

# CSCI 1710 Capstone Project: Texas Hold'em

Dylan Lewis, Samuel Jana, Jullian Hooten

## 1 Background

In this project, I worked with Samuel Jana and Jullian Hooten to model a round of Texas Hold'em, a variation of Poker, using Forge. The modeling language Forge allows computer scientists to model systems using logic and constraints. This project models an opening round of a Texas Hold'em game between four different players. Our model tracks the state of each round, from the dealing of cards to the declaration of a winner, ensuring a robust framework for simulating and analyzing poker strategies.

## 2 Model

Our model progresses through the states of a poker game, which we call RoundStates. The RoundStates that we track and constrain are: preFlop, postFlop, postTurn, and postRiver. Each of these states has specific constraints required by the state; for example, in a postFlop state there must be exactly 3 cards on the board, all players have exactly 2 cards, and all remaining cards should be in the deck. Transitions between these states are governed by the validTransition predicate, which ensures consistency in both players' actions and that the RoundState is updated to the next RoundState appropriately. Finally, our model checks the 2 winning conditions of poker; a player is the final player remaining in the round due to all other players folding, or in the final RoundState all remaining players have matched bets and one player has a superior hand. All of these data objects and fields are tracked by our model, which you are able to see in Figure 1.

Figure 1: Visualization of our Complete Model

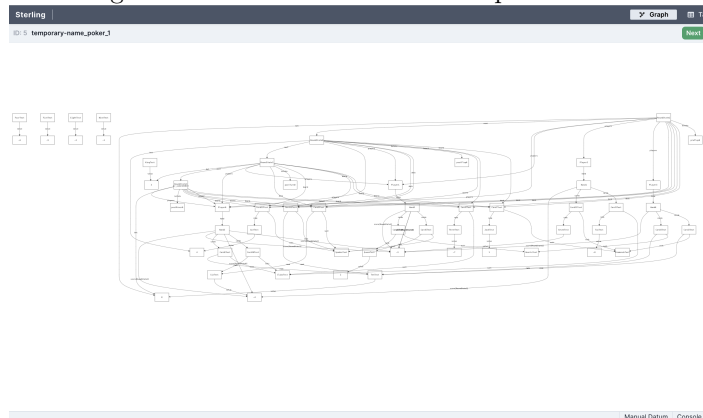
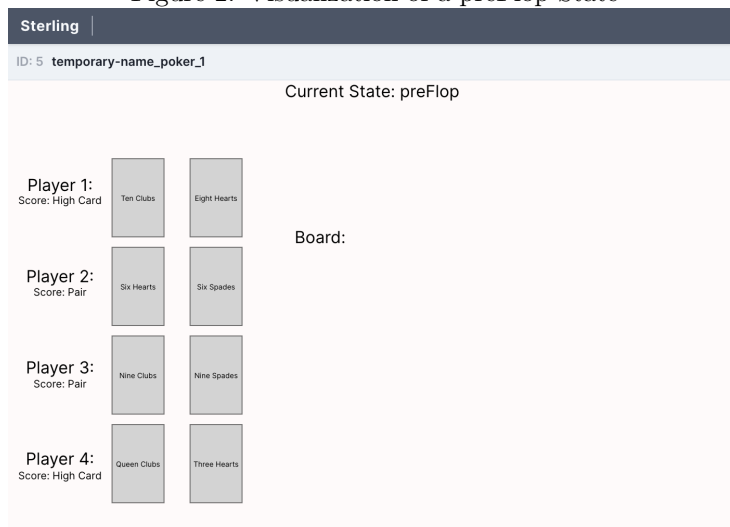


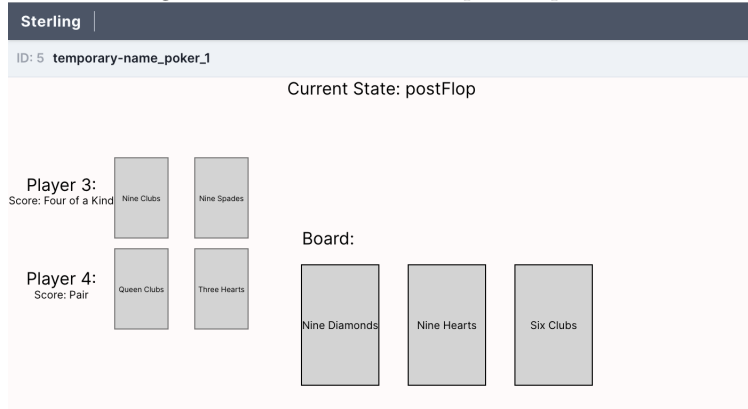
Figure 1 is a singular instance of our model visualized by Forge’s default visualizer. This image contains all players, their hands, their bets, and all information regarding each RoundState. As you can see from the figure, using Forge’s default visualizer to analyze our model is extremely difficult as there is so much information to track. This led us to the conclusion that we should create our own custom visualizer to display our model. In order to further simplify the output of the model, we made the decision that our visualizer should split each RoundState into its own page and only include vital information such as which players remain in the game and which hands they have. Below are several figures, each of which are an output from our custom visualizer, displaying a singular RoundState in one instance of our model.

Figure 2: Visualization of a preFlop State



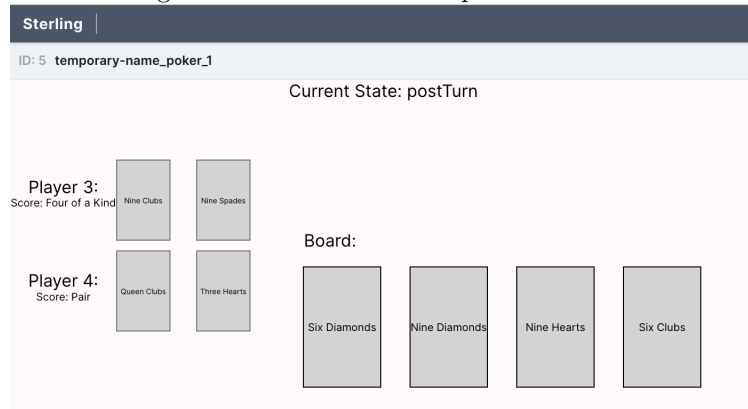
In the preFlop state shown in Figure 2, all players are dealt 2 hands and all remaining cards are in the deck. Each player's hands are evaluated, which as shown, Player 2 and Player 3 both have a pocket pair, while Player 1 and Player 4 only have high card hands.

Figure 3: Visualization of a postFlop State



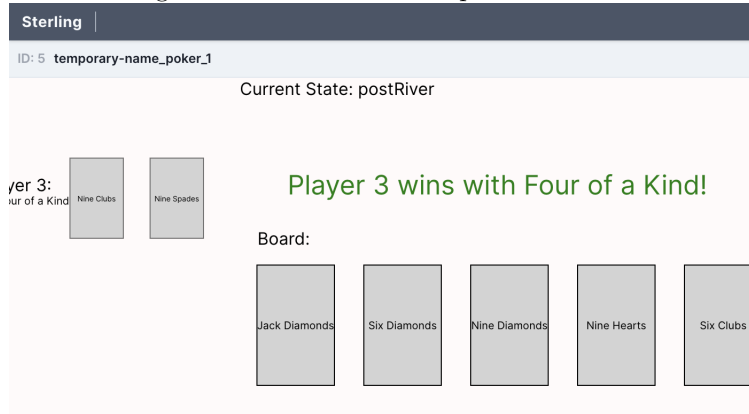
In the postFlop state shown in Figure 3, Player 3 has flopped a Four of a Kind, while Player 4 upgrades to a pair of nines. However, both Player 1 and Player 2 fold.

Figure 4: Visualization of a postTurn State



In the postTurn state shown in Figure 4, Player 3 and Player 4 hands' remain unchanged and a six of diamonds joins the board and misses both player's hands.

Figure 5: Visualization of a postRiver State



Finally, in the postRiver state shown in Figure 5, Player 3 wins the round with a Four of a Kind, due to Player 4 folding.

Following the completion of our custom visualizer, our model became much simpler to read and understand. Due to this, we could begin testing poker strategies and analyzing how they performed in our model. Here are the 5 poker strategies that we tested:

1. A player that never folds regardless of their hand and the bets made.
2. If after the flop, a player only has a high card hand, then they fold.
3. If after the flop, a player has a straight or better, then they fold.
4. If a player has the best hand in the game, then they never fold.
5. A player that will not fold regardless of their hand until they the river card is revealed, they then fold or bet in the final round depending on how good their hand is.

### 3 Findings

Our 3 main findings from our extensive strategy testing are:

1. The strategy of a player never folding their hand, regardless of the hand's strength, leads to the player winning more rounds of poker than most other strategies. However, many of their wins are with weak hands, such as a pair or high card. This strategy also leads to many lost hands, and due to never folding and always matching bets, this strategy, on average, leads to the most amount of money lost. Therefore, we do not recommend this strategy for poker players as it usually loses money; despite its ability to win rounds, other strategies would not win. A better version of this strategy, although not perfect, is strategy 5;

remaining in the round until the river and then deciding whether to fold or bet.

2. The strategy of always folding if following the flop, a player has a high card is surprisingly ineffective. While this strategy does minimize losses, it leads to many rounds lost that could have been won. For example, there were many instances where the player would have a high pair on the turn, a straight, or better on the river, which would have led to a round and chips won. As poker often proves, hands can change massively due to the turn and river cards, so always folding on the flop is not an optimal strategy. This strategy could be improved by accounting for straight or flush draws, for example if a player is only 1 card away from hitting a straight or flush after the flop, then they can consider not folding.

3. Our recommended strategy that we tested is a player never folding their hand if following the flop, a player has a straight or better. This strategy is great for 2 primary reasons. When a player using this strategy bets money on a hand, they usually win the round and the other player's chips. This means that this strategy minimizes losses as the player often only bets money when they eventually win the hand. Secondly, the player folds bad hands, which leads to a lack of losses, where the player loses money by betting with worse hands. On this point, in the very few instances where this happened, players were often only beaten when they flopped a straight and other players eventually hit a flush. This could mean a slight adjustment to the strategy by never folding; only when a player flops a flush or better could it be beneficial.