Bug Reporting

Paul Jackson

School of Informatics
University of Edinburgh
Software entomology

Bugs and new features have something in common: both are requirements from users. How does one keep track of them?

Many projects use a *bug tracking system* for both bug reports and new feature requests.

Open source tools include

- Bugzilla
- Gnats
- Trac
- RT (used by our support).

These provide extensive support for receiving, tracking, notifying, monitoring, etc.
<table>
<thead>
<tr>
<th>Ticket</th>
<th>Summary</th>
<th>Component</th>
<th>Version</th>
<th>Milestone</th>
<th>Type</th>
<th>Owner</th>
<th>Status</th>
<th>Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>#3292</td>
<td>OOS on rejoin with new pathfinder</td>
<td>Core engine</td>
<td></td>
<td>Alpha 19</td>
<td>defect</td>
<td></td>
<td></td>
<td>new</td>
</tr>
<tr>
<td>#3471</td>
<td>Units not detecting invalid path.</td>
<td>Core engine</td>
<td></td>
<td>Alpha 19</td>
<td>defect</td>
<td></td>
<td></td>
<td>new</td>
</tr>
<tr>
<td>#3505</td>
<td>Pathfinder - Units in formation stuck frequently</td>
<td>Core engine</td>
<td></td>
<td>Alpha 19</td>
<td>defect</td>
<td></td>
<td></td>
<td>new</td>
</tr>
<tr>
<td>#3551</td>
<td>[PATCH] Prohibit developer overlay cheats in rated games</td>
<td>UI &amp; Simulation</td>
<td></td>
<td>Alpha 19</td>
<td>defect</td>
<td></td>
<td></td>
<td>new</td>
</tr>
<tr>
<td>#3549</td>
<td>Secure authentication - prevent joins as a different player</td>
<td>Core engine</td>
<td></td>
<td>Alpha 20</td>
<td>defect</td>
<td></td>
<td></td>
<td>new</td>
</tr>
<tr>
<td>#3255</td>
<td>[PATCH] Prevent replay overwrites by using date and sequential ID</td>
<td>Core engine</td>
<td></td>
<td>Alpha 19</td>
<td>defect</td>
<td>elexis</td>
<td></td>
<td>new</td>
</tr>
<tr>
<td>#3271</td>
<td>OOS on rejoin - different mirage order</td>
<td>Core engine</td>
<td></td>
<td>Alpha 19</td>
<td>defect</td>
<td></td>
<td></td>
<td>new</td>
</tr>
<tr>
<td>#3526</td>
<td>Build a tower in enemy territory</td>
<td>UI &amp; Simulation</td>
<td></td>
<td>Alpha 19</td>
<td>defect</td>
<td></td>
<td></td>
<td>new</td>
</tr>
<tr>
<td>#3545</td>
<td>[PATCH] Crash the game using cheats</td>
<td>UI &amp; Simulation</td>
<td></td>
<td>Alpha 19</td>
<td>defect</td>
<td>stanislav69</td>
<td>assigned</td>
<td>Oct 23, 2015</td>
</tr>
<tr>
<td>#1791</td>
<td>Units command queue is reset when they enter new formation</td>
<td>UI &amp; Simulation</td>
<td></td>
<td>Alpha 20</td>
<td>defect</td>
<td></td>
<td></td>
<td>new</td>
</tr>
<tr>
<td>#2001</td>
<td>Melee units with big maximum range can attack through walls</td>
<td>UI &amp; Simulation</td>
<td></td>
<td>Alpha 20</td>
<td>defect</td>
<td></td>
<td></td>
<td>new</td>
</tr>
<tr>
<td>#2303</td>
<td>Update tutorials and increase their visibility</td>
<td>UI &amp; Simulation</td>
<td></td>
<td>Alpha 20</td>
<td>defect</td>
<td></td>
<td></td>
<td>new</td>
</tr>
</tbody>
</table>
Units not detecting invalid path.

We played a match today with elexis and ffm, and we noticed that units would often try to go from a point 'a' to a point 'b' without realising they wouldn't be able to reach it. So they just walk into the void.

The only we could workaround it, is by cancelling orders, and removing formations (setting it to none)

FFM stated formation should be disabled whereas I think it should be set to none by default.
What can users do?

It’s next year, you’re doing your Operating Systems practical.

You’ve done a bit of work earlier in the day, and think you’re in good shape for the deadline tomorrow.

In the evening, you start up the system . . . and it doesn’t work.

You know that if you report the problem to the course forum there’s a good chance that the lecturer will reply that evening.

What do you say in your post?
What do you think?

What is the most important thing to do?

1. Keep the bug report concise - no more than 30 lines of 80 characters

2. Include tediously detailed information about what exactly you did that didn’t work

3. Include full information about what operating system etc. you’re using

4. Make an intelligent attempt at diagnosing the problem

2, 3 yes
then 4 if you can
1 no, but starting with concise summary is good
How to deal with bugs as a user

and make programmers love you.

When something goes wrong, STOP! Don’t do anything until you’ve engaged your brain.

Compose your bug report. This should:

▶ enable the developer to reproduce the bug. That means they need to know exactly what you did, what your system is, etc.

▶ failing that, tell them exactly what went wrong. Describe everything: what you typed/clicked, what messages appeared. Don’t edit according to what you think relevant – you’re probably wrong in what you think.

▶ If you can, try to diagnose the problem – but keep your diagnosis completely separate from the report of what happened.

You will often be the developer, and you have to deal with bugs in other people’s software. Some of this may be open source. You may have to ask for help on stackoverflow or mailing lists. If so . . .

Strongly Suggested Reading:
http://www.catb.org/~esr/faqs/smart-questions.html
From Eric Raymond, open-source software advocate.

Practice similar skills when phrasing questions on course discussion forums.