

CPSC 427

Video Game Programming

Debugging



Helge Rhodin



Setup

@Helge: Pressed record?

@Class: Logged into iClicker cloud?

Logistics: Exam slot?

- *Final cross-play session*
- *Industry jury*
- *Awards*
- *Attendance mandatory*
- *Sheduled: Dec 18th, 19:00*

CPSC 427

Video Game Programming

Debugging



Helge Rhodin

Debugging

- *There will be bugs...*
- *Strategies for Fixing?*

Learning goals:

- *Knowing about different debugging techniques*
- *When to look for what type of bug*
- *Strategies for avoiding bugs!*

Debugging

- *There will be bugs...*
- ***Strategies for Fixing?***
 - Anticipate
 - Reproduce
 - *Things get terribly difficult if randomness is involved!*
 - Localize
 - Use proper debugging tools

Task: Recall bugs that you faced

- *Those that you encountered early*
- *Those you had to track down*

Catastrophic Software Bugs...



Ariane 5 Flight 501, 4 June 1996



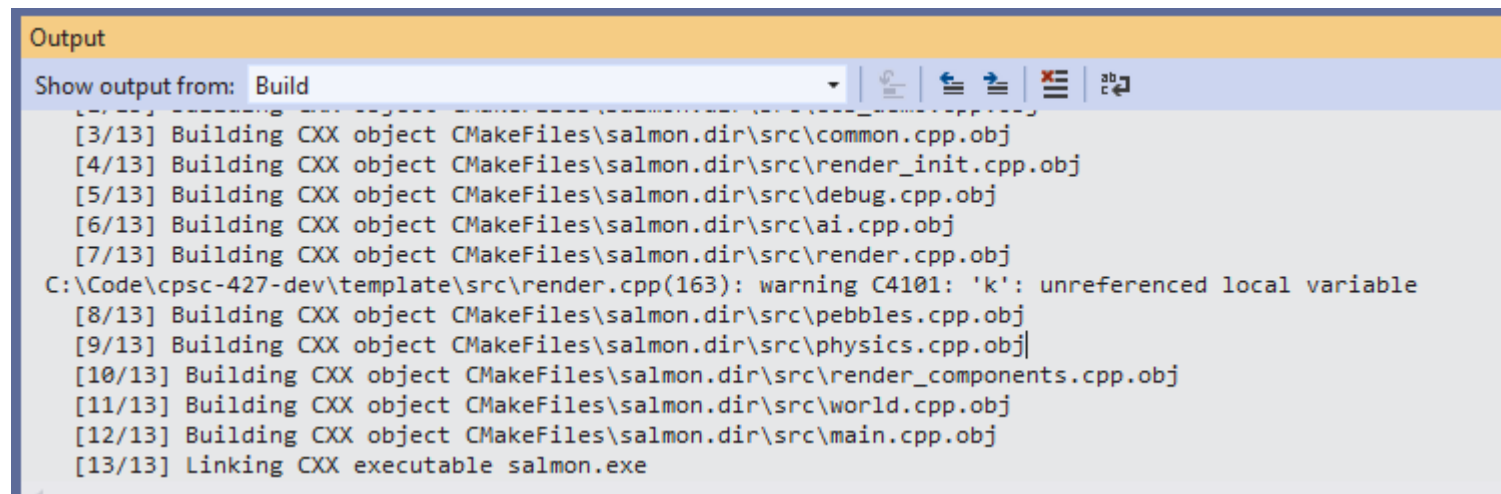
PacMan at level 256

Debugging: Strategies for Fixing?

- Anticipate I
 - *Unit tests*
 - *Logging*
 - *Explicit tests for “what can go wrong” (assert)*
 - Anything that can go wrong will go wrong... at the worst possible time
 - *State/play saving and loading speeds up debugging*
 - *Visual testing (early)*
 - *Avoid randomness (use seed for rnd)*
- Reproduce
- Localize
- Use proper debugging tools

Debugging: Strategies for Fixing?

- Anticipate II: *your compiler (with `-Wall` enabled) is your friend*
 - *“This enables all the warnings about constructions that some users consider questionable, and that are easy to avoid”*
- Reproduce
- Localize
- Use proper debugging tools



```
Output
Show output from: Build
[3/13] Building CXX object CMakeFiles\salmon.dir\src\common.cpp.obj
[4/13] Building CXX object CMakeFiles\salmon.dir\src\render_init.cpp.obj
[5/13] Building CXX object CMakeFiles\salmon.dir\src\debug.cpp.obj
[6/13] Building CXX object CMakeFiles\salmon.dir\src\ai.cpp.obj
[7/13] Building CXX object CMakeFiles\salmon.dir\src\render.cpp.obj
C:\Code\cpsc-427-dev\template\src\render.cpp(163): warning C4101: 'k': unreferenced local variable
[8/13] Building CXX object CMakeFiles\salmon.dir\src\pebbles.cpp.obj
[9/13] Building CXX object CMakeFiles\salmon.dir\src\physics.cpp.obj
[10/13] Building CXX object CMakeFiles\salmon.dir\src\render_components.cpp.obj
[11/13] Building CXX object CMakeFiles\salmon.dir\src\world.cpp.obj
[12/13] Building CXX object CMakeFiles\salmon.dir\src\main.cpp.obj
[13/13] Linking CXX executable salmon.exe
```

Debugging

- ***Strategies for Fixing?***
 - Anticipate
 - Reproduce
 - *When does it happen?*
 - *Logging + unit tests*
 - *Record/load gameplay*
 - Localize
 - Use proper debugging tools

Debugging

- ***Strategies for Fixing?***
 - Anticipate
 - Reproduce
 - Localize
 - *In time: version control*
 - *In place: logging*
 - Divide and Conquer
 - *Minimal trigger input*
 - *Don't guess; measure*
 - Use proper debugging tools

Debugging

- ***Strategies for Fixing?***
 - Anticipate
 - Reproduce
 - Localize
 - Use proper debugging tools
 - *Run with debug settings on*
 - *Run within a debugger*
 - Set breakpoints
 - Examine internal state
 - *Learn debugger options*

Exchange Experiences

- ***Catastrophic failures?***
- ***Debugging strategies that work for you***
 - Which ones don't?
 - *Can others make them work?*
- ***Elect a chair, report your groups most interesting bug and its fix***

Teams of 4

Debugging

(From Waterloo ECE 155, Zarnett & Lam)

- ***Strategies for Fixing?***
 - Scientific method.
 1. Observe a failure.
 2. Invent a hypothesis.
 3. Make predictions.
 4. Test the predictions using experiments and observations.
 - Correct? Refine the hypothesis.
 - Wrong? Try again with a new hypothesis.
 - Repeat



Debugging (From Waterloo ECE 155)

More (Human Factor) Strategies

- Take a Break/Sleep on it
- Code Review
 - Look through code
 - Walk someone through the code
- Exchange ideas on piazza

Debugging

More (Human Factor) Strategies

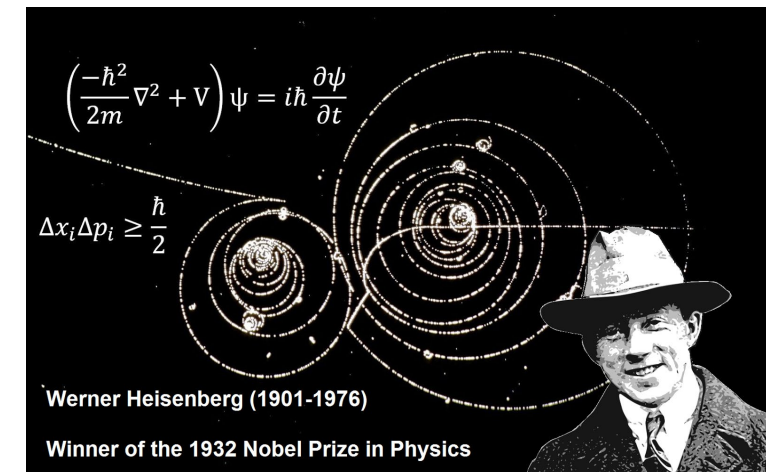
- Question assumptions
- Minimize randomness
 - Use same seed
- Check boundary conditions
- Disrupt parallel computations



Debugging (From Waterloo ECE 155)

More Strategies

- Know your enemy: Types of bugs
 - Standard bug (reproducible)
 - Sporadic (need to chase – right input combo)
 - Heisenbug
 - Memory (not initialized or stepped on)
 - Parallel execution
 - Optimization



Hard Bugs (cheat sheet)

- *Bug occurs in Release but not Debug*
 - Uninitialized data or optimization issue
- *Bug disappears when changing something innocuous*
 - Timing or memory overwrite problem
- *Intermittent problems*
 - Record as much info when it does happen
- *Unexplainable behavior*
 - Retry, Rebuild, Reboot, Reinstall
- *Internal compiler errors (not likely)*
 - Full rebuild, divide and conquer, try other machines
- *Suspect it's not your code (not likely)*
 - Check for patches, updates, or reported bugs



6	Mon	9-Oct	Thanksgiving		
	Wed	11-Oct	Lecture 7: Collisions and simple physics Tutorial: Face2Face grading A1	Peer review A1	
	Thu	12-Oct	Lecture 8: Simulation basics (makeup Monday)	Milestone 1	
7	Mon	16-Oct	Lecture 9: Simulation advanced		
<input type="checkbox"/>	Wed	18-Oct	Lecture: Team-report M1 Tutorial: Face2Face grading M1		
8	Mon	23-Oct	Lecture: IO and observer		
	Wed	25-Oct	Lecture: Debugging		
9	Mon	30-Oct	Lecture: Curves and splines	Milestone 2	
	Wed	1-Nov	Lecture: Team-report M2 Tutorial: Face2Face grading M2		
10	Mon	6-Nov	Lecture: Simple AI Cross-play M2	Assignment 2	
	Wed	8-Nov	Lecture: Two-player AI Tutorial: Face2Face grading A2	Peer review A2 (due on Thursday)	
11	Mon	13-Nov	Break		