**Flowcharts**

Flowcharts are a form of high-level designs to help convey the flow of a program step by step. Certain methods i.e. transactions, withdrawing from the game, turns etc. have been made into separate flow charts as multiple of these methods can occur in one cycle of the system. Flowcharts are a good way to break down a project into steps which can individually be coded. By viewing the control flow, such as a player rolling a dice for their turn, in the Player Movement flowchart, can lead to many different situations. The development team can find this extremely useful as they can see how the system interacts with the methods, along with how all the functionality inevitably links together. All high-resolution versions of each flowchart are available within the documentation folder under designs within the high-level section.

**Class Diagram**

The class diagram is an effective low-level designing technique that breaks down the system into entities (classes) that interact with each other (methods). This is an effective tool for programmers as they can see what classes interact with each other, along with what methods and variables are needed within each class. We felt like adopting a Waterfall approach to designing would ensure a well throughout design for the code would be constructed, eventually saving time in coding, and less time wasted refactoring code from previous cycles. As this diagram was constructed before any research on the AI was created, we just had the AI class extend the Player class. Using these types of diagrams, we can refactor non-abstracted, poorly designed code that can slow down system operations, possibly restricting updates to the code. In addition, to never using C# Visual Studio before, we were unaware on how the windows would work so researched into it slightly. Our finding showed that all windows are partial classes of the main window, so these were added to the diagram. A high-resolution image of this diagram is available in the design documentation in either image or PDF format named “Class for Monop”.

**Sequence Diagram**

Creating a class diagram for the menus from the high-level designs was quite challenging, so a Sequence diagram was also created to help try and illustrate how the user will interact with the windows. These diagrams show the interaction of objects in a chronological order. This includes showing objects and classes exchanging data, and what entity is responsible for the control of the flow.