Taxonomy of Video Game Bugs

Based on slides and research originally by Chris Lewis, published at the Foundations of Digital Games 2010 conference
Team Meetings

- Will be meeting
  - Pixture, 3:30pm, today (Wednesday)
  - Asterogue, 4:30pm, today (Wednesday)
- We’re Screwed (Thursday, 11am)
- Focus on UML
Upcoming deadlines

- Thursday, January 31: End of Sprint 1
  - 1 full day left in Sprint 1

- Friday, February 1
  - Sprint 1 report due
  - Sprint 2 plan due

- Friday (Feb. 1): team status reporting
  - Due by midnight
  - Report on team activities this week
  - Be sure to use team status reporting template
    - courses.soe.ucsc.edu/courses/cmps171/Winter13/01/pages/teamstatus-template
Upcoming events

- Winter Job & Internship Fair
  - Tuesday, February 5
  - See [http://careers.ucsc.edu/](http://careers.ucsc.edu/) for more information
Lab Cleanup Schedule

- This week: Tearable World
- Next week: We’re Screwed!

- Team duties:
  - Ensure overflowing trash cans are emptied to bin outside in 3rd floor courtyard (anytime during week)
  - By 5pm Monday and 5pm Friday (unless things get out of control, then more often):
    - Pick up food containers, bottles, etc.
    - Pick up stray craft materials, pens, etc and return to drawers
    - Clean off tables in conference rooms and big circular table
    - Report any major soda/food spills to me, so we can call cleanup crews
    - Put controllers/game boxes/etc. away (tidy up game area)
    - Report any cleaning materials needed
Sprint Report

- The output of a Sprint Retrospection meeting
  - Meeting held after the end of every sprint
  - Provides time to reflect on what happened, determine improvements

Questions to be answered:
- What things should we stop doing?
  - Actions/activities the team did that were harmful, hurt progress
- What should we start doing?
  - Actions/activities that will improve how we do work
- What is working well that we should continue to do?
  - What is working well, and shouldn’t be touched?
- What work was completed, and not completed?
  - User stories that were completed, not completed
- What is our rate of completing work?
  - Story points completed, ideal work hours completed
  - Story points/day, ideal work hours/day
  - Average story points/day & ideal work hours/day across all sprints
  - Final sprint burndown chart
The Jesus Shot
Background

**Beizer** created a taxonomy of **software engineering bugs**

“There is no universally correct way to categorize bugs”

**Many other software engineering taxonomies exist**

Too specific or general, assumes code knowledge, doesn’t help games
What’s In It

Game glitches are unintentional

Game design is a creative endeavor
Taxonomy - Overview

Implementation failures

Non-temporal
- Position invalid (clipping)
- Invalid graphical representation
- Invalid value change
- Artificial stupidity
- Information
- Action
- Invalid position over time
- Invalid context state over time
- Invalid event occurrence

Temporal
- Invalid event occurrence
- Interrupted event
- Implementation response issues

Object out of bounds at all times
Object out of bounds for specific state
Invalid information access
Lack of required information
Action when not allowed
Action not possible
Position Invalid
Invalid Graphical Representation
Invalid Graphical Representation
Invalid Graphical Representation
Lack of Information
Invalid Position Over Time
Taxonomy - ?
Discussion

Not mutually exclusive
Bugs overlap depending on perspective

Not complete
Can it ever be complete?
Testing Aid
Taxonomies improve number of relevant test cases produced by testers

Validation of bug detection software

Theoretical understanding