Player Experience – This game will be a strategic, third-person, platformer game. The player will have to determine what actions they will need to take to reach the end of the level.

Core Mechanics – Player movement(Running, jumping, sprinting, etc.), currency system to purchase abilities for each level, and score system for how quickly the player completes the level & how little currency they spend.

Gameplay Loop – The game starts with the player choosing some abilities and modifiers before the level begins. Then, the player loads into the beginning of the level and has to reach the end goal as quickly as possible.

Mechanics/Tools in this game:

1. Player Movement. Movement for moving horizontally & vertically. Vertical movement would consist of jumping mechanics. Horizontal movement would include running, sprinting, etc.
2. Platform behavior. Instead of enemy behavior, it would be platform behavior. Example behaviors would be moving along a given path automatically or when the player steps on the platform, Rising and falling platforms with very similar logic, a bouncy platform, a platform to shoot you in a specific direction (ex: Celeste magma blocks), and more
3. User Interface. User interface to display abilities, timer, scoreboards, main menu, etc.
4. Scene Transitions. Scene transitions to move from the central hub to the levels and back as well as from the central hub to the main menu and back. Scene transitions in the main menu to quit the application, mess with some settings, and transition to the central hub.
5. Singleton pattern for player score, level score, etc.
6. Persistent data between sessions. A way to save the player’s information for the levels, their controls, and progress in the game. Information can be saved to and grabbed from the singleton objects.

Timeline/Milestones:

1. Get the third-person camera to work properly. Get player actions and abilities up and working. Get proper scene transitions & have the singleton for player’s currency working.
2. Design levels, get persistent data between sessions, fix issues with the code.
3. Implement particle systems, sounds, music, and more QOL features to the game.