**“Game Plan” (not updated):**

1. Pick 3 or more noticeably different shooters
2. Analyze which parameters make their levels succeed
3. Choose shared parameters between them and discard the remaining
4. Expand on automatic analysis of those parameters

**Possible shooters:**

* CSGO
* Multiplayer
* Slow movement
* Two teams, different objectives
* “Search and destroy”
* Economy, only 3-4 OP weapons
* Molotovs + smokes cover space
* 1:55 round time
* Bomb plant switches attackers and defenders
* 1st bullet random spread, heavy recoil
* Engagements can be short, mid or long range
* Can loot dead enemies
* Wall bangable props
* DOOM 2016 (or Eternal)
* Single-player (2016 and some Eternal)
* Constant movement
* Array of main weapons
* Constant height travel
* No recoil
* Short range engagements
* Can loot dead enemies + fixed loot in the map
* Portals, explosive barrels
* Enemies surround the player
* *Let’s ignore the amount of Eternal mechanics and enemy variety*

**Update:**

What about Wolfenstein?

… Instead of choosing 3 opposite shooters, it is more interesting to analyze 3 shooter categories:

* Tactical FPS:

Online, slower, recoil, precise grenades, limited ammo, low enemy variety...

CSGO...

* Beat-em-up FPS

Single player, Lots of enemies, surround the player, switchable guns, virtually infinite ammo, no recoil, low-mid enemy variety ...

DOOM, Wolfenstein…

* Narrative FPS

Single player, need to reach some objective to progress, enemies spawn in front, usually no backtracking or only to unlock an area after getting a key, no recoil, events, higher enemy variety…

I would analyze the difficulty curve, also the distance to enemies when detected (unpredictable), simulations would proceed until the individual bot completes the level (in story mode you often complete it and don't go back)... The bot would find a door and then search for key or interruptor (different)

Half-Life...

I can analyze a bunch of shooters **(not necessarily in first person, the TFG could be renamed to shooter in general)** and come-up with pillars in their level design:

1. Shared parameters between all categories although different values
2. More independent functionalities and set-ups, dependant on the category

The prototype takes this into account and has a pre-made setup for each type.