10.014 1D Project, Fall 2023
Write a software/game to solve an interesting problem

Last updated: 26 Oct 2023

Objectives

The objectives of this project is for you to

(1) apply the python programming concepts that are taught in this course
(2) practice technical writing
(3) work together in a group-based programming project

© Singapore University of Technology and Design, 2023

Summary (click on links for details)

Your Task	**Write a python program to solve an interesting problem.** What counts as "interesting problem" can be interpreted by you. SUTD's motto is A Better World by Design You need to decide who this game/software is for, what is the problem, and why the user(s) will use it.
Group Size	**5 - 6 people by default.** A smaller group can be allowed if necessary but should not be the default arrangement and is subject to approval. Sign up sheet found in eDimension.
Constraints	**Only python standard libraries allowed** i.e. user does not need to do `pip install` to install any extra libraries. Only libraries taught in this course are allowed (e.g. `libdw, pyrebase`). Inform your instructors if you want to use the Raspberry Pi.
User Interface	**Text-based UI is fine.** You can also use visual libraries or GUI libraries (e.g. `turtle, tkinter` etc) as long as it is available in the python standard libraries.
Group Deliverables	1) A piece of writing that describes the background of the software or game, a scenario describing the user(s) and why they would use your game, how the it is played and its main features 2) Documentation of the code 3) Code for the software/game in .py file(s) or ipynb 4) A 3-minute video showcase of your game / software Items 1 - 3 Grading Rubrics | Item 4 Grading Rubrics | Example Plagiarism Policy
Amount of code expected	**A ballpark figure is 80 - 100 lines of code per member.** This is a guideline to help you decide the workload. But, we are not going to count lines of code that you submit.
Individual Deliverable	You may be required to submit a peer review. If there is one, details of this will be announced in due course.
Grading Principles	The overarching grading principle is to **understand how well you have applied the python concepts** in this course. Creativity is not graded.
Group Assessment	**Attend a compulsory group oral exam in Week 13 Session 2**. 15 mins per team: 5 min talk + 10 mins Q & A. Know the policies on attendance. Instructors will draw up a schedule.
Deadlines	Please see the course calendar in the Course Handout.
Submission	eDimension. You will be told when the links are ready.
Clearing Doubts **More information**	Ask your instructors or post on MS Teams. Read the FAQ See the Example

© Singapore University of Technology and Design, 2023

Write a software/game together as a group

Your task in this 1D project is write your own python-based software or game that

- solves an interesting problem, and
- at the minimum, has a **text-based user interface**.

It is up to you to define what "interesting problem" means. Consider SUTD's motto: **A Better World By Design**. How might you make a better world for a certain user? You are free to program anything you like, as long as you can explain to your instructors how your python program "solves an interesting problem".

You need to decide who this game/software is for, and why it will solve the user(s) "interesting problem".

Your program can be based on an existing software or game, or it can be your own creation.

We shall define a game in this project as a python program created for the entertainment of at least one user.

There are the following restrictions to be followed by everyone **without exception**:

1) Your game shall be written entirely in python.
2) You may use only standard python libraries. i.e. the user need not do a `pip install` to play your game.
3) One exception to rule 2 is that you can use the external libraries that are introduced in this course, e.g. the **pyrebase** library
4) Your game must have a text-based user interface, any other visual library used in this course or available in the standard python libraries e.g **turtle** or **matplotlib** or **tkinter** can be on top of this.

If you are not sure whether a library is in the python standard library, check if it is in the list here: https://docs.python.org/3/library/index.html .
(If you ask your instructor, the reply will be to check this list, so check it yourself.
If you cannot find the library you want in this list, and ask your instructor, the reply will be the same). Hence, here are two examples:

- `pygame` is not allowed, as you cannot find it in the list above,
- `tkinter` is allowed, as it is found in the list above.

Apart from the constraints above, there are no restrictions on what your game or software can do.

You can add any feature you want, and the software can be used by any number of users. If you are writing a game, come up with your own rules, and the game could be of any type e.g. 1-player game, 2-player game, player vs computer etc. There are no restrictions on these aspects, all we ask is that your description (see below) makes these aspects clear.

© Singapore University of Technology and Design, 2023

If your project needs the Raspberry Pi, let your instructors know.

Our course has lab session(s) with the RPi and other hardware (see the lab materials). For this year, if you wish to make use of these hardware, please contact your cohort instructors. They will assess your requirements and refer you to our lab officers.
Should you loan out the hardware successfully, you are responsible for returning them by the end of Week 14 of this term.

How much work is expected from the group?

As a rough approximate guide, approximately

80 - 100 lines of code × the number of team members.

Thus for a 5-person team, we would expect to see around 400 - 500 lines of code.

This is a ballpark figure intended to give you a guide on how much work is expected.
Your instructors are not going to count the exact number of lines and scrutinize every line to decide if it counts as one line or not.

You need not worry if the size of the program goes above or below this number, but bear in mind the principles of assessment below. Thus if you give us less code, then it will give us less evidence to assess you on, and vice-versa.

If the team is really motivated to produce more code and a very nice software/game, please go ahead but be mindful of your workload in the course and in this term.

Form your teams of 5 - 6 people

Form teams of 5 - 6 people from within your Cohort class. A smaller team is also allowed, but such teams should not be the default size and is subject to approval from your cohort instructors.

Register your team in the sheets provided in eDimension by the deadline given in the teaching schedule. After the deadline, the instructors reserve the right to assign student(s) without a team to ANY team.

© Singapore University of Technology and Design, 2023

Consult your instructors if you have doubts on the scope of the software or game that you have planned, or if you need feedback

If you have clarifications on the specifications in this project brief, please ask your instructors or post your queries on the MS Teams channel.

Any additional information about this assignment e.g. missing information, additional details, clarifications etc ought to be information that is available to all students. Your cohort instructor can help to clarify immediately.

If the query has to be referred, the subject lead will:

- Respond to messages in the MS Teams group in the channel 1D Project queries
- And/or update the FAQ at the end of this document

There will be **no response** to emails or private messages of such nature.

© Singapore University of Technology and Design, 2023

How you will be assessed

These are the underlying principles behind the assessment

Your instructors want to:

- **Principle 1.** … first gain a basic understanding of the work that your team has produced.
- **Principle 2.** … determine how well you have applied the python programming concepts taught in this course.
- **Principle 3.** … know the extent of your individual contribution to the team.

Summary - What deliverables do you submit?

You

- submit deliverables as a group (group deliverables).
- attend the group interview session in Week 13 Session 2.

The deliverables are also subject to the policy on plagiarism. These are elaborated below.

You may also be required to submit a peer review after the group interview session. Details of the peer review will be announced in due course.

The scoring is as follows.

- (if there is no peer review) **a maximum of 10 points** will count for 7% in the 1D project grade
- (if there is a peer review) **a maximum of 12 points** will count for 7% in the 1D project grade

© Singapore University of Technology and Design, 2023

As a group, submit the following deliverables

Submit the following to the 1D project folder in eDimension by **Monday 4 Dec 2023 2359 hrs** or earlier, if requested by your cohort instructors.
Details on how you might submit these items will be described later.

1) **A description of your program.** The purpose of this description is to give the reader an idea of the software or game, how it is used or played, how you got the ideas and anything else you want to highlight.

   You ought to have **one** target user in mind (since your program is supposed to solve an interesting problem, so whose problem are you going to solve?)

   Thus it may include, in no particular order,
   a) A background of your game/software e.g. its origins
   b) A scenario behind your game/software: who is it for? What are the typical characteristics of this user? What is the interesting problem that the user is facing? Why/When would the user use your program? What benefits would be gained from using your program? (See Appendix 1 for how you might write this)
   c) how the software is to be used or how the game is to be played
   d) the main features in the software or game that you wish to highlight
   e) citations / references to the sources that you made use of when planning this software or game

2) **A documentation of the code** for your software or game. Please see Appendix 1 for an example of Items 1 and 2.

3) The **python code** for your software or game. It can be in one or more .py files or in an .ipynb file.

4) A video of up to 3 minutes that showcases your software or game. It should demonstrate how it is used or played and try to make others want to use it as well. How this video is done is up to you. Videos from previous runs of the course can be found in eDimension.

© Singapore University of Technology and Design, 2023

Attend a group project interview in Week 13 Session 2

This interview is equivalent to an exam and all members must be present.

Come early and be prepared. Your instructors will work out a schedule and your team must arrive early. Get ready all presentation materials and your code before your turn begins. The interview will begin with members of the team who are punctual. Being late means less time for the interview, which may then affect the score that you receive.

Present your software for five-minutes. Begin this interview by presenting your software and highlighting its main features for a maximum of five-minutes. Please plan this carefully to be concise and avoid exceeding five minutes. Visual aids (e.g. slides) can help you organize your presentation, but are not compulsory.

Exceeding the time limit of five minutes for your talk means you did not plan your talk carefully. This leaves less time for the interview questions, which will negatively affect the score that you receive.

Q&A for ten minutes. Then show your code for the game and your instructors will then ask questions about your code. The questions may be directed to the group as a whole, or to any particular group member. We expect that all members will speak up to answer questions.

© Singapore University of Technology and Design, 2023

Policy on plagiarism

Reference all external resources in your description and/or your code.
You can certainly make use of materials on the internet, but whatever you use must be stated clearly. Some examples:

- if you use a set of game rules in your program from a web site, then mention that website in your description and state how it influenced your own design. We do not restrict you to a particular referencing/citation style, please use whatever you are comfortable with.

  > This game is a modified version of [name of game or software], whose rules / description can be found at http://somewebsite.com. We adopted the following features … and ignored the following aspects.

- You find a python code online (in a website or from an AI tool e.g. ChatGPT) that helps you to do some feature in your program (e.g. how to do blinking text), then acknowledge its use in your code. Also consider mentioning it in your description if it is a significant feature. Your instructors are certainly keen to know the external resources that you have employed.

```
"""
The code for this function was taken/modified
from the code found at http://somewebsite.com
"""
def blinking(t):
    ##python code
```

You may not share your code with any other group, nor may you copy any code from a student outside your group. This project is meant to be a project within your group that allows you to apply your programming skills. Thus, the code that your instructors see must be a reflection of your own skills.

Because you may not share your code with any other group, code uploaded to an online repository must remain private only to your team.

Thus, the code should be largely your own work. We have mentioned above that using small bits of code found on the internet is ok, but your project must not be based upon copying large amounts of code from websites.

Thus, the lack of referencing to external resources and using other people's code in your game will be treated as plagiarism.

© Singapore University of Technology and Design, 2023

On the group-based items, the rubrics are as follows

Item 1.
The description of your game or software is **clear and complete**, including clear referencing and citation of external resources. Your target user, the user's interesting problem and the scenario is clearly described. The documentation of your game is **clear and complete**.

Item 2. The code shows an **excellent use** of the python concepts in this course. At the minimum, you must show that you have used concepts from Lessons 1 - 5 of this course. You should also show:

- Modularity
- Good programming practices
- Concepts and the various data types are applied appropriately
- Code that you write to test your program

You will also demonstrate your understanding through a team interview in week 13 session 2. If your initial presentation exceeds the five minutes allowed, or your team is not punctual, it will naturally affect the score in this item as your instructors will have less time to ask questions.

Points **Rubric Item**	**0 - 1**	**2**	**3**	**4 (maximum)**
Item 1. The description of your game or software is _____ . The documentation of your game or software is _____.	unclear and incomplete. unclear and incomplete.	somewhat clear and somewhat complete. somewhat clear and somewhat complete.	clear and complete to a large extent clear and complete to a large extent	clear and complete clear and complete
Item 2. The code of your game or software shows _____ of python concepts. From the group interview, the team members are able to show a/an _____ understanding of python concepts.	little use, with much room for improvement, little or no	some use, with much room for improvement, some	a good use*, with some room for improvement, good	an excellent use*, with a little room for improvement, strong
	*Note. For these categories, at the <u>minimum</u>, we expect to see you use concepts taught from Lessons 1 to 5 of this course.			

© Singapore University of Technology and Design, 2023

On the video, the rubric is as follows

Points **Rubric Item**	**0**	**2 (maximum)**
Video	No video is submitted.	A video is submitted that gives the viewer a clear idea of your game or software.

If a peer review is administered, the scoring is as follows.

If you are required to do a peer review, every member will be required to submit one and rate yourself and your teammates on the contribution to the project.
In short, the score that you receive will be the average score of the ratings. Details of the calculations will be released later.

Points **Rubric Item**	**0**	**2 (maximum)**
Peer Review	You did not submit the peer review.	You receive the best rating from your teammates.

On attendance at the group interview, the policies are as follows.

We expect you to have contributed to the project in the programming aspect and be present at the group interview.

If you are going to be absent, alternative arrangements can be made regarding the interview and the group score awarded to you if all conditions below are met:

1) You have an LOA from the university based on medical reasons or family emergencies
2) You inform both your instructors BEFORE the scheduled start time of the group interview by writing an email and, if possible, providing the evidence
3) If circumstances are such that 1) and 2) is not possible (e.g. it is an emergency and you can't apply for an LOA in time) your instructors expect to receive your email within **24 hours after** the scheduled start time of the group interview.

© Singapore University of Technology and Design, 2023

Some questions you might have

If there are more questions, we will add to this list of questions.

1. **Are the instructors looking out for creativity in the game or software?**

   We are mainly looking out for your understanding and application of python concepts and good programming practices. See the assessment principles.

   While a nice game idea will make your presentation and video interesting, those aspects of the rubrics are small compared to the others. You do not have to go all out to implement lots of features, or crack your head to make the software or game innovative.

   You can do very well with an everyday game or software together with some of your own ideas.

2. **How do we even begin?**

   Understand the objectives and assessment principles first. Have a look at Appendix 1 if it helps you. Then sit down as a group and brainstorm ideas for your project.

   Apply the principle of stepwise refinement discussed in Lesson 4. Plan the flow of the game or software and then this will give you an idea on how to break it up into functions. This can be done through a group discussion. The work can then be divided.

   There are online tools that you can use to collaborate in programming e.g. repl.it multiplayer.

   At some point, write the description and the documentation.

3. **Do I need to conduct user interviews or surveys to justify my game or software?**

   No. <u>Do not do any user interviews or surveys.</u> You just need to agree on at least one target user and the “interesting problem” as a group. You can come up with something based on your own life experiences or even something fictitious if you’d like.

4. **Will the instructor reject our definition of “interesting problem”?**

   No. This is meant to be very open-ended and as long as you can justify your choice and you are not doing anything offensive or illegal, your instructor has no reason to reject it. Please feel free to contact your instructors if you still have lingering doubts.

5. **What if we are not sure whether our idea is sufficient or suitable for the project, and we need help in scoping the project?**

   Please feel free to contact your instructors.

© Singapore University of Technology and Design, 2023

6. **We have barely started learning python programming and now we are expected to work on a group programming project?**

   The course gets such feedback in the end-of-course feedback every year and the instructors can empathise with such sentiments. We all can recall our own struggles with programming. However, since this course is Pass / Fail, and this project is only 7% of the marks, the important thing for you is to do your best and give it a try. There are many resources available for you e.g. read Appendix 1 for an example, consult your TAs and instructors during class. You can draw ideas for this project from your own lives, or maybe from the Maths or Physics courses. Do your *own* best work based on your *own* capabilities.

7. **Can we use AI tools to generate parts of the code for our project?**

   Yes you can, and you could get useful answers with very specific queries e.g. “how do you write python code to make text blink?”. Be sure to acknowledge its use in your project documentation and at the point where the code is used.

   Most of the code in your project should be written by yourself. We suggest that you use such tools sparingly and try your best to write the code on your own based on your own capabilities.

   Bear in mind that at the group project interview, you can be asked about any part of the code of your program, and you will be expected to explain it clearly and without hesitation. Not being able to explain will be viewed negatively.

8. **There is a member of our team who has not contributed to our team. What can we do?**

   At the start of the project, draw up a team contract, which is an agreement among yourselves on who should do what. Then maintain regular communications among yourselves.

   If, in spite of such efforts, you have a teammate who is not responsive and did not contribute, then approach your cohort instructors with the necessary evidence (e.g. chat messages showing a lack of respons) and they will take it from there.

© Singapore University of Technology and Design, 2023

Appendix 1. Shortened Word Game Example

Commentary on this example

- The scope of the game is brief and probably sufficient for a group of two persons, not five. My own solution had 110 lines, including blank lines. Even then, two CTD students with a good understanding of python can write more code than this :) Hence, for a larger group, there ought to be more features.

- The scope of the python programming concepts applied covers list, strings, dictionaries and file processing. Thus it has good coverage but there is scope to apply more concepts.

- The description contains the scenario and its main features. The scenario, as it is written, is adequate. It can be expanded if you decide to add more features. The features are rather brief and you can think of how to expand its scope if there are more members in the team (e.g. more types of questions, more animations, display images etc etc)

- The documentation covers all the functions. In many parts of the description, more details can be provided.

Team 12 - 1 Malay Vocabulary Game

Members: Muthu, Ngyuen, Ong, Rafi, Xu

Description

Scenario. This game is aimed at one of our member's little 7-year old niece Kate, who is going to visit the zoo soon with her parents during the school holidays. To help her stay in touch with her Peranakan heritage, Kate's parents want her to learn and remember the Malay words for some of the animals. Kate has been taught some of these words by her mum. Currently, Kate has difficulty remembering these words. A software that tests Kate on the words and is entertaining at the same time could help solve this problem.

Description of the game. This game is a series of multiple-choice questions, where Kate is presented with a Malay word and asked to select the correct English word from 3 choices. In total, five words are presented to her. Her score is recorded. If she scores 4 or more points, she is rewarded with an animation. Otherwise, a message saying "Try again" is displayed.

© Singapore University of Technology and Design, 2023

Documentation

It is required for this game to use the random library and the turtle library.

read_data_from_file(fname). This function reads in the words from the text file **fname** and returns a dictionary where the keys are the malay words, and the values are the english equivalent. All words are in lower case.

The format of the text file is

```
Kuda, horse
Monyet, monkey
```

select_key(dd). This function takes in the dictionary with the words and returns a key selected at random.

get_random_choices(dd, key). This function takes in the dictionary with the words, and they key returned by **select_key()** and returns two outputs:

- **options**, A list of two values chosen at random from dd (they do not include **answer**)
- **answer**, The value corresponding to **key**

return_question(key, value, options) . This function takes in three parameters. **key** is a string returned by **select_key(). value** is **answer** returned by **get_random_choices()** and options is returned by **get_random_choices**(). Thus, **key** is the malay word and **value** is the correct answer. It returns a string, which when displayed, is as follows

display_animation(). This function displays an animation using turtle library when executed.

display_try_harder(). This function displays the message “Try harder” when executed.

```
What is the meaning of kuda?
1. monkey
2. horse
3. tiger
```

main(dd). This is the function that runs the game. The pseudocode is as follows.

1. Initialize **scores** as zero.
2. For attempts from 1 to 5
    a. Get a key from dd
    b. Get the options from dd
    c. Construct the question string.
    d. Prompt the user to pick an option

© Singapore University of Technology and Design, 2023

    e. Calculate the score based on the choice
    f. Display the score and update **scores**
3. Display the total score
4. If the score is >= 4, display an animation, else display the message “try harder”.

© Singapore University of Technology and Design, 2023