

Centre For Innovation

Office of Dean Students

IIT Madras

Chennai - 600 036



cfi@smail.iitm.ac.in

<http://cfi.iitm.ac.in>

Centre for Innovation (CFI)

Indian Institute of Technology Madras

Application for the post of Programming Club Project Member, Centre for Innovation (**CFI**), 2024-2025

Name:		Roll No:	
Phone:	CGPA:	Email:	

Instructions and Conditions

1. Keep the subject as: PC-01_MTR_PM_<Name>_<Roll No>
2. Name the application as: <Name>_<Roll_No>.pdf
3. **Submit the application in PDF format along with the links to all the submitted code.**
4. Mail the completed applications (as a pdf) to:
 - a. cfi@smail.iitm.ac.in (Official email ID of CFI)
 - b. programmingclubiitm@gmail.com
 - c. cs21b005@smail.iitm.ac.in

Note:

- a) You can take help from any resources.
- b) Deadline for app submission: **11:00 PM, 22 May.**
- c) Try to be concise. Don't reemphasise something you've already mentioned.
- d) Bullet points are preferred over sprawling paragraphs.

A big ocean of learning is awaiting you. **Ensure you have fun** along the way.

All the best!!!



PC-01 Multiplayer Tank Royale

Project -

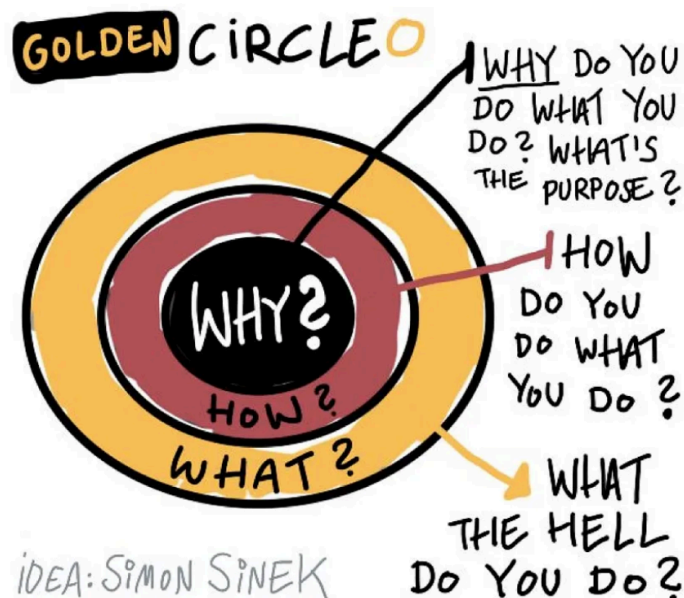
1. The aim of the project is to build a multiplayer tank game in C++ from scratch.
2. The initial part of the project will include establishing connections between various players over a network.
3. The latter half of the project involves rendering the front end and speeding up the computation with the help of GPUs.

Mentors for the Project - Anmol Panda (cs21b005@smail.iitm.ac.in), Aditya Palwe (cs22b003@smail.iitm.ac.in), Jatin V. Kumar (ee22b023@smail.iitm.ac.in)

NOTE:

1. Plagiarism checks will be done on all code submitted; kindly do not share your work with anyone. Copying work off the internet is also not advised, as others might have the same idea.
2. There will be a significant focus on the quality of the code. Use meaningful and descriptive variable and function names.

The key to any successful undertaking is to understand the Golden circle. Please answer all questions below, after giving this a thought.





Task 0: Requirements

We will be using a lot of libraries in C++. To ensure that all of them work on your system, you will need Linux as your OS or WSL if you use Windows.

It will also be required for some of the questions below.

1) Steps to install WSL ([Link](#)).

Do reach out if you have any issues while installing.

Task 1: OOPS In C++

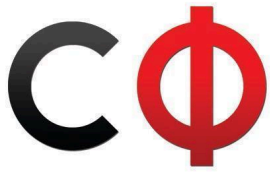
[15 pts]

1) Implement a dynamic array data structure in C++ using a class with the name “Tank”. The API for the data structure is to be exactly as defined [here](#). Strictly adhere to it. You are forbidden from using any data structures provided by STL.

You need to submit two files: one header file named “Tank.h” with the class function declarations and another file named “Tank.cpp” with the actual definitions.

Constraints:

1. At any point, the dynamic memory allocated to store your data structure should not exceed twice the size of the Tank (given by `Tank.size()`).
2. We advise you to use templates and not write classes for each data type.



Task 2: Understanding Multithreading

[30 pts]

- 1) What are threads and processes in the context of computing? Highlight some key differences between them.
- 2) What is a GPU? How does it differ from a CPU in terms of computation?
- 3) [Matrix multiplication](#) is a binary operation that produces a matrix from 2 matrices. It involves a lot of independent operations on matrix elements which can be parallelised.
 - Write a sequential C code to multiply two square matrices of size N and store it in a new matrix. Take N as input via a command line argument, and generate two random matrices of size N x N, which are an input to your algorithm.
 - Identify the independent computations done in the sequential code and using the [pthread.h](#) library, implement a parallelised version of your code.
 - Do a comparative analysis of both these methods. Use matrices of different sizes and observe their runtimes.
- 4) Refer to this code ([Link](#)). What do you think is the expected output of the code? Do you get the expected output when you run the code?
If yes, provide a brief explanation of the code. If not, provide a reason why you see a different output and how you can rectify the problem in the code.

Brownie Points: Why is the variable counter marked as “volatile”? What does the keyword “volatile” mean in C?



Task 3: Understanding Networks and Socket Programming [35 pts]

- 1) What is a socket, and how does it relate to computer networking?
- 2) What are the fundamental steps involved in creating a socket for communication between two devices over a network?
- 3) Implement a TCP server compatible with the following [TCP client](#) that streams and plays mp3 music.
 - The client enters a number from 1 to 3, indicating the choice of the song to be played, and the server then streams the corresponding song to be played.
 - The music streaming server should take two command line arguments: the port at which it will listen to (P), and the root directory of the mp3 song folder (DIR)
 - The server's console should display the following information: IP address of the client that has connected along with the song name requested.
 - We will evaluate this question over a single computer, so you do not need to ensure your code works over multiple devices.

References:

- Try to understand and learn how to run this sample [client](#) and [server](#) code. After the client and server are connected, the message "Hello World!" gets printed on the client's screen, and the connection is closed.

Hint1: The server-side code to send a music file is exactly the same as what it would be to send a text file (Why?)

Hint2: Understand how you can use [fread](#) to read from a file.

Brownie points: Since you are now an expert in multithreading, try to make the music server multithreaded so it can serve multiple clients at the same time.



Centre For Innovation

Office of Dean Students

IIT Madras

Chennai - 600 036



cfi@smail.iitm.ac.in

<http://cfi.iitm.ac.in>

Task 4: Game Design

[20 pts]

The only constraints of the game are that it should be a 2D game in top-view, and it should end in finite time.

1) Given the above constraints, what additional features or elements would you suggest to enhance gameplay?

Consider game mechanics, abilities, scoring systems, power-ups, etc.

2) Find appropriate libraries in C++ to implement the interface of the game.

3) Design a component inside the game (It could be a tank or any of the ideas you have proposed above). Make sure that it is feasible keeping in mind your proposed libraries.

We do not expect a code, but only a rough design (You can make it however you like - Paint, Canva or even a rough hand drawing).

About You

Answer these briefly and focus more on the above tasks.

1) What is your motivation to apply to this project? State your strengths and weaknesses.

2) What are your various commitments for the upcoming year? How do you plan to manage your time between them?