

Introduction AI B: practical

Results until 24-05-2018

Respondents (n) 72

Total average

Below are the total averages of all evaluations taken for this course. These averages are composed of all results on all questions, with the exception of the questions with the scales "Yes / No" and "Open question", and questions in which the set of questions states that they may not be included in the average.



Average per question

Below are the averages of all questions from all evaluations taken for this course.

 De samenwerking met mijn medestudenten heeft bijgedragen aan verdieping van de leerstof

Neutral (3.2)

Disagree to agree | 1.1σ | n 72

8 x Strongly disagree

12 x Disagree

18 x Neutral

29 x Agree

5 x Strongly agree

 The assignments helped me to process the material.

Neutral (2.9)

Disagree to agree | 1.1σ | n 72

6 x Strongly disagree

23 x Disagree

19 x Neutral

19 x Agree

5 x Strongly agree

 The assignments contributed to achieving the learning objectives.

Neutral (2.9)

Disagree to agree | 1.1σ | n 72

6 x Strongly disagree

24 x Disagree

16 x Neutral

20 x Agree

6 x Strongly agree

 The practical meetings were worthwhile.

Disagree (2.4)

Disagree to agree | 1.1σ | n 72

15 x Strongly disagree

27 x Disagree

16 x Neutral

11 x Agree

3 x Strongly agree

 The learning objectives (what you should know and be able to do by the end) of the course were clear to me.

Disagree (2.4)

Disagree to agree | 1.1σ | n 72

18 x Strongly disagree

25 x Disagree

13 x Neutral

14 x Agree

2 x Strongly agree

 The coherence between the different components of the course is

Neutral (3.0)

Very little to very much | 1.0σ | n 68

5 x Very little

14 x Little

23 x Neutral

25 x Much

1 x Very much

 Studying the course materials was necessary to successfully complete the course.

Neutral (2.7)

Disagree to agree | 1.2σ | n 72

15 x Strongly disagree

21 x Disagree

15 x Neutral

16 x Agree

5 x Strongly agree

The organisation (e.g. the planning, scheduling, method of information provision) of the course was good.

Strongly disagree



Disagree to agree | 0.8σ | n 72

(1.4)



50 x Strongly disagree 16 x Disagree 3 x Neutral 2 x Agree 1 x Strongly agree



The information provided before and during the course was sufficient.

Disagree to agree | 0.9σ | n 72

Disagree (1.7)



40 x Strongly disagree 20 x Disagree 9 x Neutral 3 x Agree 0 x Strongly agree



The time that I spent on this course was less/more/equal to the number of course credits (EC).

Less to more | 0.9σ | n 72

More (3.5)



2 x Much less 3 x Less 36 x Average 19 x More 12 x Much more



My general opinion of the course is

Bad to good | 0.9σ | n 72

Bad (2.0)



22 x Very bad 36 x Bad 8 x Neutral 5 x Good 1 x Very good



My general opinion of the lecturer(s) is

Bad to good | 0.9σ | n 72

Bad (2.3)



17 x Very bad 20 x Bad 31 x Neutral 4 x Good 0 x Very good



The lecturer had a good command of English.

Disagree to agree | 0.9σ | n 72

Neutral (2.7)



9 x Strongly disagree 17 x Disagree 35 x Neutral 11 x Agree 0 x Strongly agree

Open questions

Below are the results of each open question of this subject.

 What do you think are the strong points of the course?

n 1

- 1.
- 2.
- 3.
4. Some of the TAs were really good at helping
5. Fun assignment, good to learn another programming language as well. Nice to bring IAIA into practise a bit
6. No
7. You learn a lot about Python in a small time
- 8.
- 9.
10. it is nice that instead of an exam, the final assignment was a contest
- 11.
12. I sort of enjoyed the contest, because competing against each other in this way made the assignment feel a little more like a fun game and a little less like just another deadline for a stressful course.
- 13.
14. I think the assignments revolving around pacman are a really fun way to learn more about the topics discussed in intro AI B
- 15.
- 16.
17. "Fun" practical assignment to stimulate students
18. The course forces you to actively use the theory learned in Introduction AI A. In this way you understand the material much better
19. A strong point was that you had to program in a different language that we hadn't used before.
- 20.
21. The TA's were really helpful and tried to help us as much as possible
22. It's fun
23. I really like the learning method of just throwing us at assignments and letting us figure out by ourselves what to do and how to do it. I love the idea of that and the idea of letting us solving pacman.

24. I really liked the final project at first, and I think it could have been more. I also liked learning a new programming language, but the way I did was probably not how I was supposed to; spending hours and hours every weekend instead of just following the course, like the rest.

25. The fact there is a contest which makes it a game.

26. There were some teaching assistants who knew quite well what they were doing. It is easy to get a good grade if you just do what you need to do.

27. - Pacman is a nice way to encourage people to program (instead of writing boring programs)

28. Learning about python was nice. It was also nice to learn it by working on pacman, making it more fun and thus more engaging and easy to learn

29. None

30. The game-like setup of doing it yourself. It also is quite fun.

31. The general idea of the course: making your own agent, especially the final contest

32. Learning python was nice

33.

34.

35. The virtual world of Pacman is fun.

36. Fun way to learn about agents and search algorithms. Collaboration was rewarded.

37.

38. it's very practical

39. It built further upon Intro to AI A, so the theory was all known

40.

41. pacman is kinda fun

42.

43. ...

44. I liked the fact that I got to learn and work with python. However that was not because of the course, I had to learn it myself.

45. New language (Python), fun assignment, one assignment through the whole course.

46. It's a really hands-on experience being able to manipulate the behaviour of an agent in a game, which is both exciting and useful

47.

48.

49.

50.

51. Pretty fun as long as you know what you are doing

52. The strong points: the idea of developing a pacman ai to survive as long as possible is quite good. Also adding the competition element adds some sense of fun, even though it seems a bit backwards

to pit students against each other for a grade. Furthermore some of the TA's were really helpful and I was glad when I finally finished the course

53.

54. I liked that it was challenging and kind of fun in a game setting.

55. I found working with pacman really motivating. I had the feeling that (especially after the very theoretical introAI A) that I was doing something "real".

56.

57. The game you make is fun

58. The collaboration

59. I think the idea of programming pacman is good (the man concept)

60.

61. to learn programming you have to get your hands dirty. A practical part of introduction to AI is a good way to learn it.

62. You had to figure out a lot by yourself, so IF you understood it, you actually learned a lot from this course.

63. By doing you do get a clear insight in how the program works

64. The idea to build a pacman game is nice

65. Learning to solve problems in a different setting. Being thrown in the water has its positives. Implementing search algorithms in a fun way.

66. you learn a new programming language a bit

67. The fact that your optimalizing pacman is a kinda fun and the idea is nice, but that is really the only somewhat strong point of the course.

68. Implementation of the search algorithms, learning to read someone else's code

69. Some TAs were really prepared and helped us a lot

70. It introduces us to a new programming language

71. The assignments prepare you for the contest part. Also the contest is only 30% of the grade, which is really fair.

72. Some of the TA's were really helpful in guiding us through the assignments - others were not.



If you see areas that could be improved in the course, what are your suggestions?

n 1

1.

2. Explaining the assignments More TAs More information about python Less mistakes in assignments and manuals

3. Unclear explanations of the exercises, and there were mistakes in the exercises it self.

4. Okay so this course was a trainwreck. Every practical session there was a message that we should change something in the code because there were errors. The goal of the assignments were very unclear. I never understood how I should begin an assignment. There were like 3 tas and I have waited for over 40 minutes multiple times before somebody could give answers. Some TAs gave really bad advice like 'oh just redownload the files' or 'look at the slides', and that's where I waited 40 minutes for. The TAs didn't have coherent answers. The answers to the assignments posted on BB were not understandable and used stuff we never saw before. The course manuals didn't work with the given code, because it was copy pasted. The leaderboard was a disaster since people couldn't see how their agent did they might have gotten way lower scores. We didn't get a good introduction to python or any helpful lectures. And maybe one of the worst things is that I feel like I have learned nothing. I go to uni to learn stuff but this course was just a mess and I still don't know how to use python well or what was going on in all the supplementary files or the solutions to the assignment. Please please please change this course next year it was horrible.

5. - The assignments were not that clear, I almost never knew where to start and what the outcome should be. My first question to TAs was always: "What is this assignment about?" Then "What should the function return? Because I don't know the internal structure of the program so also don't know where it gets called." And then: "I have no clue how to write this down in Python (Java I probably do though)." So improve the assignment manuals to get rid of all those first questions because now after so many questions the TAs almost made my assignments sometimes. Also make more clear what the internals of the programs are: what happens when you write a certain line of code to run it, in which function does this new thing get called? - Do firstly one lecture on how to Python, now at the very end I know it a little bit but I am still not convinced about it. Only that manual at the beginning is not sufficient. - When above two things get solved maybe also the lack of TAs get solved (I know it is more than last year already and I appreciate your effort very well on getting more TAs). Now you could easily sit 30min-1 hour raising your hand and so not really make use of the workgroup hours that well.
- The grading code of the TAs is too complex: they often don't know themselves how it works or spend way too much time on understanding it. Also, this is not how a first years student would write it, so they can't use it to help us. - There is a real difference of knowledge and being able to help between TAs, so only keeping/getting the best ones would also help getting rid of the lack of TAs. Now I noticed you get to know all of them and who are your favourites and who not, so I noticed that people sometimes put their hand down when a certain TA came to their direction and after put their hand back up. The next thing I am going to do is not that professional of me, so sorry for that, but I am going to name some TAs to help you with this. Jordi, Sanne and Iskaj (and some more I don't know the name of) I loved, they were the best and helped very well. Try to get them next year as well! But Luca and Jules I would not have next year anymore, I'm sorry, because they didn't help that well. Luca always said "use the debugger" and pointed intimidatingly to your screen and was gone before you could ask how to fix that problem then which you found with the debugger or knew of yourself already. Or even I didn't know how to use the debugger at all and he didn't really help with that. For Jules, he didn't get the code he was provided with so often said he couldn't help us and we should ask another TA. Or you should look along with him (and write the same thing down). So another time: I'm sorry for naming, but I believe that is the best.

6. - Have TA's that help you - TA's that understand the problems - Communicate bugs - Make sure no bugs in the code you provide - Explain things - Make leaderboard work properly - Clear goals beforehand (e.g. what the report should contain (for contest) - Give hints on how to start with an assignment, like Berkeley has done in its code - Communicate in general

7. First, I think they should include "knowledge about programming/java" in the Prerequisites, because otherwise this course is near impossible. The lessons are very hard, especially because we got zero explanation beforehand and barely any support from the assistants. There were too little assistants walking around for the number of questions from students (also due to lack of explanations), and those that were there to help all gave different techniques, which meant we had to change our code a million times because of all the inconsistent suggestions. The only reason we passed this course was with external help.

8. - More TA's or more TA's who can actually help - In-depth explanation of python with lectures and practical sessions - Explanation of how to call specific function needed to make assignments - More

clear manuals explaining what we exactly is what we should do - No errors in program written by teachers - Make sure that the leaderboard actually works in time and when some students who handed in their code didn't get a score because of errors, correspond back to them to make them aware that something went wrong and what - Making everyone able to see leaderboard scores, not just when you are at ru - Better explanation of what should be written for the report for the contents - When the course has been given constantly bad feedback by students, it would be nice if the teacher is aware of that and not say that the course went really well when in reality that was not the case at all - Last lecture after contest, handing in the contest was completely a waste of time

9. Please check code before distributing it to the students. Let us know immediately if there's a bug in the code, else we will spend hours on an assignment that could not be done appropriately anyway

10. clearer manuals, less errors in the given files for the assignments, more explanation (maybe a lecture) about Python before the start of the assignments, because it was very difficult to start with the assignments with no background knowledge on python and no detailed explanation

11.

12. I had no clue what to do when just looking at the manuals. I had to google my ass off. The TA's were helpful when available, but they barely ever were available. It really felt like the course was just yelling at me: "JUST MAKE A PACMAN! JUST DO IT!", without really telling me what I should use or where to start. It was very stressful and I wish I would have been able to enjoy the course more, because from a retrospective viewpoint I really liked the assignments.

13. - Kamsteeg didn't even show up for his own course during the practical hours... It was almost as if Jordi was the teacher (he really deserves a lot of credit or maybe a little gift or something for all of the work he has done for the course). - More and better explanations could have been given in the manuals. The first exercise was very doable (the one with the supermarkets), but the next ones were very unclear in the sense of "where should I even start... and what now??". I understand that finding out yourself how the code works is a thing that belongs to programming, but this was a whole new level of discovering the code yourself. First of all, at the start of the course, we were just thrown into Python. It would be nice to have at least 1 lecture which would explain the basics of Python, or provide a series of YouTube videos with great explanation. Second of all, we had to use commands we didn't even know existed in the code! It would be nice to have a small list of functions/commands that we can use for the exercise at the end of each manual. - There were not enough (good) TA's. When having a question, there were times that we had to wait even 40 minutes or more until a TA finally came (which was obviously very frustrating). So my advice for next year: only pick TA's of whom you're sure that they can explain well and know Python.

14. The organization NEEDS to be improved. There have been a lot of bugs that either were left unsolved or were solved only a few days before deadlines. Assignment instructions more often than not needed an update due to errors/unclearness. This all made it really hard to properly focus on the assignments in the intended ways.

15. It was very unclear how to use python. Would have liked some more explanation about how to use the code or what to do.

16. - More TA's, that also know enough. Often it felt like there was only one who could really answer my questions. - The manuals need to be formatted in a clearer way, reading them was often straining and unpleasant. - Make sure in more difficult cases, that a good pseudocode is available that resembles the implementation a bit more and not something arbitrary where I must think away parts. As the resulting implementation of the code was very bad.

17. Better course materials and documentation on their python API General explanation of frequently asked questions

18. TA's should be informed better about how the exercises should be implemented, and have more time to gain an understanding of the course. The manuals should be greatly improved. They don't explain the exercises clearly, and some essential features like what a function should return are lacking. The classes and code given to the students should be much more clear. They should have

clear comments describing their uses, and/or a full explanation of the infrastructure. As most students have never seen or used Python before this in itself is not clear. In the first week the students were expected to learn Python to such a degree that they shouldn't have struggled implementing their ideas during the course. However, most of the times I felt extremely limited using Python as I clearly lacked understanding of key concepts. The workings of conversions, zip lists, and list comprehensions were completely new to me and were not explained in the first week. I would suggest extending the current explanations, and reserving more time to learn how Python and PyCharm work. Because the code and manuals were really unclear the TA's were mostly answering the same questions over and over. This took the bulk of the time during the practicals. If the code and manuals were clearer and more complete the TA's would have been less busy, and the waiting time of sometimes 15-30 minutes would have been shortened significantly. There were also some TA's that seemed to have no understanding of Python at all like Nick van der Linden and Jules van der Burgt. Some however, most prominently: Jordi Riemens, but also Sanne de Kleijn, Iskaj Janssen and Francesca Drummer were really helpful.

19. -The assignments were vague so it would have been nice if they explained it some more and maybe gave examples. -They have to come up with a better system for people with questions because sometimes you didn't get to ask your questions.

20. Planning, course structure, assignments

21. Better explanations of what is asked, better planning of the course (e.g. the mistakes that had to be corrected after prompting from students/TA's) and overall better instructions about the exercises

22. Fix the bugs. This course had more problems than assignments

23. The manual are hardly helpful ever. We had to google every single thing to understand anything or ask the TAs. If you want to keep this up you will need better manuals or waaay more competent TAs. Also there were (one particular) really bad TA, that never answered our questions. Maybe you need to better check if the TAs know what they are doing or figure out a report system?

24. This is gonna be a lot: - there were too little TA's, helping people for too long. Jordy is a great TA for instance, giving hints and then going on; he knows the code really well. Then there's TA's like Luca (!!!), Nick and Jules (so far I can think of), who did not know the code at all; they just let you copy the answers, or didn't contribute at all to your code, they just said "debug", although you already did that and that was the entire question. - the manual was horrible. The explanations were way too vague and I had no idea what I was doing until the final assignment, just because I did not learn that much from the way it was set up, AND explained. - It took a very long while for some code to get fixed on their side so we could start; it took a while for the final assignment to go online because a mistake in the deadline was made. - the final lecture was the biggest waste of time I had this year. Paul is the worst lecturer I had this year for any course, and here is why; He is not well prepared, his intonation of words is making half the room lose concentration on its own. Next to that, his personal response to questions by email and in the break always felt like you were bothering him, he doesn't seem to enjoy it. He may be a very stressed man, but this is just unprofessional.

25. The errors, more TA's, more explanations, a better bridge from 'this is a list' to 'write a depth first search algorithm'.

26. There should be more teaching assistants. There should have been a couple of lectures to get used to python. The manuals for the assignments should be clearer.

27. - Make the manuals more clear - More TA's, sometimes we waited more than an hour to get little help - There are very big differences between TA's: some of them are completely useless, as they understand little to nothing about the subject - Less bugs in the code they supply

28. Working in a pre-made program was not nice because you don't know anything about it. Also you get thrown into seemingly difficult assignments with very little python experience. There should be an introduction to python like we had an introduction to java with programming 1. Also, there were many mistakes in the course material and the necessary functions were often not covered.

29. The course was very poorly organized in every sense. The assignments were not clear, there was an insufficient number of TA's, the codes that we were given had several errors which cost us a lot of time and effort.

30. Significant lack of TA's and shortages in the manuals. During the sessions, it could take as long as 30 minutes to get a TA to help you. And many people do ask the same questions, especially the getting started questions could be tackled by using a more advanced manual. For example during the first few sessions it was very unclear how the whole architecture was, which did not help understanding the exercise. If the manual would say something like: "function x should return a list of Vectors for the path pacman has to run" that would help a lot already.

31. The manuals. Many things were just not clear enough what exactly you had to do. I would recommend a tutorial lecture at the start of the course where the lecturer could also explain what function you can use, how they work and how to use them.

32. More ta's during the lab sessions (sometimes we waited for 30 min) and it would be nice if the teacher of the course would also be present. More clearer explanation of the assignments and it would be nice to have an introduction to python at the beginning, so you can actually start off well.

33. More information, more guidance

34. The assignments were always very unclear to me. We hardly got any introduction to Python, except for the 1st assignment. I got the feeling we had to figure it out ourselves. Figuring things out is not bad, but with this little help we got (there was consistently a lack of good TA's. We've been informed that mister Kamsteeg couldn't afford more TA's and opened a discussion forum instead. However, from what I experienced, the TA's were always very busy, so I don't understand why there can't be more of them or at least take the effort to clarify the assignments?) , the gap to bridge Python was too big. Those few lines explaining the assignment were just not enough and the assignment could be split up in smaller parts, so that you have a direction to work into if you are stuck. The hints provided by some assignments were a bit vague in my opinion, they didn't help me further. Also, the whole course setup was very messy, every time there was at least another error in the code and it got announced very late. If you had already started on the new exercises I guess you were just unlucky to discover that there was a bug in it. I had to download the course files multiple times because there was some fixed version. After all, I don't have the feeling I learned or accomplished something, which I find quite unsatisfying.

35. Mistakes in the code were standard, sometimes had to be updated up to twice a week. TA's were given a non functioning answer model and therefore not able to help appropriately. There were not enough TA's. Going to a practical often meant waiting half an hour for a problem you had at home. The amount of work was not even close to the actual time required.

36. Make sure that the manuals and (especially) the source code are correct before the course starts. Make a better distribution of TA's over the groups. Schedule a lecture or something similar for each assignment period to discuss feedback etc.

37. Too little TAs TAs often couldnt help at all

38. More TA's, and also (some) better TA's. It was not uncommon to have to wait over half an hour for assistance.

39. The manuals weren't nearly as helpful, there were far too few TA's, the difficulty curve wasn't gradually increasing in difficulty but with some spikes in difficulty and then dropping to basic programming. There was no way to see whether your program would timeout because the only possible way to see that was by uploading it to the leaderboard (which only updated once a day).

40. We had to few teacher assistents

41. the lecturer mr. Kamsteeg was at times very unprofessional in his contact with the students via blackboard(most messages had very obvious spelling mistakes in them, I understand that the English of some teachers is not native speaker level but this seemed like just laziness to me, the mistakes

were really obvious typos) and via email to individual students.

42. Better inform the TAs (they often did not know how to help the students). Make sure the assignments work, there were more errors than there were total assignments. Better explain the assignments, they were often very vague, making the TAs have to explain the same simple questions very often.

43. I think there were not enough TA's. At one practical session there were only 2 TA's, which is obviously not enough for all the students, and certainly not with manuals that were not complete. I think a lot of questions were not necessary if the manual would be more clear. Also I spend way more time on this course than the number of EC's suggest. Without the help of the TA's it was really hard to find out how to do the exercises, so that cost me a lot of time at home. I asked lots of questions in the practical session but because of the lack of TA's we had to wait very long till someone could answer our question. I think a more clear manual, a bit more help with the basis of Python and more TA's can fix a lot of the complaints on this course.

44. - Make the assignments clear. For each assignment there was a very specific answer necessary which was nearly impossible to get. The entire course is just copied from Berkeley but at Berkeley they even give more background knowledge than here, at least have the decency to provide the full explanation. - Don't leave in errors and be very clear about them. There have been multiple errors which all had to be shown by students and were hardly ever communicated. Once there was the huge error with the cost-function on which a lot of student spend an enormous amount of time and when it was clear it was actually an error this wasn't even shared on blackboard. - Give the TA's more information. For starters there were hardly enough TA's to begin with, but also they had absolutely no idea what to do as in their year apparently there weren't things like a frozenset, etc. Also some of the TA's had no idea what to do and we had to wait for hours to get our questions answered. - Please just make the assignments yourself, this is lazy and ridiculous, I could just follow the online course myself (and then get a lot more information), I'm not paying tuition to Radboud to get something like this. The assignments were already impossibly hard and even harder by the 'changes to prevent us from copying online code'. Just change it altogether. - The assignments were in no way related to the course material we covered previously. The A-part didn't prepare me for the B-part at all, usually I had no idea what to do. - The assignments were way beyond our level. During programming for AI already made BFS but here in python and in this impossible environment we didn't even know where to start. - Provide a list with functions of the program. We didn't know how Python even worked before we started the course and somehow had to find out that gstate.dots.list() created a list of dots, just give us a list of commands to use. This is not something we should learn from this course. The TA's didn't know either. - Learn us how to use Python. The files with the shop were way insufficient, just give a lecture about how it works, with the classes, the list-comprehension, the not saying int, the different loops, etc. We already didn't have any lectures, that's the least that could be done. - It's ridiculous that a course is graded by assignments handed in made by two people. This means that one person could do all that work (which happened a LOT) and the other nothing while still getting a good grade. Furthermore the grading was so strict (you didn't use list-comprehension, minus 1, etc.) that the only way to actually get a good grade was by having almost an exact copy of the answers from the TA's which is ridiculous, it's promoting plagiarism.

45. The first manual didn't teach me anything about Python. Not that it contained no information, but the lay-out and structure were extremely bad. It should be rewritten. Also, give more explicit tips, e.g. tell us about gstate.dots.list(). Since we're all novices with Python, no one knew about the list() function, and it also was not clarified in gamestate.py. Also, make a good list of the functions we can use. gamestate.py with such a long file, it was hard to keep an overview. Also, the TAs didn't agree with each other. They gave conflicting information. Some TAs were straight out bad, e.g. they started explaining the whole assignment or asked 'what do you think you have to do?', instead of answering questions.

46. The manuals were really unclear about what is available for use and where to start. Also, the provided files were both very confusing, difficult to get a grasp on and had internal problems with them.

48. Less errata in the assingments

49. More TA's, less errors in the manuals, less vague manuals, less copying exercises from other universities and then editing them so they become broken

50. Better learning materials with clear instructions

51. Dont do it in pairs and provide easier assignments because I and many others just could not follow

52. I think the course is quite bad. It has an interesting, good even, premise, but the execution is quite terrible. To start we are thrown in to a totally new programming language and are expected to master it, while working in a huge code with many different packages and classes. The first exercise week was doable and seemed ok for learning the language, but the second week was an absolute terror as we suddenly had to use functions from classes we didn't need to access before. We barely understood the return types of some of these functions. An improvement would be to clear up the code and allow students to start with relatively fresh code, not cluttered by things the aren't supposed to use at that point. Look for example at the courses programming 1 and 2 for AI. They started each week with a new exercise. Now it shouldn't exactly be like that, but to start it off like that would be a very nice way to learn how to use python properly. Secondly: The code provided to us was littered with bugs and at the height we were given 2 bugfixes in one week. If the point of the given code is to provide a stable platform to build upon as a student, at least try to make it as bug free as possible. Granted people are still human, so mistakes are bound to be made. One thing that is less forgiving is the fact that the documentation was not that good. A lot of the times it was quite unclear what a certain function would do and what certain methods would return. It would have been nice to at least have a short description of what the function does and what it returns. There were some functions that had documentation, but not all. Thirdly: The range of helpfulness for TA's spanned the Mongolian empire at its height (Korea to Vienna, Austria). Also the TA's were severely understaffed, such that waiting times for students were often 15 minutes or more. I won't call anyone by name, but there were some TA's who really seemed like they didn't know what they were doing. They would try to help, thinking for a long time, finally coming up with a "Solution", which at the end didn't seem to work. Thus wasting both the time of the students, which direly needs all the time they can get, but also of the TA's who need to try to help as many students as possible. It has happened about 3 to 4 times that we got a solution from a TA's which ended up not working, sending us back to square one. On the other range of the spectrum, there were really some TA's who were top quality. They had good, sound explanations, given in a reasonable time. When students were stuck, they would give just a little bit of code as a hint to get the ball rolling. This was tremendously helpful and without those TA's we certainly wouldn't have been able to finish the assignments. Fourth(Iy?): The teacher running the course didn't seem to influence it a lot. From what we had heard the code wasn't written by him, but by one of the TA's. He wasn't even qualified to help students with Python. This is more of an opinion (which is bound to be controversial among you, because I certainly don't have all the info and only a limited view of the given teacher), but it seems quite baffling to me how a teacher who doesn't understand the language and only did a non-classical introductory talk in one of the classrooms is allowed to give this course. Most of the real work was done by the TA's, who were severely understaffed, because there was a "lack of funds". Furthermore there was even a mishap with deadlines which were actually earlier than BlackBoard announced. This sent a wave of stress down the students and was quite unpleasant, even though it wasn't THAT bad in the end. Finally: A minor issue compared to the rest, but the instructions for the final report were really unclear. There wasn't really a clear instruction posted anywhere, as far as we knew. We went to one of the TA's and asked for instructions and they showed us one of the reports from last year. They explained a bit and this was helpful, but it seems quite inefficient to make every student go to a TA for these basic instructions on a report, which just as easily could have been posted on blackboard.

53. Asking us to learn an entire new programming language from scratch with only a guide seems like quite a big ask

54. More/more helpful TA-s would have been nice. If I get a feedback that my grade is less because of "minor style issues" I would like to know what those are because without that I have no chance to learn from it and improve. Also, 100% of the grade is teamwork, so now my parter will get a very high

grade even though he didn't understand anything and I had to do every assignment. Not that I mind, but I think the grades don't reflect anything this way.

55. The course seemed a bit unorganised, there were lots of corrections both in the explanation of the assignments and the project files. The leaderboard at the contest was not so useful, and since the 0.2 second limit depends on the computer the contest is run on it was hard to know whether you can still upgrade your agent, which would improve the performance but also take more time for a step, or not. In a lot of assignment the hardest task was to figure out what the function should return.

56. 1 lecture/tutorial, 2 workgroup-meetings instead of 0 lectures, 3 workgroup-meetings. Lectures/Tutorials can be used to explain nice and special things to Python and questions can be asked for which every student would hear the answer. In the workgroups there were too few TAs for too many students for the same questions over and over again.

57. Make sure there are no errors in the assignments and explain everything better. The assignments often weren't clear, because of that we had to ask the TA for help a lot and thus there was a lack of TA's

58. Provide more information. Better assignments without errors. Provide more TA's.

59. The manuals where not clear and had errors in them. The code had errors and needed several updates. There were not enough tas and they were not all competent enough to help the students. The waiting time when you wanted to ask a question was way to long. The tas had to answer the same questions over and over, because the manual was not clear. The leaderboard needed to be fixed several times. It should hav been tested better before the contest. There was not enpugh time to get used to python. It was difficult to work with the code provided by the course

60. Some more explanation on the assignments

61. For me it was not always clear what I have to do for the assignments. Furthermore I had to google and research a lot. Even when I already had Python experience. I got a in depth knowledge of what is going on when I explained it to my fellow students. Maybe for more insights what is wanted, this course should add pre and postconditions to the functions. It was quite hard to figure out what exactly to return ... a list ... a dictionary ... just a value ... list in list in tuples ... on the other hand side this may was wanted that we experience with the already given code.

62. - A little more information/explanation could help. I, and I believe more students with me, had problems really understanding what to do for an assignment, partially because we were completely new to python. - There were a lot of errors/mistakes in the assignments and in communication. There also went a lot wrong with the final contest. I believe a lot of these mistakes could've been prevented by just looking more critically at the assignments before uploading them. - There were not enough TA's for the amount of students, so next time provide more TA's, so that we can get all the help we need.

63. - only 2 or 3 TA's had a complete understanding of the course and so were the only ones who could help us with more in depth questions, maybe find TA's that do understand python in general and have not only their own solution to the assignments but also a solution that has been discussed beforehand by all TA's so you get more useful help and no incongruent answers from the different TA's - The assignments were often worded in an unclear way. Suggestion: finalize the assignments beforehand and keep in mind we don't know what to expect of this course - Documentation in the python files was not clear, so it was really difficult to work with. Suggestion: give longer explanations for the functions, and maybe give a seperate overview of all the important classes and how they are related. - Make sure beforehand that everything works, as this avoids a lot of confusion - Maybe give 30 min tutorials when starting on a new assignment to ensure everyone knows what to do and how to start their approach.

64. It is very unclear. Not enough explanations about what to do.

65. Definitely 100% I think it would be beneficial for everyone involved to have instead of 3 lab sessions per assignment, 2 lab sessions and one lecture/tutorial. In the lab sessions 80% of the

students are sitting around, staring at their screens and waiting for the ta's to come by to explain how list comprehensions etc work. The TA's have to answer the same questions over and over again. That way no one can focus on the more essential questions. This seems extremely inefficient as most problems could have been solved with a python crash course beforehand or as I already suggested, 1 tutorial and 2 lab sessions per assignment. The years before us said that they experienced the same. Unfortunately there was no change yet.

66. organisation, the imbalance of information and ta's necessary

67. Desperate need of more TA's; Need to DOCUMENT the code!!! Why are we taught to document everything for it to be understandable but then we are given a code which is undocumented, buggy and incomplete? This is hypocritical; The whole Pacman did NOT work as real Pacman should i.e. ghosts spawn directly on Pacman or some other place and not in the central designated area; No Python introduction lecture??? Is it too much to ask for a few lectures introducing syntax of Python? Now we spent more time figuring everything on our own and in the end making many mistakes and thinking we did well. Not ok; In my opinion, A.I. part A and part B should be merged. Students could make Pacman algorithm and learn about those algorithms theoretically at the same time. First, students get to know a search algorithm then instead of those redundant exercises from A.I. part A students could implement those algorithms in Python (having in mind that some lecture was done about Python). In the end, the grade could be a multiple choice test + Pacman algorithm. I think it would be a lot cooler. I felt like no one cared about us and all the issues in the course. I would not recommend this course. In my opinion, just a bad modified copy of a good course.

68. A lot of people are very negative about the general setup and the program that we work with. I won't go into these standard complaints, but I will say some general things. I can say I have learned almost nothing from this course, outside of an introductory to python. The goal of the course is to implement the searchfunctions we have learned about in intro AI A, but in the end most time is spend understanding the whole setup of pacman. Paul Kamsteeg doesn't seem to want to abandon the program that we're using right now but I think that there are better ways to learn how to implement different kind of searches. Or we should get a general introductory and good explanation about what everything means. In the end too much time is spend trying to understand ununderstanble things without help.

69. I found this course to be really bad and almost useless. First of all, the assignments were really unclear, so the TAs had to explain every team individually what they actually needed to do, which took a lot of the solving time. Also, from what I understood, the code has been changed since last year, but the TAs were not notified in advance, so sometimes they were as confused as we were. Furthermore, the assignments had a lot of errors, a lot of new versions were published throughout the course, and also some solutions the TAs received were incorrect. The leaderboard did not work. All in all, a really poorly constructed course, I feel like I wasted my time instead of learning something.

70. - A better tutorial on how to start with Python. - Less vague assignments or more TAs, because workgroups where useless, because all TAs where constantly busy answering the same questions over and over again, a good assignment wouldn't raise the same questions for everyone.

71.

72. - Clarify the exercises more. The assignments were mostly described in 1-2 sentences, which to me is is pretty short. On top of that we had to learn Python, which we are not familiar with, and it is difficult to understand exercises with a limited proficiency in Python. - Increase the time for learning Python - we only had 1.5 week to sort of get proficient in Python, which was not enough. We started off with some simple exercises, which were okay, but then it got harder and harder and there was little support on how to tackle these exercises, which was a pity. - Maybe increase the amount of TA's? This year there were many questions asked by a lot of students, but the problem was that there were not a lot of TA's, and the TA's that were there weren't really helpful. - Make the students download the CORRECT course materials ONCE. Well, basically every week there was some kind of bug in the code (which was corrected on BB, and uploaded again) which lead to a re-update of the initial code we started off with. It is very annoying to download everything 4-5 times, or realizing that your code doesn't work as a result of using the wrong (not-updated) course materials.



Other comments and/or explanations regarding the answers you gave to specific questions.

n 1

- 1.
- 2.
3. Change this course
- 4.
5. No not really
6. This course was the worst I have experienced.
- 7.
8. Overall the course was very bad organised, full of errors and not well thought out. I hope that it will improve a lot for next year students
- 9.
- 10.
- 11.
- 12.
- 13.
- 14.
15. There were many errors and the last assignment was graded the day of the deadline for the contest. Couldn't use that for the code for the contest. It would be nice to have been able to use that as I had no clue what to do
- 16.
17. The following answers (concerning teachers) will of course make no sense as we haven't seen any teacher throughout the course
- 18.
- 19.
- 20.
- 21.
22. Not enough TA's. A lot of mistakes in the manual and codd.
23. Make no mistake, there were other really great TAs, that tried their best and were in fact really helpful. But it is really frustrating if you have to raise your hand for over half an hour and then you end up with this particular one and they just say something like its your fault if you dont understand your code/the code behind it/the debugger.
24. I don't like how the TA's didn't know the "new code" this year and then the code was super clever, so some TA's didn't know some of the syntax and what it did, so before the third excercise hour, or

even the fourth (which should be spent on the next assignment!!) was when you could start, because then they got it.

25.

26.

27. See previous comment section

28.

29. They suggested to spread between 2 rooms, but in one of them there were often no TA's present, which meant that we could either work without any possible help (which made our presence in the lab session worthless) or we could be in an overcrowded room with anyway an insufficient number of TA's.

30. already given

31.

32.

33. It is quite hard to implement what we have learned in intro to ai A when we have to do it in a language we have never used using prewritten code which is unclear and full of bugs

34. I hope the course can be improved, because with the current setup of the course, I am very unhappy. I understand it is hard to change a course while it is running, so I just hope it will be better for the people next year.

35.

36. I rated the organization of the course very low because of the bugs in the manual and project files, but these should be easy to fix.

37.

38.

39. I would've liked to see more guidance in the manual, because we received a lot of files with advanced python which we barely understood. We had to dig into these files to understand what the function we had to write got as parameters and what it was supposed to return to make sure it even ran the code.

40.

41. i still do not understand python. at all.

42.

43. The question about EC's was a bit confusing. Was it 'little - much' time spend or 'little - much' equal to the amount of EC's.

44. I think I was quite elaborate already ;)

45.

46. The amount of TAs present during the lab sessions was nowhere near enough for them to help everyone having problems, however, it might be partly fixed if the manuals for the assignments were more clear to the students.

47.

48.

49. Again, not enough TA's, bad exercises, lecturer(professor) knows less about everything than a TA
- 50.
- 51.
52. See my five pointed feedback message in the last section
- 53.
- 54.
- 55.
- 56.
- 57.
- 58.
59. The concept of the course is good. I think using the things we learnt during intro AI A to program pacman in different ways is a good way to learn those techniques. It could have been a fun challenge. Sadly the execution of the course was bad
- 60.
- 61.
- 62.
63. A note: We several times got helped by a TA whose only suggestion was to 'use the debugger.' Which in principle is nice advice, however this was the only help this TA would give us. Also when the bug was before the code we wrote and so we could not have influenced the bug. When this happened the TA after some time just walked away, mumbling something about 'trying a bit more with the debugger.' This is not what a TA should do, because the TA does not in fact help at all - the debugger does all the work. Yes using the debugger once could help you and make you understand better. After that, if it is still unclear, it is the job of the TA to help you further, NOT the debugger. Could you next year make sure TA's understand this. Another note: Because Kamsteeg only gave a 30 minute lecture at the beginning of the course, there is not a lot of information to go on for the next questions.
- 64.
- 65.
- 66.
- 67.
68. Please please, improve this course and do something about it. Paul Kamsteeg doesn't seem to take responsibility for the course and honestly as a student it feels like he is to lax to change the course and he doesn't seem to hear the complaints of the students
- 69.
- 70.
- 71.
- 72.