**ALTERNATIVE DESIGN DISCUSSION**

We have designed the class diagram for the famous Chewy Lokum Legend. However, there are unlimited ways of designing a program and its class-wise relations. We are going to mention an alternative design that we have thought of implementing, later we have crossed out.

In our program we have related, interacted and structured our board objects in a hierarchical object oriented manner as you can see in the class diagram. We have foreseen that we have to paint each different board object which belongs to the hierarchy’s lowest level (Obstacle, StripedLokum, ColorBombLokum, WrappedLokum, NormalLokum). The painting problem is the thing where we were lead to hesitation. As you can see in the class diagram, every lowest hierarchy board object implements the paintable class in order to be painted. The other alternative was to create an abstract class to paint them. We have hesitated between the abstract class idea and the interface idea for painting the objects. At the end we have crossed out the abstract class and used the Paintable interface instead. The reason we did this was a class can implement unlimited amount of interface. And if we have used Paintable abstract class we could not be able to make the board object hierarchy as been shown in our class diagram. We think that our class hierarchy is natural and is very open to polymorphism. But if we have used the paintable as a class instead of interface we would be bankrupt in terms of abstract class and could not be able to have natural abstract classes such as SpecialLokum and Lokum super classes since java does not have multiple inheritance. Creating abstract classes are easy to understand the problem for another coder and using the paintable interface makes the program more flexible for another interfaces.

As a result, we did not use the paintable as an abstract super class. We used it as an interface. It enabled us to make the program more readable, understandable, flexible and open to implement other interfaces.