- This project is using ACS and DECORATE because I didn't have time to spent to learn ZScript, also because of source port compatibility
- This project had Multiplayer COOP in mind since the beginning, but I haven't found a source port which could run this mod properly on it, so I used defensive programming in an attempt to have less bugs as possible when it can be played on multiplayer
- All the weapons are spawning their projectiles from the center of the screen, so players can have more precision
- One limitation I imposed to myself while making this mod is: Use only the resources of the game (sprites and sounds, for example), and edit them to make the experience more unique and to incentive my personal creativity
- I decided to create custom help screens, so players can get a notion of what to expect from the mod. That also make things easier to understand, since this mod adds a lot of features
- Each character has custom attributes, making the gameplay more unique and fun, without being off from Doom's gameplay feel
- I had to give to the player clone's the CORPSE flag manually, because it was bugging custom monster behavior when using "A_CheckFlag(CORPSE, "MissileEnd", AAPTR_TARGET)" on the Enemy Addons Pack (it may bug on other custom monsters too, if they use this function or similar)
- It has many custom made items, not present in any mod (as far as I know), which took my creativity to the limit, and ended in being fun items to make and use!
- Melee weapons have special primary and alternate fires because, generally, melee isnt used too often. I added those in an attempt to make players use them more oftenly to kill individual enemies
- It was really challenging to make invisible enemy portals, because oftenly the enemies spawned would get stuck in edges or in decorations. I used a custom logic in an attempt to reduce this chance. In practice, it seemed to reduce A LOT. The result seems satisfactory. I didn't find a way to completely fix it (maybe its not possible), but the "Speed 50" had apparently more success on the tests (the amount of enemies stuck is reduced drastically!)
- Every weapon has primary and secondary fire modes, but players are not forced to use secondary ones if they dont want to
- Upgrading weapons is an interesting idea, and I wanted to add in the mod. When the weapon is upgraded, it'll have a different projectile animation and/or color, so players can identify when the weapon is upgraded or not. I decided to use different upgrade rules for balancing reasons. It has these rules added:

    - Upgrading melee weapons give +100% damage and effectiveness of both primary and alternate fires
    - Upgrading pistol weapons give +100% damage (and no self harming radius damage for primary fires of projectile-based ones), no more self harm for alt fires, +25% more precision, and around +10% faster firerate (for projectile-based ones)
    - Upgrading weapon slots 2, 3, 4 and 6 give +50% damage increase and +25% more precision
    - Upgrading weapon slot 7 changes the firing pattern and gives +25% more precision
    - Upgrading weapon slot 5 give +25% damage increase (+50% damage on bosses) and seeker rockets track targets better
    - Upgrading weapon slot 8 give +25% damage increase, and +25% more precision
    - Every upgraded weapon can hit ghost foes, and is more effective versus bosses and radius damage resistant foes
- Another challenge was making special effects being beautiful, but not too resource intensive (I'm thinking of players with slow computers, I wanted them to play this mod too!)
- The Item Generator was another challenge for me. Initially, I thought of which items should they replace so the gameplay feels different, but still balanced enough to let players have fun and win maps. Also, enemies in IPOG didn't drop any ammo at all! It was difficult to make ammo balancing due to this mechanic. I also tried to mimic IPOG's original pattern for it, but with no success. The idea was to begin generating the next item only after the player took the already spawned one
- It has custom Teleport Fogs for the categories:
    - Teleported players, foes and spawned items (blue)
    - Players after someone using Dimensional Hole (purple)
    - Foes spawned from Red Portals (red)
- Another challenge was to make custom blood sprites and gibs (since the game didn't have proper blood sprites or gibs), so I had to edit what the game had to make this happen. The design appearance (NOT the sprites!) was based on gibs from Rise of the Triad (rotating gibs and gibs amount), Quake 1 (gibs blood trail), Quake Champions (center blood effect) and Unreal Tournament 99 (bouncy gibs)

- It was fun doing Objective, Bonus and Primary Items to players to take. The Objective and Primary items should be used by map editors in their maps with the following idea:
    - Primary Items are spawned only once per map, set by map editor
    - Objective Items are for final maps (episodes endings). Primary Objective Items should not be spawned if an Objective Item is present in the map (or maybe should?)
- I thought on adding a secret weapon in the mod, which should be rare to find, but incentives more exploration from the players (since its a powerful weapon)
- Another challenge for me was to do powerups (since IPOG hadn't them, if I remember correctly). So I had to think out of the box for custom ones
- I also added traps, so map authors can use them to make the maps more unique.
- In the programming area, I used KISS principle (Keep It Simple, Stupid) and XP (eXtreme Programming), and also Romero's Programming Principles when applicable. KISS means to make things as simple as possible so everyone can understand whats going on, XP means to implement, test and refine the code very frequently, and Romero's Programming Principles to deliver general quality for players and modders
- It has custom skill levels, but each one gives more variation for the gameplay compared to vanilla Doom. The easier one is very very easy, even who is not used on playing Doom can play in this skill level (I hope haha), but the last skill level is very challenging, should make you play like a (semi) professional all the time
- Another main goal for this project was to make it as most compatible as possible with custom mappacks (oldschool and newschool ones) and monster mods. That gives players and modders flexibility on what they want to play, and which features they want when they are playing
- The Exit Nexus is a very unique and interesting mechanic, which allows players to exit the map sooner. Periodically, a Bonus Item will spawn in the map, in a random position, generally not so far from where the player went. Once picked up, the player get some health and shield and some points, and is allowed to exit the map sooner if certain conditions are met:
    - Scorelimit is reached. This limit is calculated for every map start. The score limit should not be too low, to dont discourage map exploring and progressing, and not too high, so the earlier exit would only be possible only near the map's end; both cases are undesirable
    - All (sub)bosses are killed. Imagine bosses or subbosses in maps (boss maps for instance), these could be avoided if I didn't add this rule. It could break the fun factor if this wasn't added

- I also wanted to add as many mutators and gamemodes as possible, to make players entertained and not easily bored with my mod
- I had to edit some sprites (because they had some awful / defective frames, specially Dominatrix, Berserker and Veek ones) to improve visual quality on these actors
- For the Random Item Generator, when it replaces an ammo pack, its first spawn will be 2 ammo of that type (e.g., if it replaces Clipbox, it'll spawn 2 BulletAmmo items in the first time), and when it replaces a Health pack, it'll spawn a fair amount of Meditubes or Shield Recharges in the first time, and so on, for every replaced actor
- For the weapons, I decided to use specific functions, flags and properties, so all the projectiles pass thru players. Meaning: if you shoot an ally, or a friendly actor, like a clone, the projectile will pass thru it / him / her. This idea is based on Samsara, since, in multiplayer states for weapons, it had this implemented, so I thought it'll be a good addition for the mod, with changes on hitscan puffs. But that introduces a problem: Any prop can block projectiles. So a recommended flag to enable is "compat_missileclip"
- The Retribution gamemode is an idea I had with the following motivation: I wanted to give DM maps some kind of "Coop" or "Invasion" playstyle. Since this mod is compatible with coop only, why not try to create a custom gamemode with this in mind? Be advised that this gamemode is still **experimental**!
- For the Addon Enemies Pack, the infighting patterns are as follows:
    - Skill 0: all monsters may infight each other, even of the same monster type. Bosses can't infight bosses. Subbosses can't infight subbosses.
    - Skill 1 - 4: infighting is normal between (sub)bosses and non-(sub)bosses
    - Skill 5: infighting is disabled, but (sub)bosses can still damage lower tier enemies, except (sub)bosses
- For the Addon Enemies Pack, internally, have these enemy species:
    - EnemyBoss (PBoss, TBoss, KBoss)
    - EnemySubBoss (BigGuard, TWizard, KBot)
    - Enemy (others)
    - EnemyInstaWall (for BigGuard's projectiles and Red InstaWall; I wanted his projectiles to pass thru his own InstaWalls, but collide with enemies, without impacting the infighting logic for every skill level. Thats the solution I found)
- I tried to define custom ammo types without the vanilla Doom's inheritance, but it was bugging in the New HUD for some reason. It was showing 2 new entries: 1 for Bullet and 1 for Rocket. Couldn't find out why, even replacing every weapon ammo as the

custom defined ones, so I preferred to use this method, even if its ugly, its convenient for the players.

- For the Addon Enemies Pack, I had to use a very tricky spawning logic, in an attempt to make it compatible with (hopefully) all mappack styles, being oldschool or newschool. The idea is the following:
    - There are 2 actors per monster spot: a Spawner and the monster to spawn
    - First, the Spawner spawns the enemy, then goes to an idle state
    - When the monster dies, it'll kill the Spawner

      I had to do this way to make it compatible with maps that may use a variety of spawning logics (like open a doorl if all monsters of a type in an area are killed, ACS spawning logics, etc)