Team Fortress 2 Python Simulation

Abstract:

This write-up serves as both a manual and a research paper relating to the Python simulation of Team Fortress 2. The manual part explains the classes available in the simulation and the research part provides two example pieces of hypothesis testing, as well as the overall evaluation of the simulation.

# Game description:

This game is loosely based off the Team Fortress 2 first-person multiplayer shooter developed by Valve Corporation.

The game features nine classes/heroes. In the simulation, all of them move by one cell per turn and cause damage to all enemy entities once cell around them and melee damage to enemy entities inside of the same cell. Ranged damage is random within a range, and melee damage is fixed to 65. Ranged attacks can be critical (3% chance) to triple the damage and so can the melee attacks 14% of the time for 195 damage. The game spawns the bots on the field, scrambles their move order, moves them around, forces them to attack or heal, and repeats until either only one team remains standing, or 100 rounds end with no clear winner, in which case the game ends in a stalemate.

Here is a brief description of the classes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class/Hero | Simulated TF2 class | Health Points with/without overheal | Ranged minimum damage (without critical boost) | Ranged maximum damage (without critical boost) | Notes/Special abilities |
| SpeedyBoy | Scout | 125/185 | 3 | 104 | None |
| RocketMan | Soldier | 200/300 | 24 | 111 | Self-damages 10% of damage on every ranged attack every move |
| FireMan | Pyro | 175/255 | 30 | 120 | Resistant to other FireMen’s ranged attacks |
| BlackDynamite | Demoman | 175/255 | 30 | 100 | Self-damages 5% of damage on every ranged attack every move |
| LargeWeapons | Heavy Weapons Guy | 300/440 | 3 | 134 | Has 19% chance of skipping a turn to restore full HP if health is below full |
| Builder | Engineer | 125/185 | 3 | 89 | Has a 49% chance of deploying a Turret/ HealthBox on every move of one does not exist yet |
| Marksman | Sniper | 125/185 | 50 | 150 | Damages enemies in the same column and line rather than in an area |
| Healer | Medic | 150/220 | 50 | 150 | Gravitates towards friendlies rather than enemies. Restores 20 HP per move to all friendlies in one-cell range. Hurt bot will move towards him if their health is below full and a friendly Healer is the closest possible target object. Self-restores 4 HP per move. |
| Infiltrator | Spy | 125/185 | 20 | 49 | Instant kill melee attacks. Will destroy all enemy Builder’s buildings in once-cell range. 50% chance of fooling the enemy that he is located at cell 0:0 when during enemy target calculation. Ignored by the Turret when it calculates enemy positions. |

Builder can also deploy two types of buildings:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Building | Health Points | Damage | Healing | Notes |
| Turret | 100 (no overheal) | 48 | N/A | Ignores enemy Infiltrators. Fixed amount of damage. Cannot roll for critical hits or melee. Cannot move. |
| HealthBox | 100 (no overheal) | N/A | 10 | Restores 10 HP to all friendly bots in 1 cell range. Cannot overheal like the Healer. Cannot move. |

All testing involved random damage amount, critical hits and randomized move order. All simulations were done on a 9x9 playfield. The visual variant of the game with event output can be found in Main\_visual.py.

# Hypothesis 1: When teams are equal, the team with at least one healer will prevail most of the time.

*See: Healers research.py*

In order to test this hypothesis, two teams are created per simulation. Both teams feature the full Hero lineup, except the Blue team is not running a Healer, but a second SpeedyBoy instead. The hypothesis is based on the idea that the Blue team would be at a strategic disadvantage regarding HP regeneration since the only healing methods they would have would be Builder’s HealthBox and LargeWeapon’s self-heal.

Ten Monte Carlo simulations were run, with 10,000 games played per simulation:

|  |  |  |  |
| --- | --- | --- | --- |
| Run | Red rounds won | Blue round won | Stalemates |
| 1 | 5,407 **(54.1%)** | 4,590 (45.9%) | 3 (<0.1%) |
| 2 | 5,267 **(52.7%)** | 4,731 (47.3%) | 2 (<0.1%) |
| 3 | 5,435 **(54.4%)** | 4,558 (45.6%) | 7 (<0.1%) |
| 4 | 5,361 **(53.6%)** | 4,635 (46.4%) | 4 (<0.1%) |
| 5 | 5,379 **(53.8%)** | 4,616 (46.2%) | 5 (<0.1%) |
| 6 | 5,339 **(53.4%)** | 4,653 (46.5%) | 8 (<0.1%) |
| 7 | 5,284 **(52.8%)** | 4,710 (47.1%) | 6 (<0.1%) |
| 8 | 5,395 **(54.0%)** | 4,596 (46.0%) | 9 (<0.1%) |
| 9 | 5,269 **(52.7%)** | 4,728 (47.3%) | 3 (<0.1%) |
| 10 | 5,406 **(54.1%)** | 4,588 (45.9%) | 6 (<0.1%) |

As evident, the team with the healer would come on top than teams without them, simulation after simulation. The relatively small skew of this bias can easily be explained with healer’s relatively small amount of HP restored. Additional tests prove that increasing the amount of health restored by the Healer skews the win bias even further in favor of the team that runs it.

# Hypothesis 2: the established 6v6 team format is superior

*See: 6v6 research.py*

In Team Fortress 2, one established competitive format is six players against six other players. After over ten years of testing, the community had settled that the most viable and balanced team matchup for competitive gameplay is this game’s equivalent of two RocketMen, two SpeedyBoys, one BlackDynamite, and one Healer – so-called “generalist” classes that mainly rely on their damage output rather than their special abilities (except the Healer, of course).

For this simulation, two teams are spawned: Red, made out of the aforementioned “generalist” classes plus one Healer, and Blue, made out of a randomly selected six bots more reflective of the chaotic team composition seen on public servers. All bots pose the same skill and have an equal chance of being selected for spawning for Blue. To make this a more fair match, [commonly accepted class limits](https://wiki.teamfortress.com/wiki/Standard_competitive_format#Class_Limits) are implemented:

|  |  |
| --- | --- |
| Class | Limit |
| SpeedyBoy | 2 |
| RocketMan | 2 |
| FireMan | 1 |
| BlackDynamite | 1 |
| LargeWeapons | 1 |
| Builder | 1 |
| Healer | 1 |
| Marksman | 1 |
| Infiltrator | 1 |

Ten Monte Carlo simulations were run, with 10,000 games played per simulation:

|  |  |  |  |
| --- | --- | --- | --- |
| Run | Red rounds won | Blue round won | Stalemates |
| 1 | 2,262 (22.6%) | 7,730 **(77.3%)** | 8 (<0.1%) |
| 2 | 2,248 (22.5%) | 7,744 **(77.4%)** | 8 (<0.1%) |
| 3 | 2,164 (21.6%) | 7,822 **(78.2%)** | 14 (<0.1%) |
| 4 | 2,290 (22.9%) | 7,704 **(77.0%)** | 6 (<0.1%) |
| 5 | 2,275 (22.8%) | 7,718 **(77.2%)** | 7 (<0.1%) |
| 6 | 2,218 (22.2%) | 7,772 **(77.7%)** | 10 (<0.1%) |
| 7 | 2,228 (22.3%) | 7,764 **(77.6%)** | 8 (<0.1%) |
| 8 | 2,271 (22.7%) | 7,716 **(77.2%)** | 13 (<0.1%) |
| 9 | 2,287 (22.9%) | 7,708 **(77.1%)** | 5 (<0.1%) |
| 10 | 2,251 (22.5%) | 7,735 **(77.4%)** | 14 (<0.1%) |

As evident from these simulations, Blue would win decisively despite their high probability of not having a healer. The disconnect between the findings of this simulation and a decade worth of actual real-world experience might highlight the shortcomings of this simulation since it is turn-based and the damage and healing values are based off possible per-shot damage, and not per-second or per minute. The simulation also does not account for generalists’ advanced movement techniques, such as rocket jumping. This implies that the simulation is not a perfect fit for researching this hypothesis and might require further adjustments to do so.

# Research: finding the most damaging class/balancing the simulation.

*See: Damage and Frags research.py*

When designing a game with a wide array of options available to the player, it is essential to balance these options to allow for relatively similar outcomes in the long run while allowing individual match-ups to have a clear upper hand. Monte-Carlo simulations shine in situations like these, as they allow us to test systems with random compositions and uneven components and get a general picture.

For this research, Red and Blue are made out of nine players with their classes picked entirely at random with no class limits. The melee and ranged attack methods are modified to log the amount of damage caused and the number of kills.

Ten simulations of 10,000 games each were run, with the total kill and damage numbers calculated during each game. The totals were added and divided by 100,000 to calculate average damage per game. The outcomes were such:

|  |  |
| --- | --- |
| Class | Avg damage per game |
| Turret | 19 |
| Builder | 223 |
| SpeedyBoy | 243 |
| BlackDynamite | 332 |
| Healer | 309 |
| RocketMan | 354 |
| FireMan | 369 |
| LargeWeapons | 496 |
| Infiltrator | 325 |
| Marksman | 1098 |

|  |  |
| --- | --- |
| Class | Avg kills per game |
| Turret | 0.09 |
| Builder | 0.78 |
| SpeedyBoy | 0.85 |
| BlackDynamite | 1.22 |
| Healer | 1.26 |
| RocketMan | 1.31 |
| FireMan | 1.40 |
| LargeWeapons | 2.03 |
| Infiltrator | 2.27 |
| Marksman | 5.60 |

As it is evident, damage and kills directly correlate.

This research has less to do with Team Fortress 2 and more with the simulation – as it is a fact that the actual game is more or less balanced. To balance the python simulation, and, therefore, make it more accurate, the following steps can be taken per class, in order of increasing effectiveness of the class in the simulation:

* Turrets turned out to be incredibly weak – the exact opposite of Team Fortress 2. The simulation can be expanded by adding the leveling system to the turrets, like in the game being simulated.
* Builder himself is a relatively weak class who mostly relies on his buildings. His low damage is acceptable but can be increased for the accuracy of the simulation.
* SpeedyBoy’s low performance is not reflective of the actual game, as Scout is enormously damaging in the right hands. The simulation can be expanded by adding the code to simulate his high speed and dodging abilities, as well as high speed.
* Healer’s high combat effectiveness is a bug due to his top damage accidentally being left over at 150. In TF2, Medic is very ineffective in combat and instead relies on his teammates for help. The simulation can be made more accurate by fixing this bug and adjusting the healing rates of the Healer.
* BlackDynamite, RocketMan, FireMan, and Infiltrator are very reflective of the real game. The simulation, however, can be further expanded by adding explosive jumping logic that would allow explosive classes to leap at their enemies at the cost of their health rather than draining a set amount per move.
* LargeWeapons turned out to be more effective in the simulation than the actual game, perhaps due to simulation not accounting for him being the slowest class. This can be mitigated by making the movement logic more complicated.
* Marksman turned out to be extremely powerful in the simulation. In the actual game, the Sniper can only damage one enemy per shot, which can be accounted for in the simulation by stopping his attacks after the first shot.

# Conclusion

Overall, while the simulation is not perfect, it does provide some insights and can provide even more by further complicating it.

I hope you have enjoyed reading this write up and wish you all the best!

# Sources used/further reading:

Official Team Fortress Wiki (2014). Retrieved from https://wiki.teamfortress.com/

Team Fortress 2 – Wikipedia, the free encyclopedia (2018). Retrieved from https://en.wikipedia.org/wiki/Team\_Fortress\_2