# Reflective Report

## Creating the Model

Creating the model was a challenging aspect of this project. While the game itself was very simple in its design and rules translating those rules and constraints into a VDM model provided to be quite the challenge. For quite a while, I struggled with trying to create a mental model of the game and how various features could be mathematically and abstractly determined. For example, the simple concept of how the grid itself was simulated offered a range of possible implementations. Initially, I decided to portray the grid as a set of points. The user would then make moves and which would transfer the points from one set to another. However when it came to modelling a valid move this implementation added additional complexities as I now had to worry about moves wrapping around the grid. In the end, I decided to go with another implementation style (storing the grid as an x and y-axis). This was done in an effort to make the proofing process easier by removing any unnecessary complexities.

## Translating the Model

Once again, I had some troubles translating the model from the previous task. Thanks to the simplifications made during the model development process most of my model was easy to translate as most functions just consisted of basic Boolean operators. Things changed though when it came to my primary operation (makeAMove). Due to the complexity of the operation and how it called on multiple functions in different ways, it proved to be extremely difficult to translate that into Isabelle. In the end, I had to go back to the operation and see how functions could be grouped together or broken down. I managed to remove a large amount of complexity by creating separate swap turn functions and grouping the captureAnchor function and add move to play set function together. These simplifications then helped me to translate the operation.

## Issues Found During Translation

Due to the nature of translation, I was often combing through my model paying attention to every detail. This directly led to the discovery of several bugs or unintended side effects of my system. For example, while translating the invariant for the move type I discovered that its invariant could be broken if the user created a move on the minus x and y-axis. Therefore, I went back to the model and updated the invariant to counter-act this issue. Similarly, I found several other small bugs and issues while translating and I fixed those as I went along. Such as players being gaining double bonus turns, by simplifying and grouping functions together I managed to avoid this issued and removed the double bonus turns.

## Proof Finding

This was by far the most confusing aspect of the entire coursework. For most of the time working on the project, I had been entirely focused on creating the model of the game as well as learning Isabelle so that I could translate the model over. This meant that by the time I had finished with the translation I had very little time to learn how to complete the proofing process. Proofing itself is a difficult process that takes a long time to get a handle on so leaving myself so little time to learn it was significant downside of the project. I should have focused on learning all aspects of Isabelle at the start so when it came to proofing I at least had some ground to work on.

## Satisfiability Proofing

Eventually, after a great deal of playing with lemmas and theorems I eventually figured out the basics of created proofs. I then immediately set onto creating satisfiability proofs for all of my operations. At this stage, I also discovered that a few of my operations were missing post conditions so at this time I decided to add them in. Looking back at the model though in VDM I found that many of these new post conditions were already implemented through the state invariant requiring no additional work on the VDM model side. While I could have spent all day going through and constantly refining my model I eventually decided to draw the line and simply focus on getting the proofs done and hammering out the major bugs of the system.

## Sanity Proofing

My sanity proofing and satisfiability proofing went hand in hand so to speak. While I worked on the satisfiability proof cases I would often add in some sanity checks along the way or even combine the two cases. Such as ensuring that the state itself can even be initialized. This works as both a satisfiability check as well as a sanity check to ensure that the game can even be started.

## Improvements to the Model and Translations

While the model functions as expected; at least from my own tests. There is still a range of improvements that could be made. For example, if I had the time I would have implemented a function, which would automatically play a full game of dots and boxes. This could have saved me the time of having to manually input all of the moves in one by one in to the makeAMove operation. I could have also implemented more sanity checks and post conditions to the model and translations. However, due to the unexpected time spent learning and understanding Isabelle I did not quite have enough time to implement these features to the level and quality I would like.

## Conclusion

Overall, CSC3323 was one of the most challenging modules of the semester. It required a great deal of thought and creative problem solving to not only come up with solutions to nuance problems but to evaluate them against future requirements. I found it rather difficult to model the system at the same time as trying to figure out what would be easy to translate and prove. This led to some parts of the system being simplified and easy to prove while other parts being very complex and difficult to prove. If I were to do things differently I would have focused on proof far earlier as I found it to be the most complex and difficult topic in the module.

## Improvements for the Module

I found the mid-semester deadline to be extremely helpful for the development of my model and translation. One way I believe it can be improved by having a slightly earlier deadline for just the model. Much like the current system, the model would be marked at this initial early deadline and then marked again at the end. However, the student could be given plenty of feedback on their model regarding how easy it would be to translate and prove. This then gives additional time to make amendments to said model. In addition, more support in the proving stages would be extremely helpful as I found this, by far, the most confusing section of the coursework.