

“Welcome to the Great Maw! Whichever afterlife you’re into is quite, quite full at the moment, so we’re running trials to determine who gets to attend and live in ‘eternal bliss’, or whatever. Luckily, you were one of the good ones back on earth, so you get to compete! First things first, let’s get rid of that pesky body.”

**\*\* RRRRRRIP \*\***

*Sound plays as the skeleton is ripped from the body. Gore not included. We’re trying to keep this intro at least somewhat light.*

“Perfect! Now that you’re free of that useless flesh, let’s get to the bones of the situation. Ha! I crack myself up. Here’s how it works: you go in the maw, and if you can get out the back in 206 pieces, great! Otherwise, straight to purgatory. You can get another shot in, lets see, six thousand, three hundred, and twenty-two years! Not bad, looks like you’ll get to skip ahead of some of the less do-good do-gooders. Congratulations!”

**\*\* KICK \*\***

*You are kicked into the maw.*

## Welcome to *The Maw Wants Bones!*

*The Maw Wants Bones* is a **roguelike** game where you are the weapon! Enter the ever-shifting Great Maw in a long-shot bid to win your space in the afterlife, and use your skeleton to survive! Something about the magic holding your skeleton together is allowing you some ‘freedom’ in your choice of bones. Change your arms, hands, legs, feet, body, and skull around with ‘loot’ (bones of the losers before you, weapons, magic, you know, ‘dungeon stuff’) and enchant your bones with the magic of the maw; find combos and battle other skeletons for a shot to survive!

**The game has four pillars that make up the core experience:**

- **Difficult, strategy-driven turn-based gameplay**
- **Comedy! Make the player laugh**
- **Satisfying loot system; feeling of discovery with item combinations**
- **Endless replayability**

This game will be digital only, with sale on a digital retailer such as Steam with a low-to-medium price point. No microtransactions, no multiplayer. DLC is an option if the game is successful to increase its lifetime. This game is something that could theoretically be executed by a smaller-scale team to increase profitability and therefore is likely better suited exclusively to PC to decrease development difficulty, at least until it gains popularity in the roguelike scene. Development of games like this with a small team carries risks; especially in ‘biting off more than we can chew’. It will be important to plan intelligently in the pre-production phase so as to avoid an over-extension of the team.

Think Pokemon meets the offspring of Slay the Spire and Binding of Isaac -- below are the core components of the vision:

### → Combat System

- ◆ **Turn-based**, pokemon style combat, with your skeletal 'build' determining your moveset.
- ◆ Combat is not intended to be difficult from a gameplay perspective, but should be **a platform for the player to execute the strategy** afforded by their build.
- ◆ **Enemies should be procedurally generated** and diverse; they should fit within the loot system and their builds should both **educate and inspire the player**.

### → Loot System

- ◆ Loot system and build system should be **complex and deep**; a diverse set of loot with synergies and interactions is critical to the success of a roguelike.
  - *In a vein similar to Pokemon's PP; loot might have 'durability'. Bones break, after all.*
- ◆ Effective **balance** and diverse **procedural generation** is important here.
- ◆ Loot needs to fit in with the story, and will also serve as a **critical comedic lever** within the game.
  - *Try out the hard-hitting 'Skull and Crossbow-nes' in your head and arm slots today!*

### → The Map

- ◆ The map is **procedurally generated** to support **endless replayability**.
  - *Replayability could also be supported by a more roguelite-esque system where your previous runs impact your current run as enemies sift through your remains. After all, they want to get to the afterlife too! Why wouldn't they pick through your dead body?*

### → The Story & Comedy

- ◆ **Story isn't the main focus** but should still drive the game in some respects, mostly as **a foundation for worldbuilding and comedy**.
- ◆ Much of the comedy will be supported by the natural absurdity of the game's loot system and will be **baked into loot design**.
  - Paying special attention to placing comedy in the small details -- enemy names, place names, quips, and other writing elements -- will be critical.

### → Art

- ◆ **Combat Art and Animations**
- ◆ **Loot/Enemy Art**
- ◆ **Map/World Art**
- ◆ **UI/UX Art**

### → Sound

- ◆ **Music**
- ◆ **Sound FX**
  - **Combat**

- Animations
- UI/UX
- ◆ Dialogue
- ◆ Atmosphere

## The Team

Now that we understand the core foundations of what we want in this game, we are ready to lay out a hypothetical production plan for the project, detailing who we need, when we need them, and what we want to do in each stage of production.

We need a content designer -- a funny one too -- someone who can create items and item 'genres' that are both interesting in a strategic way but also entertaining to use; synergies need to be both strategically deep and funny. They need to understand how the game works, how to implement this skeleton concept successfully into the roguelike genre. Not an easy task, and involvement of the whole team will be critical. Enemies are in a similar but simpler vein, as they will mostly be carried by the item system in theory. The content designer will also need to flesh out the map, they need to understand how the world will impact the gameplay and what roles the world will serve within the game. The interaction between the overarching story (which will likely be minimal) and the gameplay will also be the responsibility of the content designer.

The systems designer will work hand in hand with the content designer in the functional and balance elements of the loot system, as well as understanding what and where and how to use procedural generation with the systems engineer. The systems designer is also responsible for fleshing out and balancing the combat system.

The gameplay engineer will work tightly with the systems designer to build the combat and loot systems as well as the procedural generation that drives the replayability of the game. Combat AI is another critical responsibility.

The misc (graphics, audio, engine, and tools) engineer will work with the artist(s?) to implement the item, character, enemy, and world art as well as building out the base systems for the game to run on. They will also implement the tools that allow automation of content implementation such as items, which will save critical time for the team. While these aren't roles that would always be combined, I think this game isn't overly complex from an engine perspective. Platform support and performance shouldn't be issues, and if needed the game could be built on an engine like Unity to save development time and resources.

It's possible this game could get by with one artist. Much of the art work will be in the development of individual art and animations for items; world art should be fairly and simple after it's locked in post-concepting and character art is virtually just an extension of item art (enemy art likewise, though there could easily end up being plenty of non-skeleton enemies in the maw; who knows where the design leads). Our misc engineer could potentially handle a majority of the implementation work, especially as it's fairly rote and will be able to be automated with tools.

Finally, we need a producer. Someone to keep everyone on track, to problem solve and fill in the gaps. A producer with some marketing knowledge would be critical to avoid having to hire extra marketing to spread word for a game like this, although contracting a marketing firm is also reasonable if necessary.

### **Pre-Production**

- Heavy concepting -- develop world, items, art, story
- Prototype combat, item balance, synergies and schemes for fun replayability
- We know the game pillars and the platform; let's flesh out the player experience.
- What tools do we need?
  - ◆ Tools to streamline implementation of items, art
- How long will it take to make this game? How many people do we need and when?
  - ◆ Can we get by with a 'unicorn' gameplay engineer that can also function as a systems designer?
  - ◆ Can an artist handle the audio? Do we need more than one artist?
    - A single artist probably can't handle all the audio and all the visuals. Can the team all work on audio together? Is the audio simple enough to avoid needing a dedicated expert?
  - ◆ Can the content designer handle the story on their own, or do we need to hire a writer? Do we scrap any story all together as opposed to contracting or hiring someone to manage it?
- Are we building our own engine or using something like Unity?
- Understand risks

At the end of pre-production, we want to know the 'vibe' of the game. We want to understand our world and art style, and to have an understanding of what the player experience is and why it is fun. We want to prepare and test our item and art implementation tools and have the pipeline ready for use as we move into production, as well as understand how long this project will take upon seeing the results of our pipeline testing.

### **Production**

- Build out all the features and systems
- Create a vertical slice to unite the vision for the team
  - ◆ Vertical slice will also be important for marketing and showing off the game, so make the vertical slice funny, too. Let the comedy do the marketing leg work with the virality of the internet.
- Make the assets, items are in many ways dependencies for a lot of other systems, so create a functional pool of items early on so other systems can be built while the breadth and depth of the item pool increases.
  - ◆ If tools for creating items are well made, everyone will be able to contribute, which will be critical. We'll need a lot of items to support this style of game, so spreading the load out across everyone will help significantly.
- Other dependencies may arise depending on the design of the planned systems, so planning effectively to manage these dependencies will be important in production.

- This team will be fairly agile as it's a smaller team, and development in the production stage should be full of iteration and collaboration
- We should plan to prototype everything, checking in with the game regularly to make sure we're still hitting reasonable targets for difficulty, strategic expression, and fun.
- To look at the triangle analogy, this game is one that is more focused on cheap and quality over speed; building the game with a smaller team will decrease cost but increase development time.
  - ◆ The smaller team should however enable the game to feel authentic to the personalities of the creators, something that will be important to the 'mouthfeel' of a comedic game like this where personality will come through in design and execution.
- Flesh out our marketing strategy and prepare the relevant cinematics/gameplay snapshots/assets

At the end of production, we want to have our game close to completion. Features and systems should be finished with assets almost finished as well (items may still be coming in). The game should be fun, replayable, and align with the core pillars. We should be ready to embark on post-production.

### **Post-Production**

- Finish the assets that aren't yet finished, and polish the assets that are 'finished'.
- Perform QA and begin the rise and fall of the bugs graph.
- We can potentially use platforms like Steam Greenlight for alpha and beta testing and to drum up hype for our release.
- Perform localization -- localization is a big area of risk for this game, as comedy isn't always translatable. This will need access to capable localization experts, likely needing some form of contracting or temporary hire.
- Execute our marketing strategy.
- As our game reaches content completeness and we move to polishing bugs, we will reach bug convergence and hopefully go through the process of bug bounce.

### **Then, we release our game!**

If it's beloved, we can add more content and release new elements and features for the game. We shouldn't really need to worry about live support, and there's no multiplayer, so we don't have to worry about server issues.