

# ÜBERSICHT ABSCHLUSSPROJEKTE

## 1. FORMALES

- Kurze, *eigenständige Programmierprojekte, selber zu schreiben, in C++*
  - Zu bearbeiten in *Zweiergruppen*
  - Abgabe (bis 1. August 2021, 18 Uhr)
    - Dokumentierter Quellcode (+ggf. Hinweise zum kompilieren)
    - Kurze, schriftliche Beschreibung (ca. zwei Seiten)
  - Vortrag: 2. August – 4. August 2021
- |           | 2. August | 3. August | 4. August |
|-----------|-----------|-----------|-----------|
| 9-13 Uhr  | Gruppe 1  | Gruppe 3  | Gruppe 5  |
| 14-18 Uhr | Gruppe 2  | Gruppe 4  | Gruppe 6  |
- In der eigenen Gruppe *Anwesenheitspflicht*
  - Terminprobleme: Mail an ludwig.striet@mathematik.uni-freiburg.de
  - Vorträge *über BigBlueButton* – bitte entsprechend vorbereiten!
  - Themenwünsche: *bis 30. Juni an ihre:n Tutor:in*

## 2. THEMENVORSCHLÄGE

Dies sind *Vorschläge* für Themen. Eigene Themenvorschläge sind *ausdrücklich erwünscht!*

1. *Einfache Spiele, mit einfacher KI, textbasiert*
  - BlackJack
  - Mau-Mau
  - Vier gewinnt
  - Kniffel
2. *Spiele mit einfacher grafischer Ausgabe (z.B. mit ncurses)*
  - 2.1. Snake
  - 2.2. TicTacToe
  - 2.3. Schiffe versenken
  - 2.4. Minesweeper
  - 2.5. Tetris
  - 2.6. Sokoban
  - 2.7. Conway's game of life
3. *Numerische Simulationen (in C++, zusätzlich geeignete Plots)*
  - 3.1. SIR-Modell
  - 3.2. Räuber-Beute-Modell
  - 3.3. Doppelpendel
  - 3.4. Zwei-Körper-Problem
4. *Rekursives Backtracking*
  - 4.1. Sudoku-Löser
  - 4.2. Damenproblem

- 4.3. Springertour
- 4.4. Spiele-Algorithmus (vgl. Rimscha – *Algorithmen kompakt und verständlich*)
5. *Bildverarbeitung (Skript zum einlesen auf Anfrage)*
  - 5.1. Bild laden – verarbeiten (s/w umwandeln, weichzeichnen) – Bild speichern
6. *Kryptografie/Verschlüsselung*
  - 6.1. Codebreaker für Caesar-Verschlüsselung mit Häufigkeitsanalyse
  - 6.2. RSA-Verfahren implementieren
  - 6.3. Primzahl-Tests (Fermat-, Solovay-Strassen-, Miller-Rabin-Test)
7. *Algorithmen und Datenstrukturen*
  - 7.1. z.B. Heap implementieren, vgl. mit `std::vector`, Theorie erklären
  - 7.2. Heapsort implementieren, vgl. mit BubbleSort (eigene Heapify-Operation!), Theorie erklären
  - 7.3. Quicksort implementieren, vgl. mit BubbleSort
8. *Speziell: Schule*
  - 8.1. Fischertechnik-Roboter aufbauen und programmieren
  - 8.2. Calliope Mini programmieren
  - 8.3. Stochastische Simulationen
  - 8.4. Monte-Carlo Integration vs Trapezregel
9. *Sonstige*
  - 9.1. Brainfuck-Interpreter schreiben (<https://de.wikipedia.org/wiki/Brainfuck>)
  - 9.2. Parallelisierung mit MPI, ggf. OpenMP
  - 9.3. BigInteger Klasse schreiben (um mit Zahlen zu rechnen, die für `long` zu groß sind), Methoden für `+`, `-`, `*`, `/`, Operatoren überladen
  - 9.4. Suchalgorithmen implementieren und vorstellen (z.B. A\*, Dijkstra)