**Slide 1:** I’m Alex and I’m the TL2+ handling the level design and level loader.

**Slide 21:** Once again, I am Alex. For the game, I handle the level builder and the movement controls for the playable character. My job will be twofold. First, it involves returning results from player inputted controls to the player. The second part involves acknowledging and building the desired level from the game save system, and sending the relevant level data and level layout to the scene engine for backgrounds and music, the NPC manager for which NPC’s should be placed where, and the collectables manager for where collectables or powerups should be placed. Additionally, the level design in its final state will be sent to the game state data store that retains all relevant information about the game scene to make the game visible and playable for the player. That store also enables a saved game to be created in conjunction with the game save system.

**Slide 23 pt 1:** Again, I’m Alex, I’m handling level design. So, once I get the level data for the new game from Bidhan and the main character information from Mark, I start building the level described by the save data, Level 1 in this case. I place all the terrain objects, passageways and the main character into the scene, making sure the player can move around and the environment reacts properly to interactions with it. Then I pass the level data and design at its current state to Hengyi to handle the background design for the level… (Hengyi’s part)

**Slide 23 pt 2:** Once I receive the updated scene information with the new background appearances and music from Hengyi, I move on to the placement of NPC’s for the level. For this, I communicate with Todd, passing along the level data to determine which NPC’s to place in the level and where they belong along with any additional information specific to those NPC’s…(Todd’s part)

**Slide 23 pt 3:** Now that I have the updated level information with the NPC’s locations and information available for me to place them, I ensure they are placed in the proper locations in the level and are able to be interacted with by the player character. Then, I move on to add the collectables to the level, passing along the level data and map design to Shan so that she can do a similar process of determining the collectables to be placed in the level, their locations, and their attributes…(Shan’s part)

**Slide 23 pt 4:** After receiving the level data back from Shan and adding the collectables to the level the same way as the NPC’s, all core elements of the game have been added to the level. At this point, the new game has been completely generated and control is passed over to the player allowing them to play the new game.

**Slide 31:** Ok, so once again, I’m still Alex and I handle the level builder that will design and create the base maps, layouts, and levels players move around in and the other features build on top of. This is my Diagram 1 for the level builder. Basically, I take the information from the game save system designed by Bidhan and use it to activate a level map loader which will determine which level needs to be generated. Then it calls that level to be designed, building in all the scenery and terrain objects that both help make the game playable and give it interesting design features the player can navigate through in the game. The levels interact with the player when the player chooses to move around, taking in the movement inputs the player gives and returning the resulting position change or effect from the terrain. All of this information is saved into a game state data store that holds onto all the information about the current game scene.

**Slide 32:** For my Diagram 2, I explore specifically level 1. Level 1 allows a player to interact with the NPC’s in the level leading to a locked passageway being revealed which will then move a platform to allow the player access to the passageway. The player can then move through the passageway to move to another level. The player may also interact with dangerous environmental elements that can deal damage to the player or they might interact with moving platforms that will change the player’s location once they move on top of the platform. The position for platforms are saved directly to a positional data store that holds that as well as all information about the player’s position. A separate save data store tracks the state of the locked passageway and the player statistics outside of position.

**Slide 33:** For my use case diagram, I have a game save system sending the level data for a level into a level map manager or loader. That will then call level one by default or it will call one of the other levels if the save system indicates that the saved game is on one of the other levels. From there, the level will set the objects including passageways, moving platforms, dangerous environmental objects, locked passageways, sliding doors, and scenery objects unless the level has already previously been visited in which case the locked passageways may be unlocked and the sliding doors may not need to be placed.

**Slide 34:** So, overall I would give my complexity for the level a 10 and a priority level of 1. The level design and loader sets the scene for the entire game creation, so it handles the placing and calling of almost every other feature making it essential for the game. Since it has to build a level and also has to interact with so many other features, the complexity level increases, and while that might be simpler for some people, I suspect this will take a bit of time to properly organize and sort out and it’s crucial it’s laid out properly so all other features work as intended.