Christopher Gundlach

This project went surprisingly smoothly, from the planning to the actual coding and execution. Once I had gone through all the requirements, the idea that stuck out to me was doing a runaway train game. I knew it would have six different cars, items in each and a player. So, I started simply with the abstract Space class. Instead of overriding and repeating multiple functions within each derived class like I did for the fighting tournament, I made one pure virtual function, the challenge function, and all the rest were inherited straight from the Space class. This allowed me to build each car class. Once I had the item and player classes, I filled in the car classes with each challenge and the items. To tie it all together I knew my Train class would function as the linked space structure. I made pointers for each car and the player, and used those pointers to run the game. Once I first compiled it, I had a couple pointer problem that I needed to fix. After that, everything compiled and worked. For a project this large it may seem surprising, but I was pleasantly surprised at how well I had planned and put it together.

Test Case	Input Values	Expected Outcomes	Observed Outcomes
Exceed Steps	Keep changing cars	Game ends after 15 steps	Game ends correctly
Win game with correct item	Pick up right item and use it	Win game	Win game
Try to exceed pouch limit	Pick up 7th item	Item not added to pouch	Item not added to pouch