Comp-361 : Testing Strategies
Lecture 9

Alexandre Denault
Computer Science
McGill University
Winter 2008

Changes in the Schedule

February
- 13th (Friday) - No class
- 20th (Friday) - Guest Lecturer - Janina Szkut
- 27th (Friday) - Guest Lecturer - Marc Boscher

March
- 20th (Friday) - Guest Lecturer - Adam Blahuta
- 27th (Friday) - Guest Lecturer - Sandy Thomson

Requirements for Demo (tentative)

- You can start 2 player game
- You can choose your ships (at least 4 types available)
- You can place your ships on a map
- You can start and end players turns
- You can move ships around
  - (and it is registered by other player)
- You can do a basic (shell) attack
  - (and it is registered by other player)
- Fog of war works
  - (and you can remove it with a cheat)
- Collision rules are enforced
- You can save/resume games
- Simple lobby system (5 students team)

What is not needed

- Support for more than 2 players
- All weapons and ships
  - Don't really need any weapon other than basic attack
  - Don't need any of the complicated ships
    (multi radar or rather large)
- Ability to sink enemy ship
- Ability to win the game

- Functionality Testing
- Playability Testing

Is the game working like it should?

How to test for functionality

Unit Tests

Checklist Tests

Make a list of all the functionalities in the game.

How often do I check the list?

Once a day?
Once a week?
Every time there is a big commit?

Who does the testing?

Everyone in the team
Especially important to test other people's stuff

Who's in charge of the testing?

Name a QA person

Where to focus?

Focus your testing on the extreme conditions.
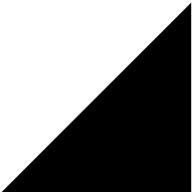
How do I know an interface is good?

Is my interface usable?

Preparing for the test

- Choose <u>one</u> team member to run the test.

- Try the test yourself first.

- Think about the questions you might want to ask during the test.

- Avoid taking about the game before the test.

- Ask them to think aloud.

- Take <u>good</u> notes.

- Feel free to ask them to elaborate on what they say.

  - Don't comment on what they say.

  - Use your best poker face.

- Protect your participants.

  - Make sure they don't get frustrated.

  - You can give hints if they get stuck.

-

What to look out for?

- Where do they click?
- What are the looking at?
  - Where is the cursor?
- What seems complicated?
- What seems simple?

Take proper notes

The tester didn't understand how the game worked.

The tester didn't know how to attack.

The tester couldn't figure out which button would allow him to attack.

Common Problems

- Users are unclear on the concept.
- The stuff they are looking for isn't there.
- There is too much stuff going on.

- Do have other people test your game
- Do listen to your testers
- Don't believe everything they say
- Don't overreact to every problems they have

What's next?

No more regular classes.

Guest lecture on February 20$^{\text{th}}$
(can include project information)