Syllabus:

Class: eSports for Engineers - practice teaching and testing yourself, using AI and classic strategy games

Course description: fine-tune your learning style using very challenging mind sports and eSports. Use learning and goal-setting best practices[1,2] to improve mastery of classic games of strategy and skill. Games included are chess, duplicate bridge, Weiqi/Go, and physics-based realistic simulations. Award-winning, open source computer game software is used, supplemented by additional free software. The course is self-taught, with instructions and documentation provided.

Requirements:
any Microsoft Windows compatible PC or laptop
knowledge of how to install and run a linux operating system
500 Gigs of disk space on computer

Helpful, but not required: fast computer, game controller, internet connection, ability to purchase and download software

Cost: free.

Course Materials and Support Forum: download the game file at:
https://github.com/sim-museum/esports-for-engineers

Course Outline:
Learning best practices of appropriate difficulty, interleaving[1] and measurable results[2] are used throughout. Specifically, on each day you are encouraged to
1. use more than one software package, giving you a variety of AI sparring partners to learn from
2. configure the software for your skill level, where levels available typically range from beginner to world champion
3. measure and record your results every week, along with analysis and reflections on how to improve. (Game output can be packaged and posted to the forum if desired.)
MON: Start with the favorite game of Bill Gates, Warren Buffet and Deng Xaioping, Duplicate Bridge. Use several of the eight different bridge programs provided, each with different training, analysis and coaching features, and with different playing strengths. Example free open source (FOSS) program: wBridge5
TUE: Realistic sim racing programs provide mental and physical challenge. If new to sim racing, try drifting using more than one sim racing title and more than one model year. Try both 1950's and 1960's era racecars, these cars are especially challenging to drive because they lack aero and electronic controls. Example FOSS program: Speed Dreams
WED: Chess. Learning features include blunder check, drills, tutorials, heatmaps and a variety of AI opponents. Example FOSS program: lechess zero, open source clone of world computer chess champion Alpha Zero
THU: realistic flight sims and flight/war sims, with civilian and combat titles. Example FOSS program: Rowan's Battle of Britain
FRI: Weiqi/Go programs with many AI sparring partners and analysis tools. Example FOSS program: katago, open source clone of world Go champion Alpha Go.
SAT: Several versions of Falcon 4, one of the most complex computer games ever made. The simplest of these versions has a 750 page manual. Example FOSS program: free falcon
SUN: Several versions of poker, including 7 card stud and Texas Hold ‘em. Example FOSS program: pokerTH
[1] https://github.com/sim-museum/esports-for-engineers/blob/master/files/metaLearning.txt

[2] https://github.com/sim-museum/esports-for-engineers/blob/master/files/ObjectivesAndKeyResults_OKR.txt
How to learn eSports for Engineers:

1. Read the documentation! Unless you study and choose your learning targets, it's hard to make any progress with these challenging games. A summary mind map for each day is available; view these mind maps using the freeplane software, e.g. freeplane MON.mm
2. When you approach a subject using several different learning methods, you strengthen your understanding. Thus choose a situation to simulate, for example sim racing with a Ferrari at the 1967 Netherlands Grand Prix, then use at least two different sims to recreate the situation. Each sim has it's own perspective, along with unique strengths and weaknesses. To maximize revenue commercial game companies encourage you to stay with one game, but you learn more by using several different games. You might apply this approach on MON and TUE as follows:
3. Follow installation instructions carefully. Installing the dozens of sims in the eSports for Engineers package, each with a complex configuration, each in a different container, and using emulation of several different versions of Microsoft Windows, is straightforward if you follow the instructions. It is prohibitively time-consuming otherwise. After installing all the games, save a copy of the entire eSports for Engineers directory, using the supplied script to delete intermediate installation files to save space if desired. This results in a large backup file (150 Gig uncompressed), but if you need to reinstall linux later you can simply
a. reinstall ubuntu 22.04 LTS linux
b. run the script to install needed packages from the ubuntu repository
c. simply copy the backup file into the reinstalled ubuntu partition and uncompress if needed. In other words, all you need to do is copy a regular directory tree, or compressed version thereof, into the ubuntu partition. No special backup/restore tools are needed. You don't have to run any of the sim install scripts again - all the sims are now installed and ready to use.

4. Make use of AI coaching and feedback. After-game analysis is available for every game; in the MON directory, bcalc lets you step through a bridge hand, evaluating each move, while Q-Plus bridge provides this feature during the game. In TUE, GPL Replay Analyzer provides many reports and graphics, in WED, scid and chessmaster annotate your games, Mig Alley and Battle of Britain in THU offers replay videos, as do all the SAT flight sims. In FRI, KaTrain and goreview partner provide annotations.

Blunder checks during a game, which warns you if you are about to make a bad decision, are available for the card and board games. Among the MON games, Jack Bridge and Q-Plus Bridge provide blunder check. Also Chessmaster in WED and KaTrain in FRI provide this feature.

For MON, play through example bridge hands, with audio tutorials, using Omar Sharif Bridge. Input these same hands into Bridge Baron 12, then use the interactive online flow chart to help with bidding these hands. Input and bid the same hand using q-plus bridge, reviewing the bidding interpretations provided. Also use Q-Plus to create text output of bridge games. Then read descriptions of standard bridge conventions in the Omar Sharif Bridge documentation, then practice some of these conventions using Bridge Baron 12 convention practice.

For TUE, simulate a '67 F1 race at Zandervoort using Grand Prix Legends, then simulate the same race using rFactor. The Grand Prix Legends cars are easier to configure and drive, and are supported by more user-friendly telemetry utilities. rFactor has fewer carsets, but more features simulated and it produces more telemetry raw data. Race the Ferrari at Monza using Speed Dreams, then using Grand Prix Legends. Speed Dreams is simpler than the other racing sim, but the best in terms of displaying real-time physics during driving.