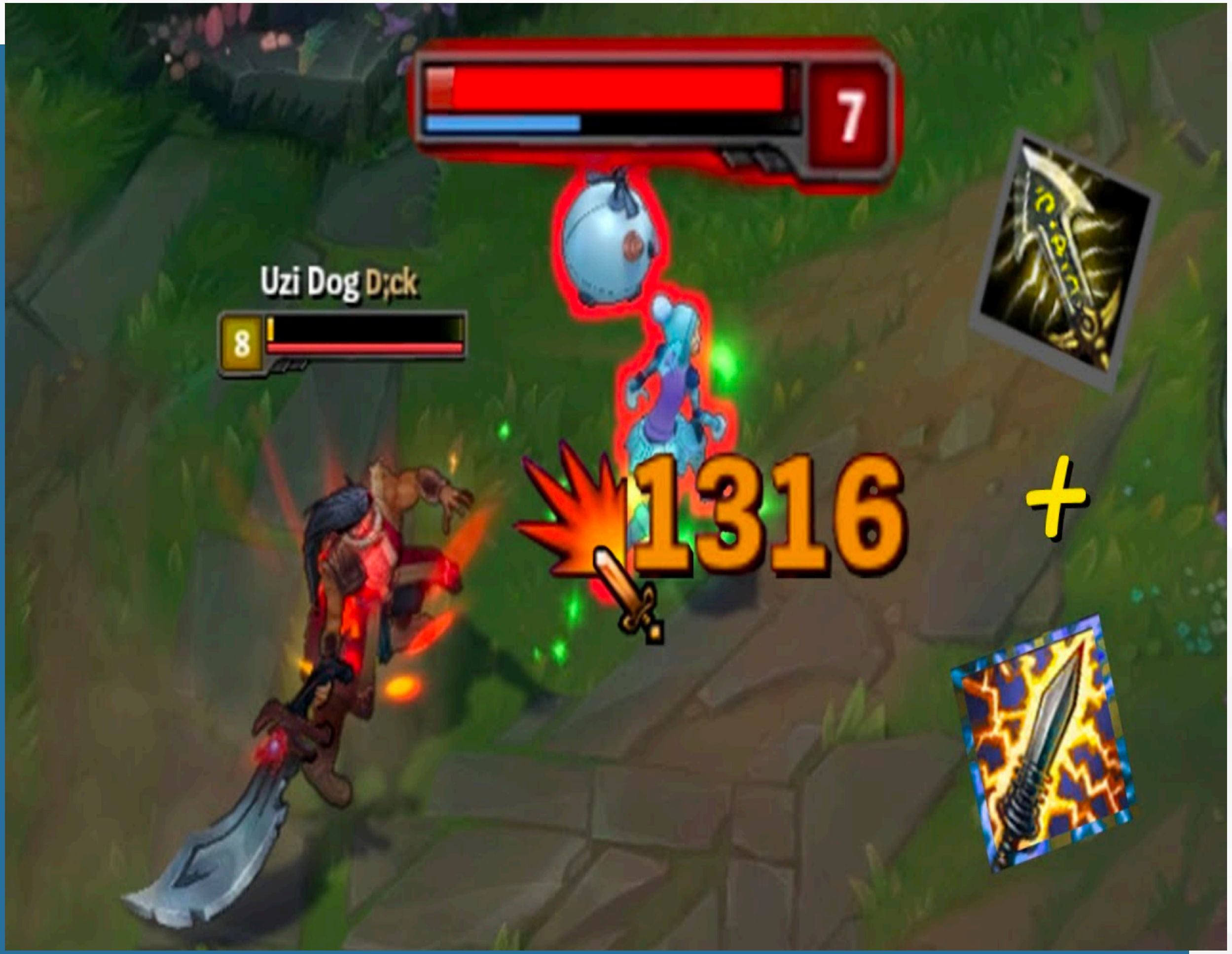
Monte Carlo simulation for different Random Number Generators algorithms in gaming, with User Interaction

Xinyu Lu

CONTENTS

01 Motivation



- Sometimes it is easy to get a critical strike, but sometimes there is no critical strike even though I hit many times.
- Random Number Generators (RNG) are a set of algorithms that generates numbers by random chance, and it plays an important role in game development. It is involved in the core gameplay and directly affect users' gaming experience.
- A simple case of RNG can be a coinflip to determine hit or miss for certain skill.

02 Background

True Random Distribution (TRD)

True Random Distribution can be approximately regarded as true random distributed, which means each number generated is independent to each other. It guarantees an evenly distributed sequence given a large enough sequence length, but it has some problems in real-world use cases.

E.g. Coinflip

Pseudo Random Distribution (PRD)

Different from True Random, PRD is introduced to generate a more evenly distributed random sequence. PRD generates random numbers in a sequence-aware manner. For a desired rate p, PRD starts with a smaller number x, and uses rate x to generate random judgement. If the previous decision is a hit, the current rate will be reduced and vice versa.

03 Hypothesis, model, design

Game

The game is relatively easy. I model the game as an RNG that produces sequence of 0s and 1s. Different RNG will be used for the game.

Hit or not?

In my project, the PRD sequence is generated according to the method used in DOTA2. Given the nominal chance, we can get a C value. The probability of an effect to occur on the Nth test since the last successful occur is given by $P(N) = C \times N$.

C	Nominal Chance	Approximate C
0.0038016583035531391017564 66	5%	0.38%
0.0147458447810726758770508 16	10%	1.5%
0.0322209143730876749751173 59	15%	3.2%
0.0557040429497818518583986 52	20%	5.6%
0.0847440918523169902752748 06	25%	8.5%
0.1189491927254039875837555 53	30%	12%
0.1579830981257470775575404 62	35%	16%
0.2015474136077540170706796 39	40%	20%
0.2493069984401631897146771 00	45%	25%
0.3021030253487419651691604 32	50%	30%
0.3603978509331686971046868 03	55%	36%
0.4226497308103742354908512 20	60%	42%
0.4811254783372291744019113 23	65%	48%
0.5714285714285714285714285 72	70%	57%
0.6666666666666666666666666 67	75%	67%
0.7500000000000000000000000 00	80%	75%
0.8235294117647058823529411 77	85%	82%
0.8888888888888888888888888 89	90%	89%
0.9473684210526315789473684 21	95%	95%

Player

The players are modeled as a connected graph. Each node represents a player, and edge between nodes means the 2 players knows each other. Connected nodes can share the generated sequence, so each player knows the sequence generated for itself and players connected as well.

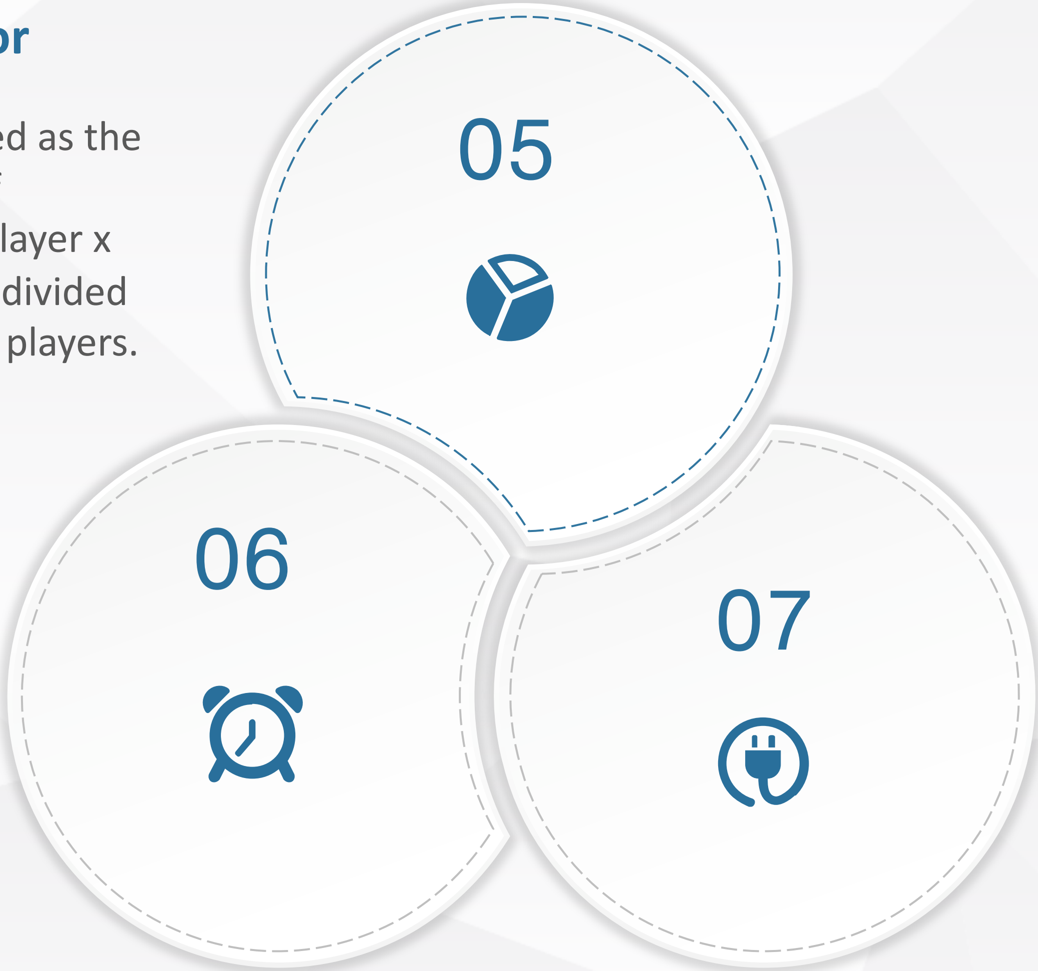
Value

Given the nominal rate and the number of trails that we try, the program will generate a sequence recording at what times we get a hit. The value of a player is defined as the length of this generated sequence for the player, that is, the number of hits we get, from specific number of trails at specific rate.

03 Hypothesis, model, design

Satisfying factor

The satisfying factor is defined as the sum of squared difference of sequence lengths between player x and all its connected players divided by the number of connected players.



Leaving game

Player with satisfactory lower than a certain threshold for certain rounds of gameplay will leave the game, with the sequence stop to generate. From this we can get users retention rate. (Under construction)

Uniformity factor

The sum of satisfying factors of all players in one graph. Smaller uniformity factor means a more general gaming experience.

04 Results

01 Graph

- 1000 nodes (players)
- 5000 edges (connection)

02 Settings

- Nominal chance: 0.3/0.5/0.7
- Number of trails: 20

03 Calculations

- Value for each player
- Average square difference of values of the graph (1000 players)

04 Results

- Diff under TRD: 84.7684/50.4824/42.2162
- Diff under PRD: 28.6593/18.8199/18.4821

(stable game experience, higher satisfaction degree)

THANK YOU !