Req 4

- Please refer to the requirement 4 UML diagram
- We are assuming that every ComputerTerminal sells the same thing

Pros

- ComputerTerminal calling the buy method for each of its buyables to create a BuyAction follows the SRP as ComputerTerminal is just responsible for displaying the BuyActions, not creating them (creation is done by the Buyables themselves).
- The use of the Buyable interface makes the system easily extendible as new Game Entities that can be bought can just implement it and have the common functionality of buy, finalCost and transaction, following the OCP.
- LSP is achieved again by the Buyable interface, as any object of type Buyable can have a BuyAction created for it by the ComputerTerminal.
- ISP can be achieved in the future when selling of objects is introduced. The Buyable interface only allows objects just to be bought, not sold as not every object to be bought can be sold. The selling of objects would be a new interface, Sellable.
- Connascence of position is reduced in each of the Buyable children as their attributes (name, displaychar, portable) are instantiated inside of the constructor and passed to the super instead of passing it through the object's parameters when instantiating the Buyable.

Cons/alternatives

- DRY is sometimes violated due to the use of the Buyable interface, as seen with the successful transaction message and buy method. An alternative would be to make a BuyableItem abstract class and then have UsableItem and WeaponItem inherit it to have BuyableUsableItem and BuyableWeaponItem, adding common logic of the transaction and buy methods as concrete methods. EnergyDrink and TolietPaper would inherit the former while DragonSlayerSword the latter. However, this introduces a lot more complexity as there's a large hierarchy of abstraction, thus this alternative was not pursued.