Lighthouse

Pre-final Presentation

Project Summary

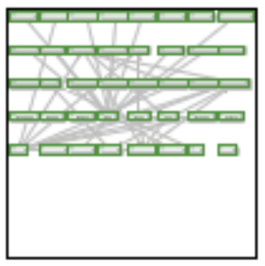
- Heuristic Evaluation
- User Evaluation
    - Phase 1: Discovering Lighthouse without guidance
        - Are users able to figure out what features do?
        - Are users able to filter to show relevant classes?
    - Phase 2: Collaborating in a group task
        - Do users integrate Lighthouse into their work?
        - Do users use Lighthouse to avoid conflicts?

Heuristic Results

- Very poor user feedback
  - Errors may go completely unnoticed
  - Very little feedback in general
- Unwieldy display
  - Mouse-only interaction
  - Doesn't support common conventions
  - Difficult to differentiate between bugs and intended behavior
  - Search not integrated

EmergingDesignView

ATN1Plugin
ATMCaseStud
BankCASPlugi
GenericPlugin
WordPlugin
AJAXPlugin
JavaPlugin
WorldBankPlugin
SystemPlugin
WebPlugin
Withdrawal
HTMLPlugin
Keypad
XLSPLugin
PerformanceP
RESTPlugin
BankECMPlug
Globals
ATMBasePlugi
CalendarPlugi
InternetBankir
CashDispense
ATMException
BankDatabase
RemotePlugin
RTFPlugin
MailPlugin
Plugin
Deposit
Screen
DepositSlot
TXTPlugin
ATM
KUDOPlugin
PDFPlugin
Account
Transaction
YodaPlugin
Fee



Experimental Method

- Lighthouse Introduction and Briefing
- Pre Test Questionnaire
- Phase 1: Exploration and Interview
- Training & Demo
- Phase 2: Coding & Conflicts, Interview
- Post Mortem Interview

Division of Labor

Design of Phases

- Phase 1 : Ankita & Caitie
- Phase 2 : Arthur & Chris
- Documents written/finalized as a team

Roles during Tests

- Ankita: Test supervision, Interviews
- Caitie: Briefing, Note taking, Training
- Arthur, Chris: Collaborators, Lighthouse/SVN Maintenance

Overall Scenario

A seasoned coder has just joined a new technology company, Mittelos Software Corp.

First, explore and get familiar with an Eclipse plugin called Lighthouse, a collaboration tool of sorts!

Given few assignments to complete, collaborators Arthur and Chris online, ready to work with user.

All on equal footing, exploring new territory, aim to see how Lighthouse will aid the process of creating code together.

Phase 1

Tasks:

1. Explore & identify icons in Lighthouse pane
2. Focus on diagram layout, class visualization mode icon
3. Focus on filters

Materials:

Task descriptions

Task worksheet

Interview:

Focuses on intuition and understanding regarding icons and navigation

Phase 2

Tasks:

1. Code "Fee" class
2. Add "overdraft" attribute to "Withdrawal" class
3. Combine code into class created by collaborator
4. Collaborate to merge all work done by participants

Materials:

Task descriptions

ATM source code

Interview:

Specific to use in coding situations

Additional Info Gathering

Pre Test Questionnaire

- Demographic data

Post Mortem Interview

- Questions regarding the overall Lighthouse experience

Recording Methods

- Notes
- Screen Recording
- Audio Recording

Pilot Study

Observed Events

- Phase 1 yielded expected results
- Phase 2
    - Total lack of collaboration
    - Tasks were confusing
    - Unfamiliarity with banking concepts
    - Frustration with bugs and unexpected events
    - Privacy concerns and fear of mistakes

Actions Taken (Phase 2)

- Developed UML and more specific instructions
- Contingencies to reset Lighthouse

User Study

- Modifications for Phase 2 showed improvement
- Bugs remained an issue, even with contingency
- Privacy concerns
- Suggestions from users:
  - Allow their changes to be hidden on demand
  - Better notification of committed code
  - Allow for resetting changes shown to others
  - Concern with showing confusing modifications

User Impressions

- "I saw Arthur [and Chris] make modifications, but when I updated [svn], nothing had changed."
- "I wouldn't use it very often because it didn't add much."
- "There's nothing wrong with the system, it just doesn't help me."
- "A change log I could pull up whenever I wanted would be more helpful."

User Impressions (cont'd)

- "Can't see what people are doing which is just annoying, which results in people pressuring others to commit, which just makes people angry."
- Felt that using it in a large project would be unwieldily
- Found it nice to know what people were changing
- Didn't have to waste time doing a diff or update periodically because the subject could just ask the other programmers what they were doing

Problems Encountered

- Enormous frustration with bugs and crashes
  - When it fails, everyone is confused and lost
  - Must not fail, or it will do more harm than good
- Phase 2 somewhat open ended
  - Leads to small differences in every experiment
    - With improvement made from pilots, changes are fairly trivial
- Difficulty in finding industry professionals
  - Professionals tend to ignore call for subjects
  - Call effective on "friends" basis
  - Opened for past industry experience users
- Users expect to see code changed by others

Impressions & Conclusions

- Industry users seem to be more enthusiastic about using and recommending Lighthouse to friends
- Lighthouse should implement additional features:
  - Add search, reset, and user documentation
  - Streamline the menus and icons
  - Make it clearer which code has been checked in
- Additional things to investigate if we had more time:
  - Lighthouse with different sized projects/groups
  - The differences in usage between academic and industry users (currently just speculation)
  - Lighthouse in long term usage

Timeline

10/07: Initial meeting with the Lighthouse team
10/19: Presentation of evaluation plan
10/22: Initial heuristic evaluation completed
10/29: Completed initial test design
11/05: First pilot test
11/08: Second pilot test
11/09: Redesigned test based on pilot feedback
11/12: First non-pilot test
11/17: Second test
11/23: Present pre-final results
11/24: Finish testing
12/12: Submit final project