PENN STATE UNIVERSITY
Department of Economics

Econ 597D Sec 001 Computational Economics — Gallant
Project Suggestion 11 — Fall 2015
Due Dec 8, 2015

The code at `http://www.aronaldg.org/webfiles/arg/compecon/src/craps/` simulates a crap game for six betting strategies. Runs that log the play of the game for each strategy are executed by `debug.sh` and runs that compute statistics from long simulations are executed by `extract.sh`. The code uses the OOP ideas that we have discussed in class. The design is laid out in `craps_base.h`. The main is in `play.cpp`.

One can play the game at `http://www.crapdice.com/crapgame.html`. This website has a good tutorial and has Las Vegas Strip odds limits, which are 3x, 4x, and 5x for the 4 and 10, 5 and 9, and 6 and 8, respectively, as does the code. Files written by debugging runs have extension `log`, e.g., `pass_line_with_odds.log`, which can be compared to play at `crapdice` to check code accuracy. Files summarizing a statistical analysis (that were written by the main in `extract.cpp` using as input the file `play.csv` that was written by the main in `play.cpp`) have extension `txt`, e.g., `pass_line_with_odds.txt`. Incidentally, the statistics are computed using the template function in `simple.h` that we studied in lecture in connection with Chapter 8 of the text.

A comprehensive discussion of the mathematics of the game is in Gallant, A. Ronald (1997), *An Introduction to Econometric Theory: Measure Theoretic Probability and Statistics with Applications to Economics*, Princeton University Press, Princeton NJ, ISBN 0-691-01645-3.

Presently, the code implements the pass line bet, with odds, the come bet, with odds, and the any craps bet; the any craps bet is called a craps check when placed simultaneously with a pass line bet or a come bet. The code implements the six player strategies that use these bets listed in file `strategies.dat`.

The project is to add at least one other bet to the code and at least one other player strategy that uses it.

According to Stroustrup, Bjarne (2000), *The C++ Programming Language, 3rd edition* Addison-Wesley, Boston MA, ISBN 0-201-70073-5, a properly written C++ program should be sufficiently readable that it doesn't need documentation. This project puts that idea to the test. The code mimics how the game is dealt in a casino, Taucer, Vic, and Ralph Cutolo (1993), *Craps Dealing and Supervising*, Casino Creations, Las Vegas NV, ISBN 13: 9780924719080, ISBN 10: 0924719087. As in a casino, a marker controls the flow. The game starts with the marker off, players place bets, the dice are rolled, the house settles bets, and moves the marker if necessary. Bets are placed and settled subject to marker position. E.g., a come bet can only be placed when the marker is on; a pass line bet only when off; come bet odds are off when the marker is off yet the come bet itself remains on.

I wrote this code to answer some questions for which the math was too tedious: How much does taking odds increase variability? (Answer: A lot, standard errors increase by a factor of five. This is mostly a scale effect because the average bet increases by a factor of 3.8.) How much does a craps check reduce this variability? (Answer: Some, the factor drops to three even though the mean wager increases by 5%.) Does continuously making pass line and come bets without odds instead of making pass line bets only, which can only be made sequentially, set up a correlation that changes the variability per hour of play? (Answer: Yes, there is a negative correlation that reduces variability. Bets per hour increase by a factor of three but standard errors only increase by a factor of two. This is a net effect because bets that establish their points are positively correlated with each other but all are negatively correlated with a freshly placed pass line or come bet. The house advantage does not change.) For the six strategies, the house's advantage ranges from 1.5 percent for low variability strategies to 0.25 percent for high variability strategies. Variability is large enough that the house advantage is hard to compute by simulation; the house advantage is not difficult to compute analytically; see, e.g., bet_m_then_n.cpp. Incidentally, from this bit of code one can see that bets that establish their points are positively correlated.

The six strategies analyzed here, as well as variants that add pass line betting to continuous come betting, are known to be best. Their converses, where don't pass is substituted for pass and don't come for come, are equally good. I have not seen discussions of strategies that mix pass, don't pass, come, and don't come. Bets other than the above have house advantages ranging from 4 to 17 percent. The main exception is a place bet on the 6 or 8, or its converse, where the house advantage is about 1.5 percent.

The game is a lot of fun due to the social interactions of the players. Given that the house pays the heat, light, rent, and help and hands out free drinks liberally, it is not clear to me that the house has an advantage.