Updates

• Homework:
  – Homework #2 graded on canvas
    • Feedback will be in the comments of your submission
  – Homeworks #3,4 due @ 11pm this Sunday 1/28
    • Turn both in to the HW0304 spot on Canvas
    • Fill out administrative (hours/problems) survey on canvas to receive full credit
Updates

• Projects:
  – Project 2 due tonight @ 11PM
  – Project 3:
    • Released
    • Due 2/2 @ 11pm
  – Expect projects hand-in every week, due Fridays
Updates

• Be sure to fill out question sheet to receive attendance points today!
• Reminder to use staff email and office hours for questions
  – We don’t check “canvas messages”
  – Emails ece642-staff@ instead of individual instructors
  – Office hours are posted as a canvas announcement
    • Watch for location changes
Today

• Project #2 review
• Project #3
• Homework presentations
Project #2 Questions?

• You’re graded on process and effort
• You do not have to have a successful implementation
• Please follow the time limits
  – 1hr for first implementation attempt
  – 2-6 hours for clean up
  – Up to 1hr for second implementation attempt
Project 3

- Change code style according to a checklist
- Asks you to use a physical copy of the checklist
  - You can pick up a copy today
- Reflect on the checklist vs changes you made in Project 2
- Extra credit: make a maze generator
1) All the code is in a single .epp file (this is for academic project management)
2) All indentation, variable naming, and organizational style is consistent
3) Code is commented, including author comment at beginning of file
   At a minimum, comment each function: purpose, inputs, outputs, saved internal state
4) Spaces used instead of tabs for indentation
5) Every variable has a meaningful name that requires no or minimal explanation
6) Variables and procedures have minimum scope
   Defining at smaller scope (such as within a ”{}” block) is optional
7) Most variables are automatic; local static only used when required
8) All variables use strongest and simplest appropriate type, with no floats
   Add extra typedefs at your discretion
9) All base types are from types.h
   Variable size is at your discretion within reason
10) Typedefs are used to define structs and enums
11) Geometric pairs (e.g. (X,Y)) are coupled using typedefs
12) #define is not used
13) “Magic numbers” are not used
   Const used for single numeric values and enum for sets of related values
14) Switch statements are used to decide among enum values rather than if/else if
15) Every switch statement has a default clause that activates ROS_ERROR
16) All conditionally executed sets of statements are enclosed by “{}”
17) No copy-pasted code is present
   Blocks of code that have the same functionality are factored out into modules
18) Code has SF of 10 or less
   Most complex routine: #globals: ___ SCC*: ___ SLOC/20*: ___ ⇒ SF*: ___
19) No bit-wise operations are used for math (e.g., use “%” instead of “&”)
20) Math is not performed on enum values

OTHER NOTES:
Project #3 Questions?
Questions?
Student Presentations