18-642 Recitation #11

April 13, 2018
Updates

- **Homework:**
  - Homeworks #36, 33 (note: out of order) due Sunday 4/15 @ 11pm
  - Homework #38 due Sunday 4/22 @ 11pm; security stories will be presented in class by everyone on 4/25

- **Project:**
  - Project 9 due tonight @ 11pm
  - Project 10 out, due 4/27 @ 11pm, last project

- **No recitation next week (Carnival!)**
- **Exam #2 is coming up 5/2**
- **Be sure to fill out question sheet to receive attendance points today!**
Today

• Project 10
• HW 36 brainstorm activity
Project 10

• Some new mazes
• Get everything up-to-date
• Write more runtime monitors
• Peer review
New Mazes

• 3 new mazes
• Brings total to m1-m9
• Acceptance tests
Everything up-to-date

• Code
  – Follows good style
  – Traces to documentation
  – Compiles warning-free with flags from Project 8

• Documentation
  – Requirements, input/output/internal variable table, statechart are latest version
  – Full traceability

• Unit tests
  – 100% transition and branch coverage, data coverage
  – Trace to latest state chart
More Invariants

• tick_monitor
  – Invariant: between calls to tickInterrupt, there shall be at most one call to each of poseInterrupt, orientationInterrupt, visitsInterrupt, and bumpInterrupt
  – tickInterrupt happens once after every call to moveTurtle, so this invariant effectively checks that moveTurtle does only one of each action per call
More Invariants

• wall_monitor:
  – Invariant: turtle checks bumped(…) for edge some time before moving through that edge
  – A proxy for not moving through walls
  – Monitor should keep track of calls to bumped(…) (but can limit how many it keeps track of)
  – Subtleties in implementation; project page gives details/hints
More Invariants

• face_monitor:
  – Invariant: for any call to bumped(x1,y1,x2,y2), the turtle shall be facing the wall segment with endpoints (x1,y1) and (x2,y2)
  – Remember that the wall segment with endpoints (x1,y1),(x2,y2) and the wall segment with endpoints (x2,y2),(x1,y1) are geometrically the same
Invariant Violations

• You are graded on the correctness of your runtime monitor
• Invariant violations are OK (but you are asked to make note of them in your write-up)
• This means you have to make sure that any violations you see are due to your turtle code and not due to bugs in your monitor
Extra Credit

• Turtle runs invariant violation-free on all mazes (points awarded per invariant)
• Implement additional invariants:
  – Write a more formal definition of the invariant in your writeup
  – Implement it
  – Yet more points for running violation-free
Peer Reviews

- Groups assigned on Canvas
- Assign scribe, leader; follow guidelines
- Focus on runtime monitor implementations
- Do a check for code style/traceability in turtle code
Project Questions?
HW 36 Brainstorming

• Pick a system from the back of your sheet and:
  – State what you think the most important security risk is to do with one of {confidentiality, integrity, secrecy}
  – If you can’t come up with one, ask the audience and come up with a mitigation

• OK to use ideas that come up today when writing up your HW for Sunday, but state “this idea came up in recitation”