

Talk To Me

Capstone Python Project

CSSE 120, Introduction to Software Development – Robotics
Fall term, 2011-2012

- **Highlighted features are required.**

You earn points for non-highlighted features *only* if you complete **all** the highlighted features successfully.

- Generally speaking, “successful” means at least 3 of 5, but note that most highlighted features are designed to be all-or-nothing.

- **You score on a feature can range from 0 (not done) to 5 (exemplary).** For the more ambiguous features (e.g. follow an *arbitrary* line ...), *exemplary* means that you can do that feature at a very high level, while a 1 (*needs improvement*) would be the score for the simplest version of that feature.

- **Available points are:**

- **Features implemented:** **XXX** points available, but capped at 800
 - More points increase your fame but not your total score
- **Quality of code:** 150 points available, but capped at 100, and as much as *minus 150* for poor quality.
- **Team deliverables (other than code, but including documentation among other items):** 150 points available, but capped at 100, and as much as *minus 150* for poor deliverables.
- **Individual deliverables (other than code, but including documentation among other items):** 150 points available, but capped at 100, and as much as *minus 150* for poor deliverables.

- **Your score (out of 1,000) is computed by:**

- 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)

Since the score for Features is capped at 800 and each of the other three categories is capped at 100 points, **the maximum possible score is 1,100 of 1,000 (i.e., 110%).**

Important: Note that if your team scores 1,400 on Features (a HUGE score), but the other three items are terrible, your score is a $(800 - 150 - 150 - 150) = 350$. Since that is of 1000, it is a clear F.

For those interested, the last page of this document provides a breakdown of the points by subcategory.

Start and Quit: *The user can, via the graphical user interface (GUI), for her robot:*

Features (functionality)	Weight	Exemplary		Satisfactory	Needs improvement		Not done	Weight x Score
		5	4	3	2	1	0	Points
1. Enter (and later use) the port number to which to connect	1							
2. A port number is stored in a file and displayed (and used) as the default. Program allows the user to set this default.								
3. Connect via the current port number (and then possibly do other actions)	1							
4. Disconnect (and then possibly do other actions)	1							
5. Quit the program gracefully	1							
6. Switch between modes (passive, safe and full)	1							
7. Reset the sensors in a way that is generally successful	2							
8. Choose a demo and start that demo running	2							
9. Stop the demo that is running	2							
10. Stop the operation that is currently running (exemplary requires this ability for ANY operation)	9							
11. Pause/resume the operation that is currently running (exemplary requires this ability for ANY operation)	9							

Total possible for Start and Quit: 25 + 120 = 145 points (required features, additional features)

Display Project Information and Robot's Current State: *The user can, via the graphical user interface (GUI), easily see information about the project and her robot's current state:*

Features (functionality)	Weight	Exemplary		Satisfactory	Needs improvement		Not done	Weight x Score
		5	4	3	2	1	0	Points
12. Name of this term's project (<i>Talk To Me</i>)	1							
13. Course name and current year and term	1							
14. Brief description of the project (you can quote from the description we provided)	1							
15. Names of the authors of the program, each with a short (possibly fictitious) bio	2							
16. For each author, a short paragraph describing the main features for which that author was the technical lead	2							
17. Total person-hours placed into the project through Sprint 1; through Sprint 2; through Sprint 3; through the entire Project.	1							
18. Anything else interesting about the project	2							
19. Current port number	1							
20. Current mode (off, passive, safe or full)	1							
21. Current values of all the sensors (cliff, LEDs, music playing, XXX, demo running GREEN IS NOT RIGHT YET, WILL FIX.	6							
22. Number of the demo running (or an indication that no demo is running)								
23. Position (x, y, direction) of itself. Odometry error?	6							
24. Position (x, y) of Gamma-7	6							
25. Current and past talk (NO, IN TALK SECTION?)	4							

Total possible for Display Project Information and Robot's Current state: 40 + 130 = 170 points (required features, additional features)

Tele-operation: *The user can, via the graphical user interface (GUI), make her robot:*

Features (functionality)	Weight	Exemplary		Satisfactory	Needs improvement		Not done	Weight x Score
		5	4	3	2	1	0	Points
26. Have a user-specified linear speed to be used in subsequent movement	3							
27. Have a user-specified angular speed to be used in subsequent movement	2							
28. Move linearly (forward/backward) at the current linear speed until the user tells the robot to stop	3							
29. Spin at the current angular speed until the user tells the robot to stop	2							
30. As in the two previous items, but with linear and angular motions concurrently	3							
31. Move linearly (at the current speed) a user-supplied distance (and then stop)	3							
32. Spin (at the current speed) a user-supplied angular distance (and then stop)	2							
33. As in the two previous items, but with linear and angular motions concurrently (stopping each when its distance limit is reached)	3							
34. Move linearly a user-supplied time (and then stop)	3							
35. Spin a user-supplied time (and then stop)	3							
36. As in the two previous items, but with linear and angular motions concurrently (stopping each when its time limit is reached)	3							
37. Modify its linear/angular speed, while the robot is moving (exemplary requires the ability to do this during any of the above types of motion)	6							
38. Modify the requested distance (and time, if implemented) the user asks the robot to travel, while the robot is moving	3							
39. Turn on or off a stop-when-bumped-while-moving option (exemplary requires the ability to do this during any of the above types of motion)	6							
40. Move to a user-specified position (relative to the robot's initial placement)	6							
41. Move to a user-specified direction (relative to the robot's initial placement)	3							
42. Do, via a single control, any of several interesting user-specified movements (exemplary requires a large selection of choices available)	6							
43. Do other interesting remote control of movement (score depends on difficulty and creativity)	6							

Total possible for Tele-operation: 75 + 255 = 330 points (required features, additional features)

Finding Gamma-7 (autonomously): *The user can, via the graphical user interface (GUI), make her robot (Beta-12) move to Gamma-7¹, given that Gamma-7 is:*

Features (functionality)	Weight	Exemplary		Satisfactory	Needs improvement		Not done	Weight × Score
		5	4	3	2	1	0	Points
44. Within 15 feet straight ahead from Beta-12's current direction	3							
45. On the circumference of a square whose diameter is 10 feet and whose center is Beta-12's current position	3							
46. As in the previous item, but <i>within</i> the square (not necessarily on its circumference)	3							
47. As in the previous item, but for a square whose diameter is 20 feet	3							
48. As in the previous item, but for a square whose diameter is N feet, where the user enters N	3							
49. Somewhere along a path that the user supplies by giving way-points (that is, x/y coordinates relative to Beta-12's initial placement), given that there are no obstacles on that path (exemplary requires a complicated path)	6							
50. As in the previous item, but there may be obstacles on the path (exemplary requires several obstacles)	6							
51. Somewhere along a wall of the classroom	3							
52. Somewhere in the hall outside the classroom	6							
53. Somewhere along a given, oval-shaped black line	3							
54. Somewhere along a given, more-curved black line	6							
55. Somewhere along an arbitrary line (which you are NOT given in advance), with the following restrictions on the line: it is reasonably dark on a reasonably light surface, it has no intersections, no sharp turns, and is somewhere between 1 and 12 inches wide. You can score partial points if your robot is able to do some lines but not others, and you can do a reasonable amount of calibration before your run.	9							
56. As above, but the line has intersections.	6							
57. Within 15 feet straight ahead sometime within the next 30 seconds (for this item, Beta-12 must find Gamma-7 soon after Gamma-7 is placed)	3							

Total possible for Finding Gamma-7: 75 + 240 = 315 points (required features, additional features)

¹ All movement will start from a known point near the middle of the classroom. All movement must complete within a reasonable length of time and within a reasonable distance from Gamma-7. Exemplary generally requires ending within 6 inches from Gamma-7. Beta-12 can recognize Gamma-7 by hearing her say a message that includes an encoding of "I don't love you anymore" – no other robot will be saying that special message.

Serenading Gamma-7 (via music or lights or dance): *The user can, via the graphical user interface (GUI), make her robot:*

Features (functionality)	Weight	Exemplary		Satisfactory	Needs improvement		Not done	Weight × Score
		5	4	3	2	1	0	Points
58. Play whatever note the user specifies for a user-specified duration (exemplary requires allowing durations longer than 4 seconds)	2							
59. Play a simple song of your own choosing	3							
60. Play an <i>interesting</i> song of your own choosing	2							
61. Play any of a selection of songs from which the user chooses	3							
62. <i>Compose</i> and play its own <i>interesting</i> songs (exemplary requires <i>genuinely</i> interesting)	6							
63. Set its LEDs to whatever state the user specifies	2							
64. Perform a simple LED light-show of your own choosing	3							
65. Perform an <i>interesting</i> LED light-show of your own choosing (exemplary requires <i>genuinely</i> interesting)	2							
66. Perform any of a selection of LED light-shows from which the user chooses	3							
67. <i>Compose</i> and perform its own <i>interesting</i> LED light-shows	6							
68. Perform a simple dance of your own choosing	2							
69. Perform an <i>interesting</i> dance of your own choosing	2							
70. Perform any of a selection of dances from which the user chooses	3							
71. <i>Compose</i> and perform its own <i>interesting</i> dances (exemplary requires <i>genuinely</i> interesting)	6							

Total possible for Serenading Gamma-7: 50 + 175 = 225 points (required features, additional features)

Talking to Gamma-7: *The user can, via the graphical user interface (GUI), make her robot:*

Features (functionality)	Weight	Exemplary		Satisfactory	Needs improvement		Not done	Weight × Score
		5	4	3	2	1	0	Points
72. Ask Gamma-7 to “Talk to Me”	3							
73. Indicate whether or not Gamma-7 is talking (assuming that Gamma-7 is within hearing range)	3							
74. Send and display-as-sent a single-byte message to Gamma-7, chosen by the user from the list of single-byte messages. For exemplary, you must encode messages per a standard that your instructor supplies.	5							
75. Receive and display-as-received a single-byte message from Gamma-7, chosen by Gamma-7. For exemplary, you must decode messages per a standard that your instructor supplies.	5							
76. Send and display-as-sent a multi-byte message to Gamma-7, chosen by the user. For exemplary, you must encode messages per a standard that your instructor supplies.	12							
77. Receive and display-as-received a multi-byte message from Gamma-7, chosen by Gamma-7 (via your instructor). For exemplary, you must decode messages per a standard that your instructor supplies.	12							
78. Talk with Gamma-7 via a standard that you design and implement, that meaningfully <i>augments</i> the instructor-supplied standard	12							
79. Talk more meaningfully with Gamma-7 via a <i>protocol</i> that you design and implement, on both your robot and on Gamma-7.	24							
80. Be a Rogerian therapist for Gamma-7, ala Eliza (google for <i>Eliza therapist</i>).	40							
81. <i>Win back Gamma-7’s heart</i>	<i>Priceless</i>							

Total possible for Talking to Gamma-7: 80 + 500 = 580 points (required features, additional features)

User interface: *The user interface allows the user to:*

Features (functionality)	Weight	Exemplary		Satisfactory	Needs improvement		Not done	Weight x Score
		5	4	3	2	1	0	Points
82. Do the basic actions via a reasonable GUI	6							
83. Determine that the user interface is <i>visually attractive</i> . (This will be your instructor's subjective opinion. Be forewarned that <i>exemplary</i> will be hard to earn.)	6							
84. Determine that the user interface <i>works nicely</i> , that is, how <i>easy is it to use</i> the interface without instruction? how much <i>effort is required</i> to do actions? is it easily <i>extended?</i> and so forth. (This will be your instructor's subjective opinion. Be forewarned that <i>exemplary</i> will be hard to earn.)	6							
85. Uses multiple screens consecutively in a meaningful way	3							
86. Uses multiple screens concurrently in a meaningful way	3							
87. Allow control of parts of the interface via the keyboard	3							
88. Allow devices other than the mouse to control the robot (joystick, external touch screen, wii mote, ...)	6							
89. Store and retrieve the conversation codes in files that the user could edit	3							
90. Store and retrieve songs in files that the user could edit	3							
91. Store and retrieve light shows in files that the user could edit	3							
92. Store and retrieve other interesting, relevant information in files that the user could edit	3							
93. Do actions with a nice, <i>interesting</i> user interface beyond buttons (LOTS of possibilities here – menu's, drop-down's, sliders, progress bars, ...). Your score will depend on how nice and how <i>interesting</i> your user interface is, in your instructor's subjective opinion. Generally speaking, each new TYPE of user interface widget earns you points, while repetitions of widgets of the same type do not. Also, the above items do not count toward this one (no double-counting).	12							
94. Do other <i>interesting</i> things (not scored in the above) related to the user interface	6							

Total possible for User Interface: 90 + 225 = 315 points (required features, additional features)

Hardware: *Your robot makes use of hardware that you ADD to the ROBOT:*

Features (functionality)	Weight	Exemplary		Satis- factory	Needs improvement		Not done	Weight x Score
		5	4	3	2	1	0	Points
95. Camera: Your user interface shows what it sees	12							
96. Camera: Your user interface shows a processed form of what it sees (e.g. blobs)	12							
97. Camera: Your robot uses it to do something interesting that has no direct relevance to Gamma-7	12							
98. Camera: Your robot uses it to do something directly relevant to Gamma-7, e.g. finding her	12							
99. Camera: ??	12							
100. Other sensor(s): Your user interface can display its reading	6							
101. Other sensor(s): Your robot uses it to do something interesting that has no direct relevance to Gamma-7	6							
102. Other sensors: Your robot uses it to do something directly relevant to Gamma-7	6							
103. Other sensors: ??	6							
104. Motor(s): Your user interface can make it act (operate)	6							
105. Motor(s): Your robot uses it to do something interesting that has no direct relevance to Gamma-7	6							
106. Motor: Your robot uses it to do something directly relevant to Gamma-7	6							
107. Motors: ??	6							
108. Other hardware:	?							
109. Other hardware:	?							
110. Other hardware:	?							
111. Other hardware:	?							

Total possible for Hardwar: 0 + 540 = 540 points (required features, additional features)

Your Ideas: *Suggest your ideas to us, we'll tell you whether or not they earn points:*

Features (functionality)	Weight	Exemplary		Satisfactory	Needs improvement		Not done	Weight x Score
		5	4	3	2	1	0	Points
112.	?							
113.	?							
114.	?							
115.	?							
116.	?							

Highlighted features are required.

Page 11

Total possible for Features: 500 points on required features, plus **970** points on additional features
= **1,470** points (*but limited to a maximum of 900 for Features – more is great, but won't increase your score*)

Other required team deliverables:

Deliverable	Weight	Exemplary		Satisfactory	Needs improvement		Not done	Weight x Score
		5	4	3	2	1	0	Points
117. Screen Sketch	1							
118. Release Plan for Sprint 1	3							
119. Release Plan for Sprint 2	3							
120. Release Plan for Sprint 3	3							
121. Structure Chart or other document that organizes the code for Sprint 1	1							
122. Structure Chart or other document that organizes the code for Sprint 2	1							
123. Structure Chart or other document that organizes the code for Sprint 3	1							
124. Final version of code: is decomposed into functions in a reasonable way	6							
125. Final version of code: Every module (file) has a comment at the top that lists the author(s) and briefly describes that module	1							
126. Final version of code: Every function has a doc-comment that says what that function does	6							
127. Final version of code: All code meet the standards imposed by <i>Source ~ Format Code</i> (control-shift-F) in Eclipse	3							
128. Final version of code: no magic numbers	1							
129. Final version of code: functions are reasonably sized (generally 5 to 20 lines of code, but exceptions are quite possible)	1							
130. Final version of code: there is one line (no more and no less) between each function definition	1							
131. Final version of code: meets the other coding standards that we have demonstrated throughout	1							

Total possible for Team Deliverables (except code): 65 points

Total possible for Quality of Code: 100 points

Team total: _____ of XXX.

Other required individual deliverables:

For _____

Deliverable	Weight	Exemplary		Satisfactory	Needs improvement		Not done	Weight × Score
		5	4	3	2	1	0	Points
132. Task List for Sprint 1	1							
133. Task List for Sprint 2	1							
134. Task List for Sprint 3	1							
135. Code in “your” module during and at end of Sprint 1: Every function has a doc-comment that says what that function does								
136. Code in “your” module during and at end of Sprint 2: Every function has a doc-comment that says what that function does								
137. Code in “your” module during and at end of Sprint 3: Every function has a doc-comment that says what that function does								
138. Code in “your” module during and at end of Sprint 1:								
139.								
140.								
141. Peer evaluation for Sprint 1	1							
142. Peer evaluation for Sprint 2	1							
143. Peer evaluation for Sprint 3	2							

Total possible for Individual Deliverables: 35 points

Individual Total: _____ of XXX.

Instructor’s judgment of the degree to which you contributed to your team (your points are multiplied by this number):

Appropriately: 100% Not as well as we expect: _____

Your score = (Team Total + Individual Total) × Contribution Multiplier

= (_____ + _____) × _____

= _____ of 1,000 (but the maximum possible score is 1,100, i.e., 110%)

Other required individual deliverables:

For _____

Deliverable	Weight	Exemplary		Satisfactory	Needs improvement		Not done	Weight × Score
		5	4	3	2	1	0	Points
144. Task List for Sprint 1	1							
145. Task List for Sprint 2	1							
146. Task List for Sprint 3	1							
147. Peer evaluation for Sprint 1	1							
148. Peer evaluation for Sprint 2	1							
149. Peer evaluation for Sprint 3	2							

Total possible for Individual Deliverables: 35 points

Individual Total: _____ of XXX.

Instructor’s judgment of the degree to which you contributed to your team (your points are multiplied by this number):

Appropriately: 100% Not as well as we expect: _____

Your score = (Team Total + Individual Total) × Contribution Multiplier

= (_____ + _____) × _____

= _____ of 1,000 (but the maximum possible score is 1,100, i.e., 110%)

Other required individual deliverables:

For _____

Deliverable	Weight	Exemplary		Satisfactory	Needs improvement		Not done	Weight x Score
		5	4	3	2	1	0	Points
150. Task List for Sprint 1	1							
151. Task List for Sprint 2	1							
152. Task List for Sprint 3	1							
153. Peer evaluation for Sprint 1	1							
154. Peer evaluation for Sprint 2	1							
155. Peer evaluation for Sprint 3	2							

Total possible for Individual Deliverables: 35 points

Individual Total: _____ of XXX.

Instructor's judgment of the degree to which you contributed to your team (your points are multiplied by this number):

Appropriately: 100% Not as well as we expect: _____

Your score = (Team Total + Individual Total) × Contribution Multiplier

= (_____ + _____) × _____

= _____ of 1,000 (but the maximum possible score is 1,100, i.e., 110%)

Other required individual deliverables:

For _____

Deliverable	Weight	Exemplary		Satisfactory	Needs improvement		Not done	Weight x Score
		5	4	3	2	1	0	Points
156. Task List for Sprint 1	1							
157. Task List for Sprint 2	1							
158. Task List for Sprint 3	1							
159. Peer evaluation for Sprint 1	1							
160. Peer evaluation for Sprint 2	1							
161. Peer evaluation for Sprint 3	2							

Total possible for Individual Deliverables: 35 points

Individual Total: _____ of XXX.

Instructor's judgment of the degree to which you contributed to your team (your points are multiplied by this number):

Appropriately: 100% Not as well as we expect: _____

Your score = (Team Total + Individual Total) × Contribution Multiplier

= (_____ + _____) × _____

= _____ of 1,000 (but the maximum possible score is 1,100, i.e., 110%)

For those interested, here is a breakdown of the points:

Category	Points available from high-lighted features (team must complete these to get other points)	Points available from additional features (but total from all features is limited to 900)	Total points available
Remote Control	210	100	310
Play music	90	130	220
Decode GCD messages	60	15	75
Delivery	60	525	585
User interface	80	200	280
Total from Features:	500	970	1,470 (but limited to a maximum of 900)
Process: Team deliverables	65	—	65
Process: Individual deliverables	35	—	35
Total from Process:	100	—	100
Quality of Code:	100	—	100
Total:	700	970	1,670 (but limited to a maximum of 1,100 , i.e. to 110%)

So:

- A team that is *perfect* on all the high-lighted features, process and code, but does nothing else, scores 700 points, i.e., a bottom C.
 - High-lighted features must be completed (not necessarily to perfection) before any additional points can be earned from Features.
 - It is pretty easy to earn 100 points of 100 for process.
 - Code that is poorly documented and poorly structured, but otherwise OK, earns only 40 points of 100, for quality of code.
- Each letter grade increase (C to B, B to A, A to A+) requires about 7 additional “easy” features or fewer additional hard features. Note: “easy” is relative to the others, which range from hard to impossible.
- A team that is *perfect* on all the high-lighted features, process and code, and does all 13 of the “easy” additional features, scores 900 points, i.e., a bottom A.
- Getting to 100% or 110% requires doing some hard features.
- A team that earns the maximum possible on Features (900), and gets 50% of the Process and Quality of Code points, earns 100%. Earning 110% requires perfection on Process and Quality of Code, as well as maxing out on Features.