Introduction: This game project was inspired by Knossos Palace and Cretan mythology. The level keeps the minoan aesthetic by using red, gold, black, and sandstone as its primary colors, as well as having minoan age lighting and design elements.Minotaur's Might takes the player on an adventure as they play as a Cretan guard patrolling Knossos Palace during a minotaur invasion. It is the player's job to find the king in the palace and help escort him out of the palace to safety. In this third person stealth action/adventure level players will avoid patrolling minotaurs, dodge their attacks, complete puzzles to unlock abilities, collect relics, help a citizen save their cat, and save the king. This all takes place in a spacious 2 story palace with a hulking minotaur patrolling the courtyard.

Link To Gameplay Video: Minotaurs Might | Gameplay Walkthrough | Level Design

https://youtu.be/MhVN1UG2L1k

What Went Right:

State Machines - The greatest victory I had was setting up my state machines. I set up the state machines in the ways I had in the past for previous projects and was a bit hesitant that it wouldn't serve the AI in this map well since it is a close quarter map, but the AI functioned the exact way I wanted them to in the end. I had a hiccup while implementing the patrol state and getting the minotaurs to navigate, as I will mention below, but after surpassing this obstacle I was able to get the state machine working in a way that was easy to manipulate and easy to add to.

Using maps and references for level design - In order to add more realism and immersion I wanted to keep the same general outline and design of the palace, so I opted to use a map of the ruins found online from a cretan archeology website. This allowed for rapid 2D designs and freed up more time for me to focus on objective placement, level flow, and planning each route. This in the end worked out to be a great advantage and allowed for a more malleable and versatile design early on in the planning phase of the levels design.

Hit Mechanics/ Loss Condition- When designing the level in the first sprint I was hesitant when considering adding the minotaur AI because I was hesitant that the animations wouldn't interact well, or that I would run into an issue when implementing the player getting hit. I started by trying to do a sphere trace at a certain time in the minotaurs attack animation. This allowed me to get a base idea for how I wanted the mechanic to be implemented. Then I tweaked this by having the minotaur do a sphere trace at a location that I made using an empty container attached to the Minotaur's fist. This allowed for accurate hit scans and really became advantageous when working on the dodge timing mechanic.

Dodge Timing - While making the dodge I wanted it to feel like the player was actually jumping out of the way and evading the attack, much like in the dark souls series. I knew that with the size of the minotaur's arm and the size of the sphere trace that even if the player were to perform a well timed dodge, that it would still trigger the sphere trace. To keep the dodge

timing aspect I decided to add a bool that became true at the beginning of the dodge sequence and turned false after a delay. I added a variable to the delay and this became my evasion timer. This allowed me to tweak the dodge mechanic much finer making it so the player needed to time their dodge to avoid the attack within x seconds.

HUD Feedback & Ability Feedback - After playtesting the after the first week of blockmesh iteration I knew a major key to this level would be the player feedback. I focused a lot on setting up the interaction feedback by giving certain interactables text overhead while opting to use interaction text on the HUD for other scenarios where the overhead text would be blocked like NPC interaction. The HUD feedback such as the 'New Ability Unlocked' animation (when unlocking abilities), '+1 collected' animation (When collecting coins), death animations, and the ability icon progress bars, all played a part in helping to grab the players eye and notify them when they have achieved something or performed a task. This helped tremendously with replayability because players were able to catch certain things one time around like unlocking all of the abilities and would want to go back through the level again to unlock both abilities.

What Went Wrong:

Overscoped - I originally scoped to design a ~10 room palace level with AI enemies, a dodge system, a main objective of helping an NPC escape, an optional objective to help an NPC find his cat, 2 different puzzle systems that allowed players to unlock 2 player abilities, destructible walls and the list goes on. These design features piled up quickly over time and throughout the first initial week of blockmesh I was more focused on designing the palace as close to the 2D layout as I could, while making tweaks along the way to help with flow and allowing space for the player to move throughout. The second sprint I was more focused on getting each of the mechanics functional with a solid feedback base and placed well enough that players will be drawn to each of these features, but as mentioned before the list piled up and I begin running through each of these systems and trying to iron each one out individually and eventually I ended up needing to cut mechanics during the second sprint and lost a lot of time that could've been used to help design minor details to help guide the player and help build more story building in the environment.

Issues With Destroying Destructibles - As mentioned with the prior issue this mechanic needed to be removed because of issues triggering the damage. I have worked with destructible meshes and destroying walls with projectiles in the past and thought nothing of adding this into the project to work more with the chaos destruction system in UE5. With the prior knowledge of the system I thought I could work out a nice level mechanic, but instead it ended up being more of a hassle because I needed to implement a system where the player can grab an explosive jar, then throw the jar, and how can the player aim the throw without a path prediction line, so all of these elements began adding up with this mechanic. Before too long I realized that in order to make this deadline I would need to cut this mechanic and focus on the other mechanics presented in the level. The one upside of the early implementation of this mechanic is that I have a throwing and projectile path prediction system in the game, but it just needs to be implemented and tweaked in a later sprint.

Patrol State Not Triggering - A major issue I ran into early on was with the Minotaur AI's patrol state moving to its first location then not resuming its route. The way the patrol state is set up allows the designer to just place navigation points around the map and use the eyedropper tool to select specific points. I placed all of the points throughout the level and added them to their rightful minotaur and ran the game, but realized they weren't moving after their first point. I tried stepping through the state machine and ended up noticing that the overlap wouldn't trigger and this led me to realize that the nav points need to be larger in order for the minotaurs to trigger their overlap statement. This issue ended up taking a few hours of time that could've been used elsewhere in the design.

Gates and Valves - After playing getting the minotaurs patrolling and taking the destructible walls mechanic out of the level for the time being it left the level very open between rooms. Originally the destructible walls acted as a gate system between rooms and would help guide players, but without this mechanic it left the guidance a bit astray. Another issue that I noticed after playtesting was that players would backtrack throughout the level and lose their bearing while trying to avoid the minotaurs. The addition of a few strategically placed gates would have benefited the level tremendously.

Guiding the Player - As mentioned before, guiding the player was a major issue after the removal of the destructible walls. I noticed after the first few playtests that I ran, that players would backtrack often and often got lost throughout the level. While I wanted to add in some aspects of exploration, I would've liked more exploration with a purpose and having the various unfilled spaces left this palace looking more uninhabited rather than under siege. With the workload of the various mechanics I lost time that could've been used to improve defining the paths with minor and major landmarks to help the player find their way better. Another solution that I considered adding to alleviate the issue of players getting lost was the addition of a minimap on the HUD in combination with a visual checkpoint system telling the player the initial direction and distance to their next checkpoint.

Conclusion: When starting the project I had my reservations, while some were well warranted like over scoping, I was able to overcome many others and present an enjoyable level. This level is a great base to build on and I am excited to add and tweak it to bring it together more. In summation, this project was a great learning experience, as well as a great experience used to build a new environment and level with different new mechanics that I haven't implemented before.