**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 includes responding to player inputs and loading game levels based on what I receive from the game save system. Then I call the scene engine, NPC manager, and collectables manager to add backgrounds, NPCs, and collectables to the level. That information as well as the map layouts I send to a data store that holds the game state for a saved game.

**Slide 23 pt 1:** 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. 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 have the updated scene info from Hengyi, 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:** With the information I receive from Todd, I properly update the level with the NPC’s. Then, I move on to add the collectables to the level, passing along the level data and map design to Shan to similarly determine the needed collectables information for the level…(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 the basic elements of the game have been added to the level. At this point, the new game is designed and playable.

**MY INDIVIDUAL SLIDES**

**Slide 30:** Ok, so once again, I’m still Alex and I handle the level builder that will make the base levels players move around in and 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 to generate the correct level. Then it builds all the scenery and terrain objects that make the game playable and make it interesting to move around in. The levels interact with the player when the player chooses to move around, taking in the movement inputs and responding with their result. All of this information is saved into a game state data store that holds onto all the information about the current game scene.

**Slide 31:** For my Diagram 2, I explore specifically level 1. Level 1 terrain objects involve a moving platform the player can stand on, a locked passageway, and some dangerous environment elements. The player can interact with NPC’s to unlock the passageway and automatically move a sliding door to go through the passageway to the next level. Standing on moving platforms will change the player’s position and touching dangerous objects will decrease their health. All the resulting positional data for the player and platforms are stored in a positional data store and all data about the player health is stored in a save data store.

**Slide 32:** This is my class diagram which breaks down into a level map system that holds all the different levels with their unique layouts and a collection of terrain objects that are used by the level map for each level. Terrain objects can include damaging environment objects, normal passageways or locked passageways, and moving platforms that can be modified as sliding doors.

**Slide 33:** For my use case diagram, I have the game save system sending the level data to a level map loader that calls level 1 unless instructed otherwise by the save system. From there, the level sets the 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 feature complexity a 10 with a priority level of 1. The level design and level loader set 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, it could be complex to manage and it’s crucial it’s done right for everything else to work.