**DESIGN DOCUMENT FOR:**

**BEYOND THE GATE**

**CONCEPT:**

The concept of Beyond the Gate is a combination of rouge light elements with tactical turn based combat. Set in a fantasy world, the game is played through a combination of five heroes each with different skills and abilities that must be used in the correct combinations and manner in order to progress further and further into an unknown land. The game plays out in a series of rooms, each with a unique encounter of enemies, traps, and environmental conditions. At the completion of each room the player will have multiple paths they can take in order to progress, with the ultimate goal of destroying a final boss. Players will be able to see a map with details about any room they have encountered so far in the team creation stage, so planning a potential route depending on which characters they bring to the fight will be an important part of the game.

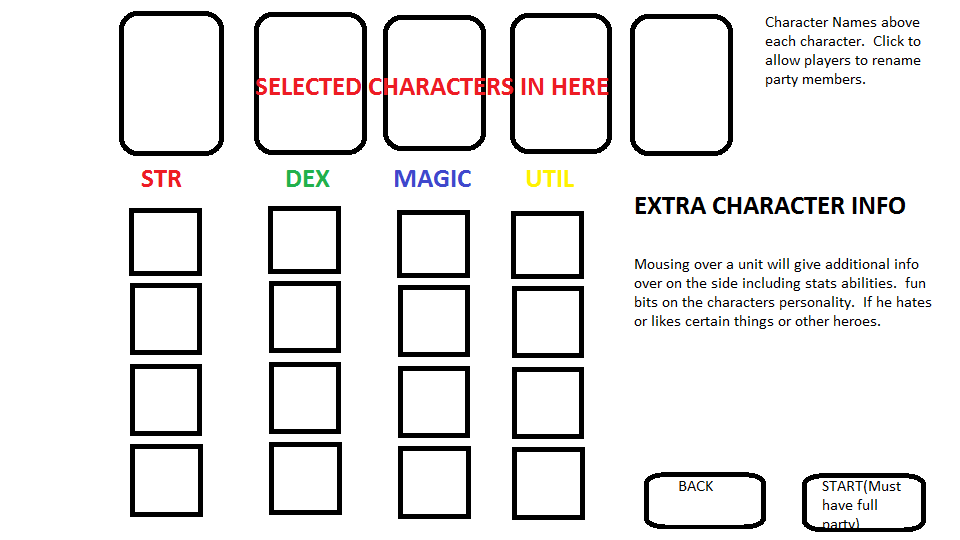
The game is meant to be extremely challenging. Any character killed in the game will experience perma-death, and having the entire team killed will result in a game over, forcing the player to start over again from the start with a different team. Some rooms in the game may change depending on actions taken in other rooms as well, further emphasizing the route players take. Certain routes, environmental interactions, and items will only be available if certain classes are selected for the run.

The game is planned to be released in 5 stages, each stage adding additional rooms, equipment, enemies, mini-bosses and final boss to the game. The final boss will be the end goal of each installment, and unlike all other rooms, once a final boss has been slain he will not return, but have a new encounter placed inside his room. Defeating a final boss will also unlock an additional starting point, further increasing the complexity of route selection for seasoned players. The goal is to create a game with extremely high replayability and skill cap, while appealing to those that like a challenge and a more strategic style of game play. Multi-player is a consideration for future installments but not the original release.

**INTERFACE:**

A main menu will be utilized and should include at least the following: New game (with warning that any currently saved game will be deleted if selected), continue game, book o knowledge (information on what the players has encountered so far in the dungeon including a map of explored areas), options, multiplayer (greyed out for now), and exit.

**Character Selection:** Upon selecting new game, characters will have to form a party.

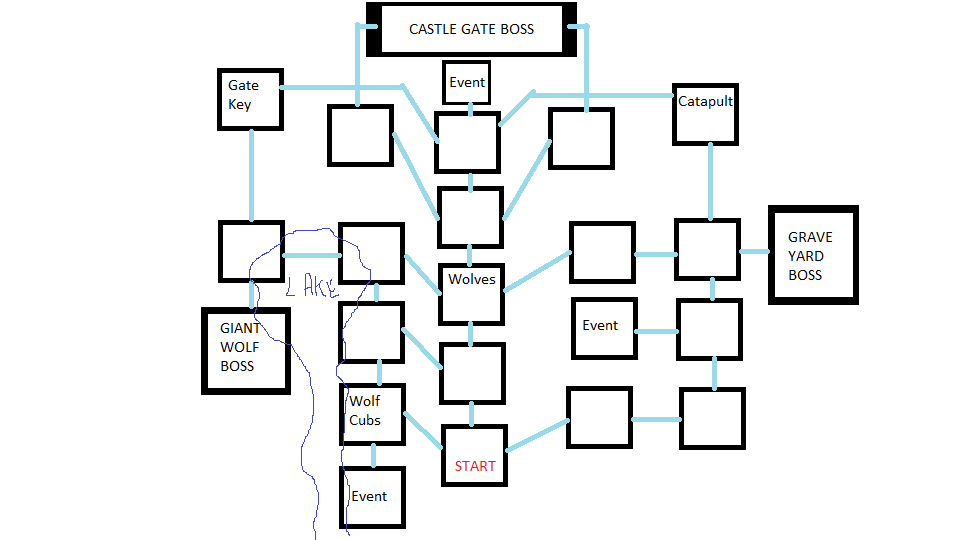


The character selection screen will show 16 characters. These characters will be separated into four categories based on primary role or stats. Strength characters are all front line fighters and tanks. Dexterity characters are your more well rounded rouges and ranger types. Magic are a combination of offensive and defensive magic classes. Utility are characters that stray from the norm in a significant way and are generally far less useful in combat but provide a significant bonus in another way. See character classes section for more details.

All 16 characters should not be available when first playing the game, but should be unlocked through rescuing NPCs from various locations found around the dungeon. This will create an additional feeling of permanent progression as players advance in the game. At the start two characters from STR, DEX, and MAG should be available giving only a small amount of variance in team structure. However each of the three bosses in the first area should unlock an additional character from each class giving a total nine characters to choose from once a player has defeated each of the starting area bosses.

When a character is selected in this way the right side of the screen will show information on them, a basic blerb that describes the character, their basic stats, and a selection of starting skills. After a progression system is implemented for the characters, each character should receive two points to use at the start of the game. These points can either be used on skills to start the game with or upgrading a piece of starting equipment. This equipment list has the potential to change as characters unlock secrets within the game, giving a degree of permanent progression. However, none of these items should be needed in order to beat the game, only provide a different early game advantage over choosing a skill level up.

**Zones:** The game will be played by players advancing through a series of rooms that make up a zone. Each zone will have a unique tile set with enemies, encounters and bosses. The structure of a zone will never change, allowing seasoned players to plan a route through the game to attempt to reach late game bosses, while allowing newer players the opportunity to explore and learn about the world. While the layout will not change, what is encountered in each room will be randomized to a point. Certain rooms will always contain certain things or certain types of encounters. Below is a sample map for the first area.



Each room the party moves into will have some sort of encounter. These encounters can be a straight up fight, an NPC or NPCs interaction (which could devolve into a fight), or an event which can consist of any number of non combat related encounters. Rooms marked with the event key word are guaranteed to provide a certain type of event within that zone. For the starting zone, there are three such events that will be placed randomly within the three event rooms. Some events or NPCs may be useless or provide negative or diminished returns unless other rooms or items have been visited or acquired by the party first. Certain rooms will have a guaranteed fight inside them. On this map we have two such locations marked as Wolves and wolf cubs. In these two rooms you will always encounter these types of enemies, however the number, mix and starting location of those enemies will be procedurally generated to ensure repeated play throughs don’t become repetitive in those places.

Each Boss in this first zone can be reached in as little as 5 rooms. However, there are items, events and enemies within the zone outside the fastest path to each boss that will make tackling those bosses easier. The giant wolf boss for instance, will be more difficult if one of the two wolf encounters was done, but easier if both were done. The castle gate boss can have the terrain altered by using the catapult or gate key. The graveyard boss can be made significantly easier with an item sold by an NPC who is found in one of the randomly placed guaranteed event rooms. Defeating the boss generally grants a boon of some kind and moves the party into the next zone. Each boss guards the entrance to a different zone.

The overall world will be set up like a zone made out of zones. At launch the intent is to have 3 world bosses available for the players to defeat. These bosses are killed permanently when defeated and also create a permanent change in the world. Killing any of these bosses also results in a successful run. Defeating all 3 world bosses unlocks the ability to go after the final boss. Expansions could be released that add on new zones world bosses and final boss. An additional challenge mode could also be implemented where the players have to defeat all world bosses and the final boss in a single run (impossible mode essentially).

**ROOMS:** The direction the party enters a room from should be dependent upon the direction they traveled to get there, so they can arrive from the top, left, right or bottom of the screen. When first entering a new screen, characters should be sown to walk onto the map from outside the viewable camera to predefined points on the map for entrance on that side. Most rooms will begin with a small cutscene of the characters remarking on what they see in the room, possibly moving around and doing things (however these mini cutscenes should be an option to turn off so veterans don’t end up getting annoyed if they have seen it many times already).

When players complete a room with either a combat or non-combat encounter they may be presented with an additional minor cut scene, or may be thrust directly into another event. When the room has been fully resolved, the player will be presented with the choice of which direction to move. This will be done with a set of cards that show a picture of the zone with a brief description of what is in that direction. Selecting a card moves the party to the next room and then we repeat.

**Non-Combat Encounters:** These encounters can be with scenery, objects found along the way or NPCs. These encounters are designed to play out like a choose your own adventure scene. The actors in the room will do something and the player will have options on how to interact with the scene or the NPC based off the characters he has in his party.



Different graphics will be used if the character has something he wants to say (which could be to the other characters in the group if not an NPC interaction) or action he wants to take. For example when talking to an NPC shop keeper two characters may simply want to ask what he has for sale, while if you braught the hunter he may recognize something the merchant is wearing an want to mention it which could lead to a discount with another dialog choice, while the rouge may have an action choice to stab the merchant in the back or attempt to steal from him. Having certain characters in the group may actually even override certain dialog choices, such as the assassin character mistaking an NPC for the man who killed his father and attacking him for vengeance unless a very specific set of dialog choices are made to talk him down.

Some choices will be characters specific and can only be taken when that character is present in the group, while others will be generic and can be randomly assigned to any character who does not having anything unique to say or do for interaction with the object or NPC in the scene. Left clicking on a speech or action bubble will select that choice and the scene will progress from there. Players should have the ability to mouse over their characters to check current stats / hp / equipment during these encounters as current health etc, could play a role in how they decide to approach certain situations and we don’t want them to regret sitting on the blood throne with their mage because they forgot he only had 1 hp left.

**COMBAT ENCOUNTERS:** Combat encounters will begin with prepare for battle stage. This stage simply consists of a free move phase for the player. Because their units will not be strategically organized after the mini cut scenes and predefine walk on points, the player will be able to either (based on play testing) freely move his characters around inside a deployment zone, or move each character half their normal move distance from their current location (this second method could provide opportunities to create more tactical depth by forcing certain characters out of position due to cut scene interactions etc). This phase could be skipped all together for ambush encounters and such as well.

Turn Order Computation: Combat will be done in a turn and round based system. Every unit in the fight has a number of available actions, two in most cases. At the start of each round the game will compute the turn order based on the number of actions each side has available to them. Whichever side has the most number of actions has the advantage and will get both the first and last turn in the round (in a tie, the player is given the advantage unless a specific encounter overrides this rule).

The number of turns as well as the number of actions that can be taken in a turn will be variable based on the size of the teams. The maximum number of turns either side can have is five, and the logic will always attempt to give two actions per turn when possible. The turn order generated at the start of the round will be displayed at the top of the screen with simple circles. As actions are taken these will go grey.



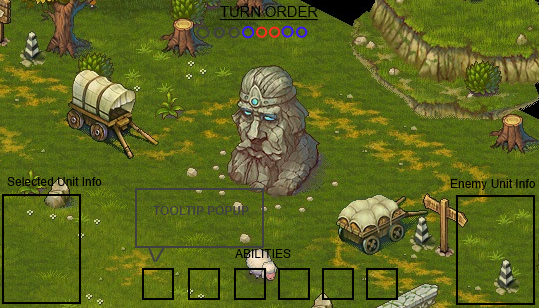
Player has more actions than enemy. Enemy could be either two units with two actions per round or three enemies with the standard two actions per round.



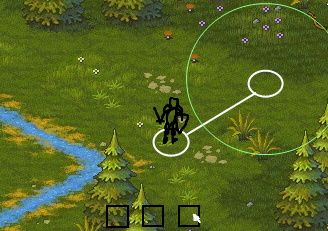
Player and enemies have same number of actions in this example. The player wins the tie and has the advantage and must have the first and last turns. In order to maintain two actions per turn where possible, the enemy gets less turns but more actions per turn.

When all actions have been taken a new round begins and the turn order is recalculated.

Units that are killed during a turn will still take their remaining actions, but the only action they may take (with a few exceptions perhaps for certain types of monsters) is the death’s door action, which does nothing. Note that any actions that can be taken to resurrect units or events that bring new units into the fight should only occur between rounds. At this point in development it seems to complicated to be worth the additional logic coding to add in new actions or have to consider re-ordering the turn on the fly.

At the start of the player turn, the closest player unit to the current location of the camera (who still has actions remaining) is automatically selected and the camera moved to focus on them. The player is then free to choose any character and proceed with their turn.

UNIT ACTIONS UI: Player units will generally have 4-5 abilities and 1-2 passives that for a total of 6 total skills (this could change based on play testing). The abilities for the selected unit should be visible at the bottom of the screen. Mousing over the abilities provides a pop-up tooltip that provides detailed information on what it does. Left clicking on an ability selects creating a range circle / arc / line / whatever to show where it can be used from the players current position. If the character right clicks for movement with an ability selected these range indicators should follow their intended destination to show the player what would be in range of that ability if they moved to that position. Any units that can be effected by the ability that are inside a range indicator should be highlighted in some way to reinforce this as where their hit box is may not always be obvious. Abilities can be deselected by left clicking on the ability again, left clicking on the unit or selecting another ability or unit. In order to activate an ability, the player must not have any movement queued up, and then must select the target of the ability, or placement of the indicator for AOE type abilities with a left click to execute it.





We will want the ability UI blocks to automatically pull from the selected units list of abilities to populate them, and then execute script based on the parameters for those abilities. NOTE that many of those abilities will act in unique and interesting ways (Abilities that let your barbarian pick up terrain pieces and throw them at enemies is more fun than swing sword do X damage).

MOVEMENT UI: In addition to their abilities each unit can also move. Move is not listed as an ability, but can be accessed simply by right clicking on the map with any unit selected. A right click will automatically generate the most direct path to that point for the player. However, this may not be the best path as it may go threw terrain that slows the character down or applies other status effects, or walks them by enemies that could attack them on the way by etc. To allow players to avoid these things, waypoints will be utilized.

**Right Click / Hold right click**: Delete last waypoint and pathfind to point. On waypoint to delete waypoint.

**Shift right click**: preserve last waypoint and pathfind to point. On waypoint to move waypoint.

**Left click**: on final waypoint to move, on unit to reset path and deselect abilities.

Additioanl items of consideration when coding the movement system should be potential features such as terrain areas that reduce movement points when going threw them (Character normally moves 10 units, but moving through water costs 1.5 moves per meter traveled or some such). Units can run over traps that could slow them or immobilize them in the process of movement. So a character may not actually reach the destination they set out for when they committed to their path based off what the unit encounters on the way there. Some characters could also have the ability to traverse certain kinds of obsticles that most can’t (like a rouge ability to climb walls, or a nimble character who can jump over objects that count as low cover or have a certain height value, or a giant who can push certain objects out of his way while he runs). Some sort of visual indicator should be implemented to show where these abilities are used, or when a character ill knowingly enter a terrain effect (like changing the color of the line for the portion that is inside a terrain with an effect).

**COMBAT RESOLUTION:** The method for resolving combat attacks will likely change significantly over time with play testing. To start we will keep it simple with automatic hits that have damage reduced by a set value of armor. Then increase complexity over time. Features that are being considered: Defensive stats for dodge block and parry, character stats such as strength or dex being compared against each other to effect chance to hit or damage results, flanking, getting locked in melee combat with free auto attacks going off every so often, chance to hit reduced based on range, being in cover, etc.

This section will be updated as decisions are made on what features to implement. Characters who are reduced to below 1 hp will either be knocked out with a chance to save them before the die forever, or just killed our right.

THE STACK: The resolution of each turn should be done through a stack method. The stack should be able to pull information from characters that are effected by actions to see if they have any abilities or equipment or status etc that would change the outcome of the ability being resolved. Using a stack would also allow for units to ready actions to be used when certain things happen around them (think over watch from X-Com). Some abilities may come with charge up times as well and so may result on the same turn that another unit takes it’s action. As abilities that need to be resolved check for and trigger additional effects that need to be resolved they will all be added to the stack and resolved in order of a stack (ie first in last out). Not sure at this time if there would be cases where we would want an ability to resolve in a different order, but should probably code it with that ability just in case.

**AI:**

**OTHER FEATURES TO CONSIDER:**

* Level up system or only progression through equipment found??

**ROOM / ZONE IDEAS:**

goblins and traps

the great wolf mother mini boss

graveyard and skeletons

the mists and the ghosts

necromancer mini boss

the experiments / chimeras

mad scientist laboratory

another adventuring party

the banquet hall

the smiths forge

lava cave

icey halls

spiders den

the great duelist mini boss

gholem factory

orc raiding party

orc strong hold entrance

orc armory

orc warband leader mini boss

sewer entrance with rats

mole people village

mole men engineers

the mole king mini boss

the long hall

knights of the lady midnight

the mirror of midnight

her royal advisors

the lady of midnight (first installment boss)