Info3202: Assignment 1

1. Application of Abstract Factory Method and Builder Method

To utilise these methods I focused on the two main products of this first iteration of the pool game; *the table and the balls.* By placing these I was able to expand using design patterns from there.

The abstract factory method was created so that a client could initialise a game without being exposed to the working underneath. Firstly, an interface “factory” was used to hold abstract methods of the Stage1Factory to use. The reason I chose this was because in the next iterations of development, further stage factories can be implemented without sacrificing the structure or the product structure. Secondly the products were divided into generic and specific types with Ball and Table being the generic and “Stage1ball” being inherited. The reason for this is because I foresee potential implementation changes especially when dealing with the white ball, so it was better to create a generic type for later. The table class for this iteration I thought was unnecessary to use a specific class since additions to the table can be easily added onto the class such as holes, bumpers etc.

The concrete factory “stage1factory” extends to both products, however when dealing with the creation of the balls a builder design pattern was implemented. The stage1factory calls upon a Ball director to oversee the construction of the balls that were passed through via a JSON file. The builder is set up so that the generic product is ball with a generic builder that supply common methods. The instantiation of the specific stage1ball however come from the stage1BalBuilder. The builder is constructed as to allow any inherited builder to be called on by the director by calling the generic builder.

1. Advantages and Disadvantages of the Design Patterns

*Abstract Factory Design Pattern:*

AdvantagesThe abstract factory by nature of separating the concrete classes from the client allows development of the classes separately to the client as we can see in the UML diagram below. In addition, since it allows the creation of multiple factories it allowed the two products to be developed separately but readily accessible and can be easily extended by building more factories.

Disadvantages: With regard to this iteration of the development however the design pattern may have been overkill. Due to only have one concrete factory and only one concrete version of each product there is a high overhead for a relatively simple implementation. In addition, if new products are added it will require further extensibility and may potentially lead to spaghetti code.

*Builder Design Pattern:*

Advantages: The builder design pattern is extremely useful if you suspect to have multiple instances of a certain class which would lead to many different constructors being created. However, with the builder decouples the construction of the object from the class and passes the responsibility of that onto the director class. As such many permutations can still be called using the same director.

Disadvantages:

The issue with the builder for the level of development though is the ball class only had one concrete class and as such was not really utilising the builder’s potential. This is something however that will allow it to be extensible for further development .

1. UML Design

