**Beginning the game:**

If the tutorial is off, the game begins in the entry room to a normally generated dungeon.

If the tutorial is on, the game begins in a pre-generated tutorial dungeon that goes through a basic linear dungeon entrance and then proceeds to a normal part of the dungeon. The tutorial portion includes a series of books or something of the sort that introduce basic combat mechanics and game mechanics. For instance, a series of platforms to get to a high up door telling you how to jump, just tells you how to move, then a weapon stand that gives the basic sword and then introduces combat mechanics against a very weak enemy. On top of this, include a scriptable object with a bunch of booleans to show information pop ups for new weapon types and items and mechanics as they are acquired.

**Dungeon Loot:**

Materials: materials are things that can both be collected as drops from monsters, such as bones, and from the environment (ore veins in the caves, chests in the dungeons, etc.). Materials can then be given to npcs in town and sold or turned into consumables, trinkets, or weapons or turned into the Architect to construct npc buildings.

Trinkets: items that give small buffs to the character they are equipped to. These can be simple, like an amulet of protection that increases resistance, or complex, like an item that reduces resistance to give health.

Melee Weapons: items used by the character for attacking with the basic two attack buttons. How the weapons are used varies by attack. See weapons system.

Ranged Weapons: these are weapons that can be activated if they are in the player’s hotbar.

Currency: probably limited to just

Consumables: items like potions that can drop and be used from the hotbar

**NPCs:**

**Architect:** Appears after you emerge after your first dungeon run. He is the architect of the town that arises around the dungeon. He shows you the list of all of the NPCs in town, how to get them to come or how to upgrade them. To upgrade them, you give the architect materials required and he upgrades them. The architect cannot be upgraded.

**Bartender:** Appears after the player meets the requirements to unlock one other character. They are the person who can be talked to to access the menu of all of the player’s characters they can choose from. Upgrading the tavern upgrades the number of people the player can have.

**Blacksmith:** Appears after the player finds a non-default weapon. They are the person who some weapons can be bought from and who can be used to craft weapons. In order to craft using the materials from an area, the blacksmith must be upgraded with materials from that area.

**Engineer:** Appears after finding a blueprint for a modifier (like godly sword or something). Allows the player to use materials to give a weapon specific modifiers. Upgrades allow weapons from a specific area to be given modifiers.

**Social Worker:** Appears after the first character death. Can be paid to spend time with a character to give them a new characteristic. Can be upgraded to allow for more people to be “treated” at once.

**Diviner:** appears once her divining orb is found. Can be paid to enchant mundane trinkets into a random trinket (cheap) or specific trinkets (expensive). Upgrades from specific areas allow for trinkets from that area to be crafted.

**Hunter:** appears after a ranged weapon is found. Basically the blacksmith for ranged weapons.

**Classes:**

Every class begins with a single base weapon used to explain why they are proficient in said weapon type.

|  |  |  |
| --- | --- | --- |
| **Nerd** | No favorite Weapons |  |
| Unlocked immediately since the starting character | | |
| All basic low stats | | |
| **Redneck** | Heavy Weapons (starts with a hatchet) | Ranged (starts with a shotgun) |
| One of the starting classes available once the bartender comes | | |
|  | | |
| **Janitor** | Heavy Weapons (starts with a mop) | Piercing Weapons (starts with a broom) |
| One of the starting classes available once the bartender comes | | |
|  | | |
| **Mailman** | Light Weapons (starts with an envelope) | Thrown (starts with a package) |
| One of the starting classes available once the bartender comes | | |
|  | | |
| **Priest** | Thrown (starts with a “holy book”) | Light Weapons (starts with a holy symbol) |
| Can be bought once the temple room is found in the dungeon | | |
|  | | |
| **Chemist** | Thrown (starts with a beaker) | Ranged (starts with an acid sprayer?) |
|  | | |
|  | | |
| **Farmer** | Piercing Weapons (pitchfork) | Heavy Weapons (starts with a hoe) |
|  | | |
|  | | |
| **Lawyer** | Thrown (briefcase) |  |
|  | | |
|  | | |
|  |  |  |
|  | | |
|  | | |

**Weapons:**

Types:

* Heavy Weapons
  + Greatswords
  + Scythes
  + Greataxe
  + Warhammer
* Light Weapons
  + Daggers/shortswords
  + Hatchets
  + Clubs
* Piercing Weapons
  + Lances
  + Spears
  + Rapiers
* Ranged
  + Cannons
  + Guns
  + Bow and Arrow
  + Crossbow
* Thrown
  + Bombs
  + Darts
  + Ninja stars
  + Potions

Stats:

* All:
  + Damage (as a percent of the physical/magical damage)
  + Damage type (physical/magical)
  + Knockback amount (displayed as low, high, etc. but contained as a float amount)
  + Description
  + Speed (difference between the different weapons (scythe attacks faster than greatsword)
  + Icon sprite
  + Held sprite
  + DOT properties
    - Dps
    - Duration
    - Particle color / type
* Melee:
  + animator
* Ranged:
  + Hidden:
    - Recoil (reduced by knockback resistance)
    - reference to projectile prefab or raycast
    - when projectile instantiated, a setup function will be called to pass in the properties from the ranged weapon
    - projectile speed
    - air resistance (how much the projectile slows in the air)
    - projectile gravity
    - bounces
    - on hit or on detonate (if on detonate, detonate delay, radius)

**Character stats:**

**Visible:**

Health: The amount of health the character has

Attack Power: the base number that weapon damage is based on. For example, one swing with a greatsword might deal 200% attack power

Magic Power: the base number that some ranged/thrown weapon damage is based on

Resistance: a percentage reduction of incoming damage

Movespeed: the speed the character moves at

**Hidden:**

Knockback Application: how much the character knockback the character deals. Is a multiplier to the static amount of the weapon. For example, a hammer will display “high knockback” on its tooltip and will be a high multiplier of this value.

Knockback Resistance: a percentage of knockback the character will ignore.

Jump Height: how high the character can jump

Double jump count: the number of jumps after jumping off of the ground

Dash Cooldown: the cooldown of dash. 0 means dashes not enabled.

Dodge Chance: the chance to ignore incoming damage

**Controls:**

A: attack

X and Y: attack buttons

4 hotbar slots: dpad

Up to 4 abilities: c stick

Use alternate weapon: zl

Dash: L and R

**Character List (barkeep):**

Use a scrollable list of buttons. When the button is pushed, it displays that character’s stats and equipped items in the top 1/3 of the barkeep’s UI. There is then a button next to the displayed character stats that says swap. When this is pressed, that character is used.

Stored info for each character: baseClass scriptable object, an indexed array of length 6 of the equipped items (2 weapons, 4 trinkets), an arraylist of traits, and an indexed array of upgrade points (length equal to the number of visible stats).

In the barkeepManager script, have a struct that includes all of the info above. Then the barkeep contains a list of these structs for all of the different characters.

**Animating Weapons:**

All weapons will have a reference to a gameObject. That GameObject is a prefab that holds the sprite when held and the weapon’s animations when used.

When the weapon is put into the primary equipment slot, the GameObject will be instantiated from the scriptable object and then parented to the character GameObject and then have its anchored positions set. The GameObject’s animator will have triggers that are called when attacks are made.

**Trinkets:**

Abilities of the trinkets are encoded by having a single class/monobehavior with all of the actual code for the abilities. In said monobehavior, there is a method that takes in an enum for the ability to use and then performs that ability. Each trinket scriptable object will then contain an enum value that will be accessed whenever the ability is used from the hotbar.

**Dungeon Areas:**

Entrance: Normal Western castle, inhabited by various undead, animated armor, etc

Below the castle are the caves, where there are dnd cavern creatures like spiders, slimes, cloakers, fake stalagmites

From here, there are different parts in the other 3 cardinal directions.

* In one direction is a marble Greek/Roman inspired area with monsters inspired from Greek myth. Stuff like satyrs, sirens, centaurs, etc.

**Character Traits**

|  |  |
| --- | --- |
| Versatile | A bonus to using non-favored weapons |
| Vulnerable | Lowered Resistance |
| Weak | Lowered Knockback |
|  |  |
|  |  |

**Map Generation:**

A grid of nodes. Each node stores its position, any room openings, and any connections. For example, a 1x2 room will be two nodes with a stored connection between them. Then use dykstra’s to ensure continuity to every room. Then have data structures that store possible rooms and their rarities and fill in the generated graph.