

Secret Agent Man (Sam)

Draft

Capstone Python Project

CSSE 120, Introduction to Software Development – Robotics
Fall term, 2012-2013

Your robot is a Secret Agent Man (Sam) who gathers secrets by:

- *Finding various Bond Girls – Anya Amasova, Wai Lin, Honey Ryder and others. Each Bond Girl has a specified way to find her, as described below.*
- *Receiving and sometimes sending, numbers via its IR receiver and sender. The Bond Girl will send the signal (and sometimes receive a signal) according to a protocol specified in another document.*

You score points for the secrets that your robot gets, as well as for other features that you implement, as follows:

- **You score on a feature can range from 0 (*not done*) to 5 (*exemplary*).** Generally speaking:
 - 5 points: Does the feature in as interesting, powerful and user-friendly way as one could imagine.
 - 4 points: Does the feature in an interesting, powerful and user-friendly way, but one could go a bit further with the feature.
 - 3 points: Does the feature in a reasonably interesting, powerful and user-friendly way, but could be extended significantly.
 - 1 or 2 points: Does the feature in a barebones way.
 - Sum the points you earned in the above four categories
 - Multiply that sum by your Contribution Multiplier – your instructor’s judgment of the degree to which you contributed to your team (100% for appropriate contributions, less for less than appropriate contributions).
- Your score is a combination of the weighted sum of features, ***along with scores for quality of code, team deliverables and individual deliverables. The non-features score can raise or lower your features-score.***
- ***Highlighted features MUST be attempted by the team.***
- ***Each individual must do at least one “complicated” chunk of work*** – something whose weight is at least 4. Each individual must contribute to the GUI significantly in their own file.
- ***The program must be able to run from ANY of your computers*** (except that you do not need to be able to connect to a real robot, if you are having Bluetooth troubles).

For any feature, if you have any doubts about what that feature requires, just ask.

In the following, the “finds ...” feature can be earned independently of the “gets the secret...” feature, and vice versa.

Secret Agent Man (Sam) features:

1. [Weight: 3] Sam finds Anya Amasova, via **teleoperation**.
[Weight: 1] Sam gets the **secret: A single number**.
2. [Weight: 4] Sam finds Mary Goodnight, by **following move/turn instructions** given by the user at run-time. Your program must allow the user to enter how far and how fast Sam is to move and/or turn – no teleoperation. The best solutions will include multiple algorithms for the actions and will investigate the accuracy of each.
[Weight: 4] Sam gets the **secret: A sequence of numbers** that starts with the START-MESSAGE number and continues until an END-MESSAGE number is sent.
3. [Weight: 3] Sam finds Fiona Volpe, by **going forward and stopping when either**:
 - a. Sam **bumps** into something she has placed in Sam’s path, or
 - b. Sam **reaches a black line** that she has drawn across his path.
 [Weight: 6] Sam gets the **secret: A sequence of numbers, decoded** into a message per the protocol you are given.
4. [Weight: 6] Sam finds Wai Lin by **following a curvy black line**, stopping when Sam bumps into something. The best solutions will be able to follow “difficult to follow” lines.
[Weight: 3] Sam gets the **secret by whispering the magic word (a single number)** and hearing the sequence of numbers, decoded into a message that Wai Lin sends in response. The user specifies the magic word at run-time.
5. [Weight: 6] Sam finds Jinx by **following a curvy wall**, stopping when Sam crosses a black line. The best solutions will be able to follow “difficult to follow” walls and even navigate around small obstacles.
[Weight: 6+] Sam gets the **secret by whispering and hearing a sequence of numbers per a specified protocol**.
6. [Weight: 6+] Sam finds Pam Bouvier, by **visiting a sequence of waypoints** (x/y coordinates, with the origin/axes specified by Sam’s current position and direction) specified by the user at run-time. The best solutions will deal with obstacles.
[Weight: 12+] Sam gets the **secret by offering Rogerian psychotherapy, ala Eliza** (<http://en.wikipedia.org/wiki/ELIZA>).
7. Sam finds Honey Ryder, by **using the camera**. The best solutions are successful even if Honey takes evasive action. The best solutions deal with the possibility that Honey is partially obscured.
[Weight: 6] Sam gets the **secret by singing, dancing and displaying a light show** (and then hearing the single number that Honey says if she likes the show). Honey particularly likes shows that are composed by Sam himself and/or are choreographed, and shows in which the three components (sing, dance, lights) can be concurrent or not (at her whim). She also likes Sam’s built-in demos.

Other features:

8. [Weight: 1+3 for a and b, those are the only required ones here] **Sam's display:**
- A brief description (possibly fanciful) of this project, along with the course name and term.
 - For each team member: her name, a short (fictitious if you like) bio, a list of the main features she was responsible for, and (**updated at the end of each Sprint**) the total person-hours she spent on the project during the Sprint.
 - Information about Sam's internal state (battery charge, etc).
 - Information about what Sam is currently sensing and, where appropriate, the history of such sensing. In particular: The current camera image, and an interpretation of the current/past images (e.g., Sam is currently seeing Honey Ryder, or her conch, or ...).
 - A map of the world per the what Sam has seen on this run.
9. [Weight: 9] **Sam's user interface:**
- How well does it obey principles like those at:
 - Ben Schneiderman's ***Eight Golden Rules of User Interface Design*** at: faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html
 - Apple's ***principles of IOS development*** at: developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Principles/Principles.html
 - The variety of TYPES of interface widgets demonstrated. (The more, the better.)
 - Does the interface make good use of files? Of the Internet?
 - In your instructor's subjective opinion, is the interface visually attractive?
 - In your instructor's subjective opinion, is the interface user-friendly in its operation?
10. [Weight: 10+] **Sam's other hardware:** Does Sam use external sensors? Motors/servos? For interesting purposes? Does Sam's control use multiple cores?
11. [Weight: ??] **Sam's ... [You suggest something interesting!]**