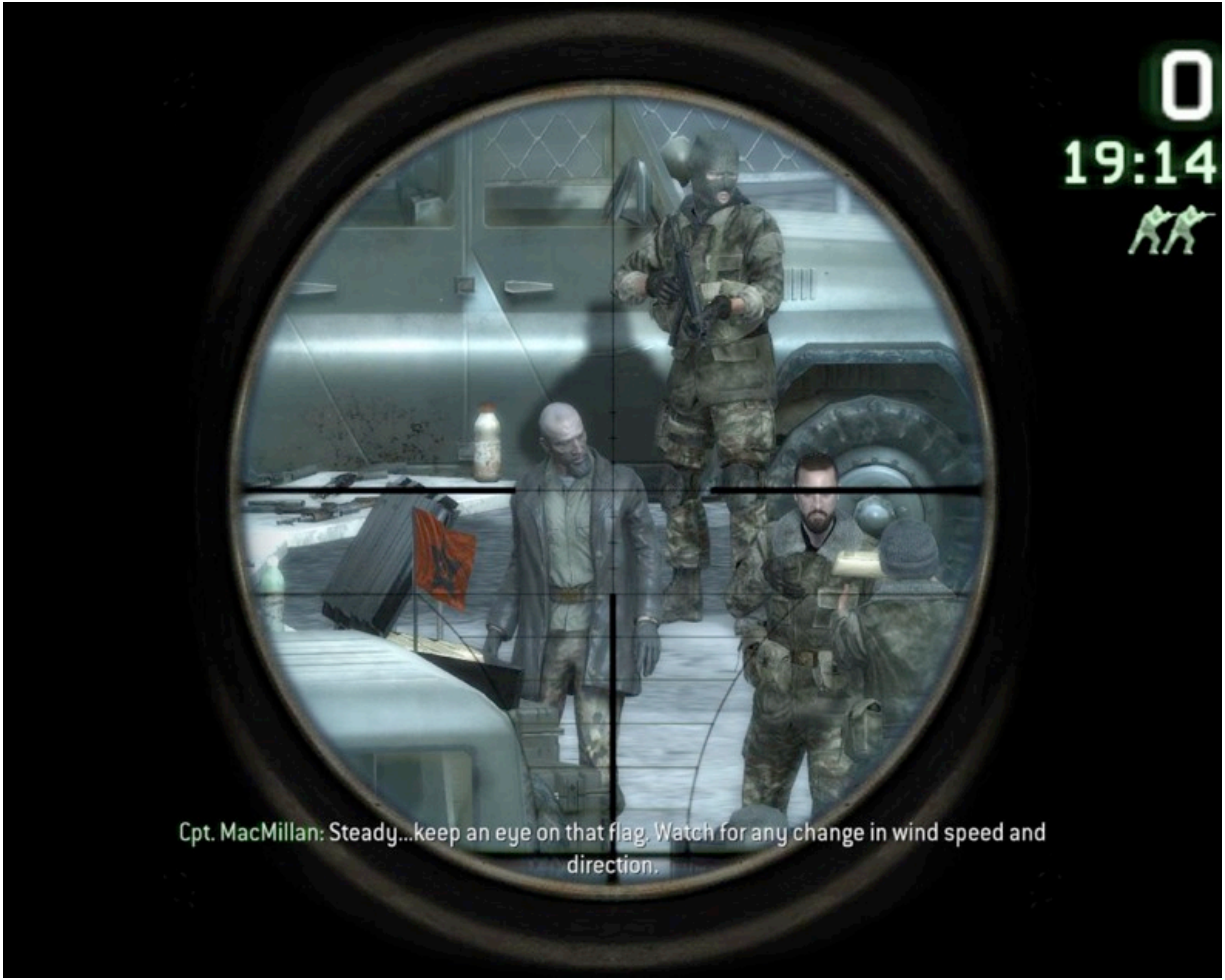
HALO[1] (Highly Addictive, sociaLly Optimized) Software Engineering

Swapneel Sheth, Jonathan Bell, Gail Kaiser
Department of Computer Science, Columbia University
New York, NY 10027
{swapneel, jbell, kaiser}@cs.columbia.edu

[1] There is no connection between our proposal and any games with similar names. The authors have no affiliations with any companies involved in producing such games... except as customers.



Cpt. MacMillan: Steady...keep an eye on that flag. Watch for any change in wind speed and direction.

AT.01

Blazix
<Obscurum>

FarmVille

HALO Software Engineering

- These games mirror the competitive-collaborative nature of SE
- We propose a new approach to Software Engineering called HALO (Highly Addictive, sociaLly Optimized)
- Use the benefits of gamification to aid the SE process
- Originally proposed by Jim Whitehead in his roadmap for collaboration in SE (2007)

Background: Behavioral Psychology

- Operant Conditioning
  - Rewarding participants for good behavior
- Flow
  - Single-minded immersion
- Applications to games
- Jane McGonigal's Blissful Productivity
  - Gamers are happier working to beat a game than relaxing

HALO Game Mechanics

- HALO represents SE tasks as quests
  - Simple: closing a bug
  - Complex: porting the code to a different OS or platform
  - E.g., give basic training to a new intern and introduce him to artifacts like code repositories and bug report systems
- Create parties for difficult quests - this would highlight the collaborative nature of SE
  - E.g., create a party of developers covering the several kinds of expertise needed to internationalize the code for a new region
- Quests can be chained into series of quests - an analog for representing the multiple steps with intermediate deliverables for complicated use cases or bug fixes, or daily build/test processes

HALO Game Mechanics (2)

- Quests can be created in HALO during the initial planning stages for SE projects
- New quests and quest series can be added on the fly, as needed
- Doesn’t have to follow the MMORPG concepts strictly - concepts from other games or familiar to conventional SE like priorities and deadlines can be added

Blissful Productivity in HALO

- Operant Conditioning
  - Social in-game rewards such as titles, leveling
  - Currency points could map to real life - better parking spaces, free lunch, gift cards
  - Could be used to provide metrics for existing programs such as "Employee of the Month"
  - Rewards scaled with time in game (becoming less common but more valuable); larger rewards for more difficult tasks and collaborating with others
- Flow Theory
  - Clear Goals, Concentration, Loss of self-consciousness, Direct feedback, Balance of ability and challenge

HALO Software Development Environment

- IDE plugin that would communicate with users and keep track of progress and achievements
- Task sensitive and context sensitive for integration with the SE lifecycle
- Initially text based, although could become 3D
- HALO would need an automated way for figuring out when quests have been completed
  - Examples: running a unit or regression test, triggered by events such as code check-ins, statically analyzing the code, explicit feedback from users

Related Work

- CHIME (1999)
  - Immersive VR for collaborative SE
  - Allowed users to walk around in a 3D world and interact with SE artifacts such as code files, bug reports, and email archives
  - Focused on the artifacts of SE, rather than tasks; HALO focuses on the latter
- MARVEL (1987) and Oz (1993)
  - Automated software workflow for team software development later extended to geographically distributed teams and a web-based GUI

Related Work (2)

- SE education games - Software Hut (1977), SimSE (2004), Card Game (2005)
  - SimSE was designed a single-player game - no multiplayer features
  - Software Hut and Card Game are primarily competitive games - limited or no collaborative aspects
  - All games focused towards teaching SE in a classroom - not intended for "real-world" SE
  - Professional SE is usually done collaboratively in teams and this is our focus
- Collaborative games for science education - DinoQuest (2008)
  - These games focus on general science education, not SE

Research Agenda

- Mapping of different SE processes and methodologies to games
  - Should the Agile methodology and the Waterfall model have the same game mechanics?
  - If there is Global Software Development, do we need different game mechanics?
- Building game engines for software development
  - Would these be similar to the traditional game engines?
  - What would be the software architectures and design patterns for these games engines?

Research Agenda (2)

- Quest design
  - How can quests be created so they are satisfying and engaging?
  - What kind of quest templates do we need for SE?
  - How can existing systems like bug tracking be integrated into HALO?
- Evaluation of games like HALO
  - Do we need new methodologies?

Conclusion

- New approach to SE called HALO (Highly Addictive, sociaLly Optimized)
- Builds upon properties of popular online collaborative games
- Described the game mechanics of HALO and how it would fit into typical SE processes
- Highlighted some of the future research challenges

Acknowledgements

- The authors are members of the Programming Systems Laboratory, funded in part by NSF CNS-0717544, CNS-0627473 and CNS-0426623, and NIH 2 U54 CA121852-06

Photo Attribution

- Dunechaser (Flickr)
- http://joshynet.wordpress.com/
- http://www.cod4central.com/

HALO (Highly Addictive, sociaLly Optimized) Software Engineering

Swapneel Sheth, Jonathan Bell, Gail Kaiser
Department of Computer Science, Columbia University
New York, NY 10027
{swapneel, jbell, kaiser}@cs.columbia.edu