenge of time-constrained, team-oriented computational thinking [cf. Wing 2006].

Related Game R&D Efforts

- *ACM Programming Contest*
 - focusing on production of correct solutions, not SE.
- Commercial or independent game industry sponsored competitions
 - *Microsoft Imagine Cup*
 - *Make Something Unreal* (Epic Games, Intel)
- Participation involves use of vendor-specific game software tools or game creation libraries
 - Limit technical choices and game genre
- Game Festivals
 - IGDA Global Game Jam, IndieCade, etc.
 - Focus on game as product, not teams, nor SE

More related efforts

- *Robocup* competition [Barrera, et al 2005]
 - The organization of the Robocup consciously fosters the use of free OSS software as a way of improving the level of the competition.
 - All software produced by the organization is therefore released under a free software license and most of the teams do share their code.
 - Winning code is distributed to next year's contestants (i.e., encourages design/code reuse)
- *Google Summer of Code*
 - Students "compete" to be selected to work on OSS project and receive financial stipend for successful internship. Not team-oriented, mostly code sprint.

Games in Software Engineering Education

- Teaching introductory and specialized SE concepts using games [Claypool and Claypool 2005, Sweedyk and Keller 2005, Wang 2011]
 - positive effect is that students are clearly motivated by game projects which likely resulted in higher enrollments and more effort put into the project.
- Games that model and simulate a team-oriented approach to SE process and project management education [Navarro and van der Hoek 2004, 2010, Longstreet and Cooper 2012, Zhu et al, 2007]
 - SE project work as a role-playing game
- Modding *as an approach to end-user game software engineering* using software extension techniques and tools common to OSS development [Scacchi 2004, 2011].

Observations, Lessons Learned and Conclusions

- Game software development competitions are fun, hard work, low-cost, short-term, intensive, and not motivated nor rewarded academically (no grades or tests given).
- Game software development competitions can serve as a testbed for exploring, observing, or evaluating new SE tools, techniques and concepts.
 - Equalized and balanced competitions represent time-compressed ways and means for conducting empirical SE studies.
- These competitions may help students and others in industry learn the value of presenting SE experiences that entail tough technical, time-constrained team collaboration challenges, that are perceived as a fun thing to do.

More observations and conclusions

- Game-centric SE may be a viable strategy for helping to *make SE education more fun and engaging*. Games are a medium and strategy for updating SE education.
- *Balanced team-oriented game development competitions* can be used as:
 - ways and means for advancing SE education
 - conducting empirical studies of SE processes and tools in time-compressed schedules [cf. Bendifallah and Scacchi 1989].

More observations and conclusions

- The *subjective criteria employed to evaluate the products or results of game development competitions* represent an expansion of topics addressing the importance of non-functional software requirements over functional requirements in this domain for software engineering.
- Game development competitions also represent a relatively unexplored domain for *empirical studies of collaborative teamwork in software development* [Mistrik, et al 2010],
 - those that rely on online artifacts (e.g., game design documents, persistent chat transcripts, game screen layout and artwork mockups) within shared repositories and other social media [FutureCSD 2012, Scacchi 2010].

Acknowledgements

- Thanks for the hundreds of students within the UCI Video Game Developers Club for their efforts. <http://www.clubs.uci.edu/vgdc/blog>
- The research described in this presentation has been supported by grants #0808783 and #1041918 from the National Science Foundation.
- No review, approval or endorsement implied.

References

- ACM (2012). ACM International Collegiate Programming Contest, Accessed 29 May 2012. [http](#)
- Barrera, P., Robles, G., Canas, J.M., Martín, F., Matellan, V. (2005). Impact of Libre Software Tools and Methods in the Robotics Field, *Proc. 2005 Workshop on Open Source Software Engineering*, ACM, New York.
- Bendifallah, S. and Scacchi, W. (1989). Work Structures and Shifts: An Empirical Analysis of Software Specification Teamwork, *Proc. 11th. Intern. Conf. Software Engineering*, Pittsburgh, PA, ACM and IEEE Computer Society, 260-270, May.
- Claypool, K. and Claypool, M. (2005). Teaching software engineering through game design, in *Proceedings of the 10th Annual SIGCSE Conference on innovation and Technology in Computer Science Education (ITiCSE '05)*, pp. 123–127, Caparica, Portugal.
- Elliott, M., Ackerman, M.S., and Scacchi, W. (2007). Knowledge Work Artifacts: Kernel Cousins for Free/Open Source Software Development, *Proc. ACM Conf. Support Group Work (Group07)*, Sanibel Island, FL, 177-186, November 2007.
- Franke, N., & Hippel, E. V. (2003). Satisfying heterogeneous user needs via innovation toolkits: the case of Apache security software. *Research Policy*, 32(7), 1199-1215.
- Game Engines (2012). *Free, cross-platform, real-time 3D engines*. [http](#). Also see *List of Game Engines*, [http](#), Accessed 29 May 2012.
- Fullerton, T., Swain, C., Hoffman, S. (2004). *Game Design Workshop: Designing, Prototyping and Playtesting Games*. CMP Books, February 2004.
- FutureCSD (2012). Workshop on the Future of Collaborative Software Development, *ACM Conf. Computer-Supported Cooperative Work*, Seattle, WA, February. Accessed 29 May 2012, [http](#)
- Hamilton, I. (2011). UCI Students Build Games in a Week, *Orange County Register*, 13 April 2011, [http](#) Accessed 17 February 2012.
- IEEE GameSIG (2012). *Intercollegiate Game Showcase*. Accessed 29 May 2012. [http](#)
- Longstreet, S. and Cooper, K. (2012). Using Games in Software Engineering Education to Increase Student Success & Retention, Online poster, Accessed 30 May 2012. [http](#)
- Mistrić, I., Grundy, J., van der Hoek, A., and Whitehead, J. (2010), *Collaborative Software Engineering*, Springer, New York.
- Navarro, E. and van der Hoek, A. (2004). Software Process Modeling for an Educational Software Engineering Simulation Game, *Software Process Improvement and Practice*: 10 (3), 311-325.
- Navarro, E. and van der Hoek, A. (2010). Multi-Site Evaluation of SimSE, *Proc. 40th. ACM Technical Symposium on Computer Science Education*, Chattanooga, TN, March 2009.

References

- Rogers, S. (2010). *Level Up!: The Guide to Great Video Game Design*, Wiley.
- Salen, K. (2007). Gaming Literacies: A Game Design Study in Action. *J. Educational Multimedia and Hypermedia*, 16(3), 301-22.
- Scacchi, W. (2002). Understanding the requirements for open source software development, *IEE Proceedings--Software*, 149(1), 24-39, February 2002.
- Scacchi, W. (2004). Free/open source software practices in the game development community, *IEEE Software*, 21(1), 59-67, January/February.
- Scacchi, W. (2010). Collaboration Practices and Affordances in Free/Open Source Software Development in I. Mistrík, J. Grundy, A. van der Hoek, and J. Whitehead, (Eds.), *Collaborative Software Engineering*, Springer, New York, 307-328, 2010.
- Scacchi, W. (2011). Modding as an Open Source Software Approach to Extending Computer Game Systems, in S. Hissam, B. Russo, M.G. de Mendonca Neto, and F. Kan (Eds.), *Open Source Systems: Grounding Research, Proc. 7th. IFIP Intern. Conf. Open Source Systems*, 62-74, IFIP ACIT 365, Salvador, Brazil, October 2011. Also appears in *Intern. J. Open Source Software & Processes*, 3(3), 2012.
- Schell, J. (2008). *The Art of Game Design: A book of lenses*, Morgan Kaufman.
- Sweedyk, E. and Keller, R.M. (2005). Fun and games: a new software engineering course, *ACM SIGCSE Bulletin*, 37(3), 138–142.
- von Hippel, E. (2005). *Democratizing Innovation*, MIT Press, Cambridge, MA.
- von Hippel, E., & Katz, R. (2002). Shifting Innovation to Users via Toolkits. *Management Science*, 48(7), 821-833.
- Wikipedia, *Hackathon*, [http](http://en.wikipedia.org/wiki/Hackathon). Also known as *Code Sprint*. Accessed 17 February 2012.
- Wing, J.M., (2006). Computational Thinking, *Communications ACM*, 49(3), 33-35.
- Wang, A.I. (2011) Extensive Evaluation of Using a Game Project in a Software Architecture Course, *ACM Transactions on Computing Education* 11(1), February 2011.
- Wu, B. and Wang, A.I. (2011). Game Development Frameworks for SE Education, *2011 Intern. Games Innovation Conference*, IEEE Consumer Electronics Society, Orange, CA, 98-100.
- Zhu, Q., Wang, T., and Tan, S. (2007). Adapting game technology to support software engineering process teaching: from SimSE to MO-SEProcess, in *Proc. 3rd Intern. Conf. on Natural Computation, (ICNC '07)*, 777–780, Haikou, China, August.