* Essential:
  + ~~Set up version control~~
  + Handle Text
  + Convert Player to Entity
  + Entity Component System
  + Input Handler
  + Interactivity Context Menu
  + Set up a scalable storage object that you can use to sort entities based on y-axis depth as a bootleg depth buffer before bliting to screen.
* Achievable:
  + Basic TCP Networking
  + Rebindable keys
* Pipe dream:
  + AI with pathfinding
  + Procedural Dialogue System, that doesn’t feel like shit.
  + Procedural Quest System, that doesn’t feel like shit.

A game where most objects are interactable in a modular way that allows you to achieve a higher level of immersion by allowing you to sometimes do functionally pointless options, but which allow for a greater level of roleplay and character development.

Entities and objects are to be constructed int such a way as to be modular and easily updatable, so that if no interactions exist in the moment, they can be added later to a broad selection of entities and objects.

In genre terms, the game should feel like a single-player MMORPG, where the player has a vast world to explore and progress through – some parts procedurally generated, others hand-made. In reference to other existing titles, it will incorporate systems similar the procedural modularity of Dwarf Fortress’ Adventure mode, Old-School Runescape’s skill levelling system, and the general plot outline and quest structure of Fallout: New Vegas.

Narratively the game contextualises itself as a bounty hunting game, where you are given one procedurally generated target somewhere in the world with no concrete information on where they are or who exactly they are. As the player converses with NPCs and completes minor tasks throughout the world and the various settlements within it, they uncover more about the target. In order to make the progression more compelling, the deductions are based on excluding misleading information, rather than directly receiving an answer. The player is to make up their own reasoning and decide what to trust and what to not.

The player will ideally have several options when confronting the target, ranging from killing them on the spot, sparing them and leaving them alone, taking them in alive, or abandoning the quest entirely and living a somewhat simulationist experience.

The game should be designed around the idea of hardcore permadeath, but can optionally have a mode with very sparse checkpoints. An alternative softcore mode can be essentially a somewhat mocking version where you are either an immortal character who cannot be killed and this is mentioned via additional dialogue as negative with all of the suicidal options are disabled and are replaced with funny quips and the player is nudged towards the regular hardcore mode for the “full” experience, since the softcore version will end up mercifully easy when consequences aren’t there.

* When creating all entities, make a singular sprite SDL Surface for them to reference at creation instead of having every single entity initialise sprites and textures for themselves. Animation might make that change, but for now keep it in mind.