Techincal Report and Concerns

Alejandro Nuñez Barrera

Arcagames project is an application designed for managing arcade machines, allowing users to purchase, customize, and enjoy different types of gaming machines. The system supports various machine types, each with its unique attributes and configurations. Users can also manage the addition of video games to the machines, with the price of the machine adjusting accordingly when games are added or removed. This report focuses on the technical aspects of the project, including the architectural choices and the use of design patterns such as the Factory Method.

The overall architecture of the Arcagames project is modular. Each module is responsible for a specific domain in the application. For example, machines.py manages the different types of arcade machines and their respective attributes, while videogames.py deals with the video games that can be associated with the machines. The users.py module manages the user-related logic, and main.py serves as the entry point, orchestrating interactions between the various modules. This modular design allows for scalability and easy maintenance, where different parts of the system can evolve independently without breaking other components.

One of the key technical highlights of this project is the use of the **Factory Method design pattern** in both machines.py and videogames.py. In **machines.py**, the Factory Method is used to dynamically create instances of different types of arcade machines based on user input or predefined configurations. For instance, if the user selects a "Dance Revolution" machine, the factory will return an instance of the DanceRevolutionMachine class. This mechanism allows the application to handle a wide variety of machine types—such as shooting machines, racing machines, and virtual reality machines—without modifying the core logic. The implementation of the Factory Method ensures that adding new machine types in the future would be seamless, as the factory can be easily extended without changing existing code.

Similarly, **videogames.py** uses the Factory Method to handle the creation of different video games. Video games can vary in type, resolution (SD or HD), and other specific attributes, depending on the machine they are assigned to. The factory ensures that the right type of game is instantiated based on the machine’s category, ensuring that only compatible games are added to the machine. This guarantees that games for a "Dance Revolution" machine, for example, will have the right set of attributes and features, like the correct difficulty levels and gameplay modes.

Another significant feature of the Arcagames project is the ability for users to customize arcade machines. Each predefined machine type comes with standard attributes such as material, dimensions, weight, power consumption, memory, processor, and a base price. Users can customize the material of the machine, which affects properties like weight,

power consumption, and overall price. Additionally, users can add video games to these machines, which will automatically adjust the machine's price based on the game's cost. The system also accounts for the removal of games, decrementing the price of the machine accordingly. This dynamic pricing mechanism ensures that the machine’s cost always reflects the value of its components.