Concurrent Poker Player

Milestone Three

The Parallel Poker Team
Mark Jenne
Ian Roberts
Bennie Waters
Sarah Jabon

October 20th, 2009

Instuctor:
Sriram Mohan
Rose-Hulman Institute of Technology

Revision History

Date	Revision	Comment
10-09-09	1.0	Initial draft
10-20-09	1.1	Addressed Drew's comments

Signatures

Mark Jenne	
Ian Roberts	
Bennie Waters	
Sarah Jabon	

Contents

Part I
Executive Summary

The parallel poker player project involves designing and developing an application that exhibits the benefits of parallel computing [1] over sequential computing [2]. The application will showcase this through a demonstration of different computer-based players involved in a Texas Hold 'Em poker game [14]. Numerous key elements discussed in this document:

- Background Information - Essential information from Milestones One and Two
- Scenario - A description of an user's experience with the application
- Screen Shots - Images that represent how the application may look
- Decision Tree - A description of how the system will decide which player wins

The poker application will be a visually stimulating demonstration of the differences between parallel [1] and sequential computing [2].

Part II
Introduction

Milestone Three is the third document of the poker application's requirements and specifications. The contents of this document - scenarios, screen shots, and a decision tree - draw upon the core elements of Milestones One and Two, such as user needs, system features, use cases, and supplementary specifications. The scenarios in this document, paired with the supplementary specifications from Milestone Two, will help the team develop test cases. Based on the test cases, the team will develop an interaction prototype, which will be completed by November 2009. The design of the actual application will be determined by the use cases, supplementary specifications, and the interactive prototype. Once the design is established, the team will create a prototype. This prototype will be fully completed by February 2010. Finally, both the prototype and the incorporated design will be used to create the complete system. The system will be complete by the end of the academic year in May 2010. The objectives of this document are to concisely summarize user interaction with the poker application in terms of scenarios and screen shots. With this core information, it will be possible to develop aspects of the project in fuller detail in the future.

Part III
Background Information

Included here are excerpts from Milestone Two that contribute to this document.

1 System Features

The following section describes all system features that have been approved at this time. Refer to the feature scale in the appendix for information regarding the attribute values for each feature. The developer assignments have been revised for the features and two of the features were divided up per release. The graphical display feature has been divided into two features: a basic display for the first release and an interactive interface for the second release. The ability to support multiple automated players feature has also been divided into two individual features: support for only two players for the first release and support for more than two players for the second release.

1.1 System Features List

Feature Number	**Feature**	
1	Graphical Display	
Status	**Priority**	**Effort**
Approved	High	Medium
Risk	**Stability**	**Target Release**
Low	High	First release
Assigned To	**Rationale**	
Benjamin Waters Sarah Jabon	To provide an attractive display for users who are utilizing the system. This feature is required for the first release as it will provide the means of communicating changes in game state and performance statistics to the user.	

Feature Number 2	**Feature** Interactive Graphical Interface	
Status Approved	**Priority** Medium	**Effort** Medium
Risk Low	**Stability** High	**Target Release** Second release
Assigned To Benjamin Waters Sarah Jabon	**Rationale** To provide an attractive and interactive interface for users who are utilizing the system. The priority is medium because it is desired for the user to be able to interact with the final system. The user interface features are designated to Benjamin Waters and Sarah Jabon due to their development experience of front-end systems.	

Feature Number 3	**Feature** Computational Performance Display	
Status Approved	**Priority** Critical	**Effort** High
Risk Low	**Stability** High	**Target Release** First release
Assigned To Benjamin Waters Sarah Jabon	**Rationale** Constantly provides performance statistics for the analysis of the parallel computing [1], which will reinforce the superiority of parallel computing [1] to the viewer. This feature will be wrapped up in the graphical display of the first release and has a priority of critical because it is necessary to showcase the superiority of parallel computing [1].	

Feature Number	**Feature**	
4	Parallel Computing [1] Based Player	
Status	**Priority**	**Effort**
Approved	Critical	High
Risk	**Stability**	**Target Release**
Low	High	First release
Assigned To	**Rationale**	
Ian Roberts Mark Jenne	Essential for the demonstration of the parallel computing [1] technology in comparison to traditional sequential computing [2]. This feature has a priority of critical because it is a core element of the system. Ian has some experience in working with parallel computing [1] and so will be in charge of the application's usage of and connection to the parallel computing [1] API [11].	

Feature Number	**Feature**	
5	Sequential Computing [2] Based Player	
Status	**Priority**	**Effort**
Approved	Critical	High
Risk	**Stability**	**Target Release**
Low	High	First release
Assigned To	**Rationale**	
Ian Roberts Mark Jenne	Essential for performance comparison between parallel [1] and sequential [2] computing in the application. This feature has a priority of critical because it is a core element of the system. Ian will also be in charge of the sequential computing [2] player to keep it consistent with the parallel computing [1] player.	

Feature Number	**Feature**	
6	Support for Human Player	
Status	**Priority**	**Effort**
Approved	Medium	Medium
Risk	**Stability**	**Target Release**
Low	High	Second release
Assigned To	**Rationale**	
Ian Roberts Mark Jenne	User can interact with the application and play the game with the automated players. The priority is medium because it is not required for the system, but it would be beneficial to the user for demonstration purposes. The target release is the second release; the application will first be developed with only two computer-based players.	

Feature Number	**Feature**	
7	Artificial Intelligence [12] Engine Driving Computer-player Decisions	
Status	**Priority**	**Effort**
Approved	Critical	High
Risk	**Stability**	**Target Release**
High	Medium	First release
Assigned To	**Rationale**	
Mark Jenne Ian Roberts	The automated players need to demonstrate basic analysis and reasoning skills and be able to make informed decisions. The artificial intelligence [12] engine poses a high risk. If the engine is developed and not yielding intended or appropriate results in the decision-making process, the structure may need to be reconsidered with a different reasoning system. Since all of the actions of the automated players rely on this feature of the system, it is of critical importance that it be developed properly despite the level of risk present. The AI [12] engine is also the least stable as it may require alterations or redesign if it is not yielding desirable results. Mark has experience in AI [12] development and so will be in charge of the development of and implementation of the AI [12] engine.	

Feature Number 8	**Feature** Support for Two Automated Players	
Status Approved	**Priority** Critical	**Effort** High
Risk Medium	**Stability** Medium	**Target Release** First release
Assigned To Mark Jenne Ian Roberts	**Rationale** The system, particularly the AI [12] engine should be able to support two automated players, one sequential [2] and one parallel [1]. This feature has a priority of critical because it is a core intended function of the system.	

Feature Number 9	**Feature** Support for More than Two Automated Players	
Status Approved	**Priority** High	**Effort** High
Risk Medium	**Stability** Medium	**Target Release** Second release
Assigned To Mark Jenne Ian Roberts	**Rationale** The system should be able to support more than two automated players. This feature has a priority of high because it is a highly desired feature of the final system.	

Feature Number 10	**Feature** No-Limit Texas Hold 'Em Poker [14]	
Status Approved	**Priority** Critical	**Effort** High
Risk Low	**Stability** High	**Target Release** First release
Assigned To Sarah Jabon Benjamin Waters	**Rationale** Game structure of the application which offers a sandbox environment to demonstrate the parallel computing [1]. All the rules of Texas Hold 'Em poker [14] will be implemented. This feature has a priority of critical because it is necessary to satisfy core requirements of the system as specified by the client.	

Feature Number	**Feature**	
11	Ability to Select Type of Player Types	
Status	**Priority**	**Effort**
Approved	High	Medium
Risk	**Stability**	**Target Release**
Low	High	Second release
Assigned To	**Rationale**	
Sarah Jabon Benjamin Waters	Gives the user the ability to designate automated players using sequential [2] or parallel computing [1]. The priority is high because it is not required for the system, but it is important for the user to interact with the application.	

Feature Number	**Feature**	
12	Ongoing Summary of API [11] Feedback	
Status	**Priority**	**Effort**
Approved	Critical	High
Risk	**Stability**	**Target Release**
Low	High	First release
Assigned To	**Rationale**	
All team members	Provides the CDP team with feedback that they can use to improve the API [11]. The priority of this feature is critical because it is one of our client's core requirements.	

2 Supplementary Specifications

The following is a comprehensive list of all system requirements. They are compiled from the features listed prior and should be sufficient to capture all requirements to which the system must constrain.

2.1 Usability Requirements

1. The system must allow the user to start the poker simulation using six clicks or fewer
2. The system should use no language other than English
3. The system will use standard representations of chip value and card type in Texas Hold 'Em [14] for easy recognition
4. The system shall look and feel like standard Poker applications that can be found online, such as at [3] and [10]
5. The system shall be simple enough that it takes a user no longer than two minutes to learn to initiate the game
6. The system should display statistics about the both sequential [2] and parallel [1] players clearly on the screen so the user can view them easily

2.2 Functional Requirements

1. The system shall follow the standard rules of Texas Hold 'Em [14] as specified at [9]
2. The system must display the current game state
3. The system should show animations for bets and card actions
4. The system will allow the user to select the number of hands to simulate
5. The system must display the actions of each computer player as they occur, such as "Player 1 has folded."
6. The system must display the differences in runtime for parallel [1] players and sequential [2] players
7. The system should display the cards graphically
8. The system must allow the user to select the number of players (second release)
9. The system must allow the user to select if each player is parallel [1] or sequential [2] (second release)

2.3 Performance Requirements

1. The system will complete approximately one hand per four seconds
2. The system must record how long it takes a player to complete each move
3. After running for the specified number of hands, the system will display statistics in the form of average runtime of each player.
4. The system must not intentionally make the sequential [2] player inefficient
5. The system must use an artificial intelligence (AI) [12] algorithm that functions well in a parallel [1] environment

2.4 Reliability Requirements

1. The system must be stable for as many rounds are required to win a game
2. The system may experience no more than one error per 100 executions

2.5 Supportability Requirements

1. The system must support at minimum two players
2. The system should be designed so that it can be modified or updated by someone who was not involved in the initial design process
3. The system must not have any aspects that inhibit modifications
4. The developing team is responsible for maintenance until June 2010, when responsibility for maintenance will be transferred to the client

2.6 Hardware and Software Interfaces

1. The system must have a visually appealing graphical user interface
2. The system must use either a Windows [8] form or a Windows [8] presentation foundation (WPF)
3. The system must use Window's CDP API [11]

2.7 Documentation, Installation, Legal, and Licensing Requirements

1. The team members must document their experiences with the CDP API [11] The system's use of .NET [4] code is under a license from Microsoft [6] for Visual Studio [7]
2. The system must have all design considerations and actions documented
3. The system need not include a user manual. However, it is recommended that the system have one

4. The intellectual property of the application shall be the sole property of the Rose-Hulman students developing it
5. The system may be used to its full extent by the client at any time
6. The system will only need a select number of files to run. It will simply be a desktop application
7. The system will no require the user to install any applications

2.8 Design Constraints

1. The system must implement the API's [11] provided by Microsoft's [6] .NET [4]
2. The system must run on a Windows [8] platform

2.9 Training Requirements

1. The system should require no training to use
2. The system should have a "Help" option at the user's disposal

2.10 Deliverables

1. Milestone Documents One to Five
2. Final deliverable, which includes updated versions of all Milestone Documents, client comments, and lessons
3. Individual engineering journals
4. Reports with feedback on the API [11]

3 Data Flow Diagrams

The data flow diagrams in this section describe how different data flows throughout the system. Several of these diagrams have been altered to reflect changes in system structure. This set of data flow diagrams currently represents the final system to be completed by the end of this academic year. All intended functionality is represented here with the exception of the potential for the user to be able to play along with the automated players.

3.1 Context Flow Diagram

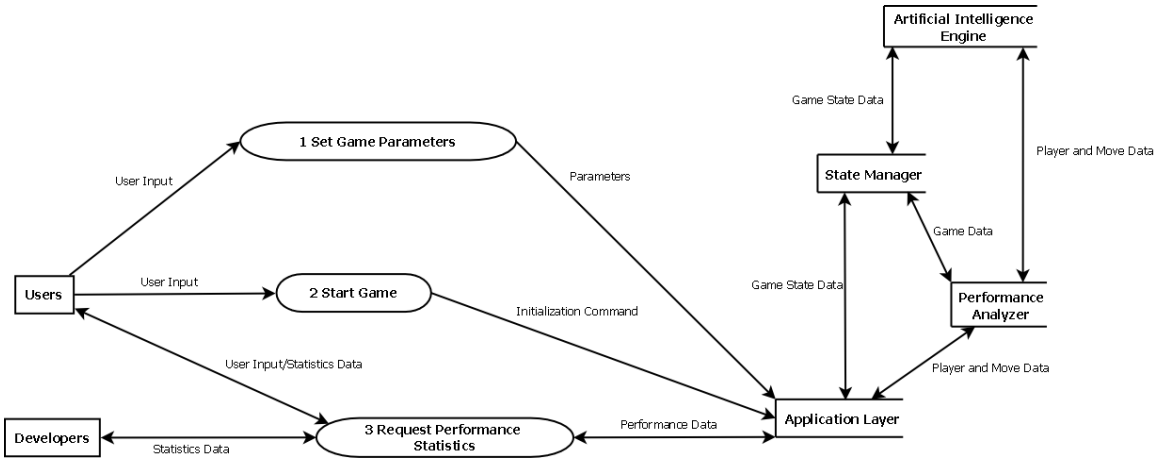

Figure 1: Context Flow Diagram

This diagram is an overview of the flow of data between the users and the back end components governing the system. The user specifies game and player parameters and then initializes the system. The application plays through the desired number of hands and provides feedback in the form of performance statistics during gameplay. The user can also request performance feedback for the game as a whole once it has completed.

3.2 Level 0: 1. Set Game Parameters

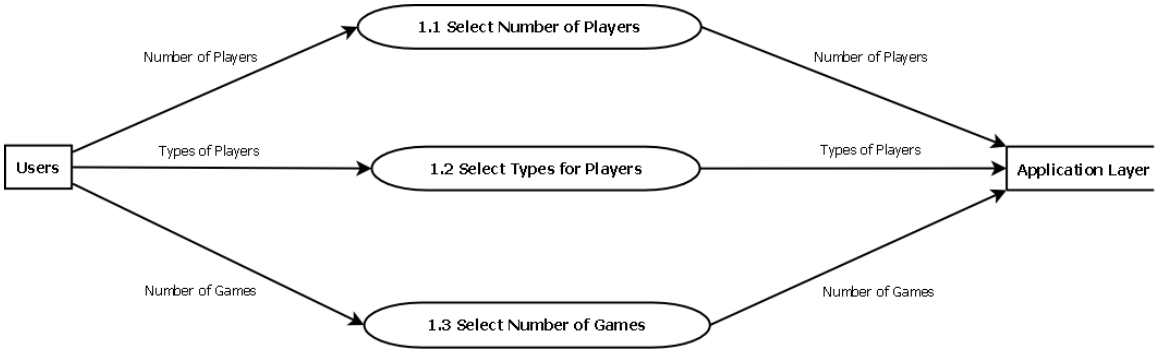

Figure 2: 1. Set Game Parameters

These three processes handle user input parameters for the creation of the game state. The user will be able to specify the number of players to play in the games, the type of each player, and the number of hands to be played. This data is passed to the application layer for game initialization.

3.3 Level 1: 1.1 Select Number of Players

The user selects the number of players. The user must indicate at least two players to play the game. This sub-process is not expanded as all it requires is the user to specify the number of players either by manually typing an integer or selecting one of them provided in the form of buttons.

3.4 Level 1: 1.2 Select Types for Players

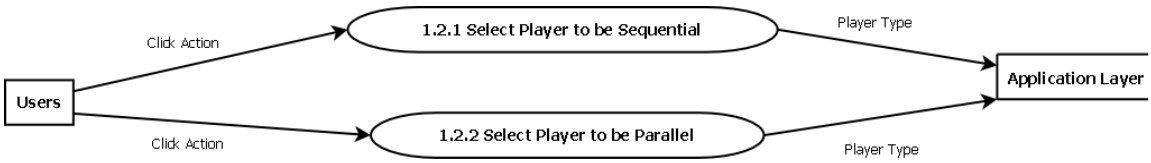

Figure 3: 1.2 Select Type of Players

The user selects the type for each automated player. The user has the choice of making each player either parallel [1] or sequential [2]. Performance statistics will be maintained for all automated players.

3.5 Level 1: 1.3 Select Number of Games

The user selects the number of hands to be played. This sub-process is not expanded because it only requires the user to input an integer value representing the number of games to be played by the automated players.

3.6 Level 0: 2. Start Game

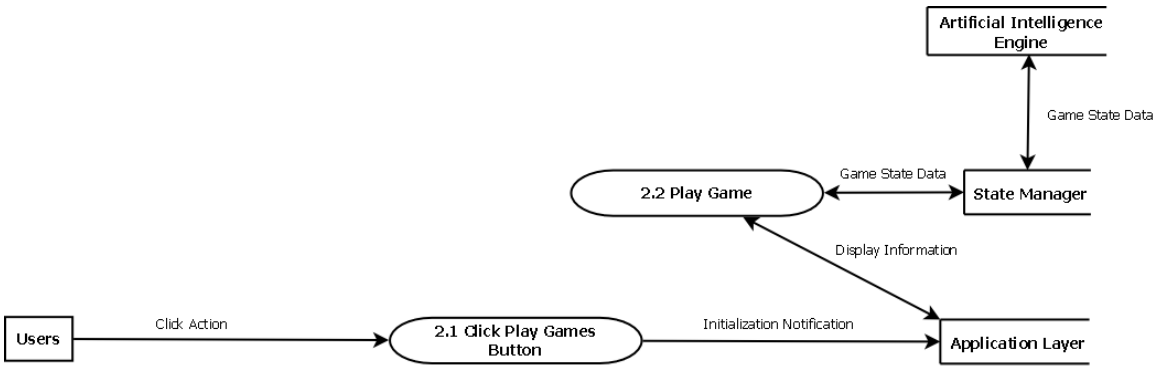

Figure 4: 2. Start Game

This diagram provides a basic overview of the user starting the application once the game parameters have been set as well as of what the system does once the user has selected start. The user is not actively involved in the playing of the game.

3.7 Level 1: 2.1 Click Play Games Button

The user will simply click a button that will cause the application to play through the designated number of poker hands with the given player parameters. This sub-process is not expanded because it only involves the user clicking a button to start the game.

3.8 Level 1: 2.2 Play Game

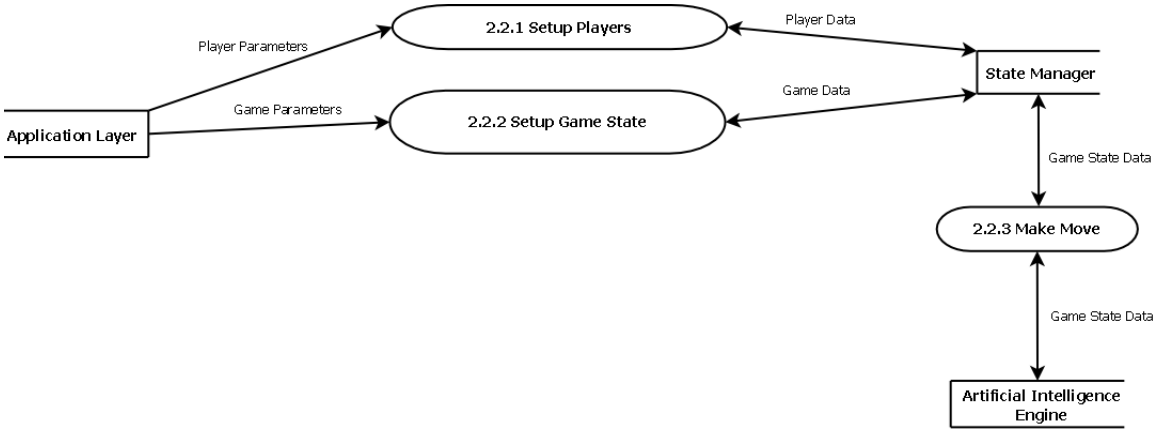

Figure 5: 2.2 Play Game

This data flow diagram reveals how the application layer, state manager, and artificial intelligence [12] engine work together within the system to initialize the game state and play through the hands.

3.9 Level 1: 2.2.1 Set Up Players

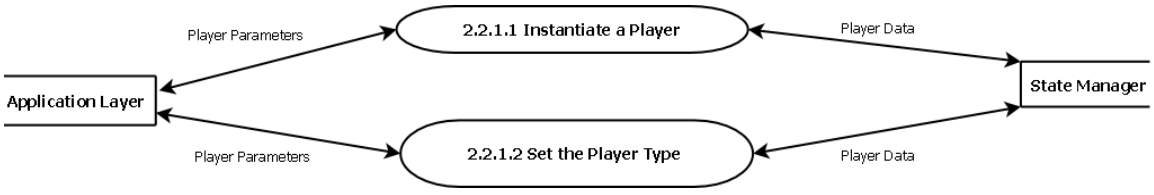

Figure 6: 2.2.1 Set Up Players

The application layer takes the player parameters input by the user and uses them to instantiate the desired number of players and give each player the user-selected type. The application layer communicates this data with the state manager component of the system.

3.10 Level 1: 2.2.1.1 Instantiate a Player

The application layer will create player instances to meet the desired number of players as specified by the user. This sub-process is not expanded because the instantiation is code functionality that will be handled through object management within the software.

3.11 Level 1: 2.2.1.1 Set the Player Type

The application layer will set each player to be sequential [2] or parallel [1] based on the types the user has selected for the players. This sub-process is not expanded because the software component governing player creation will handle setting the player type.

3.12 Level 2: 2.2.2 Set Up Game State

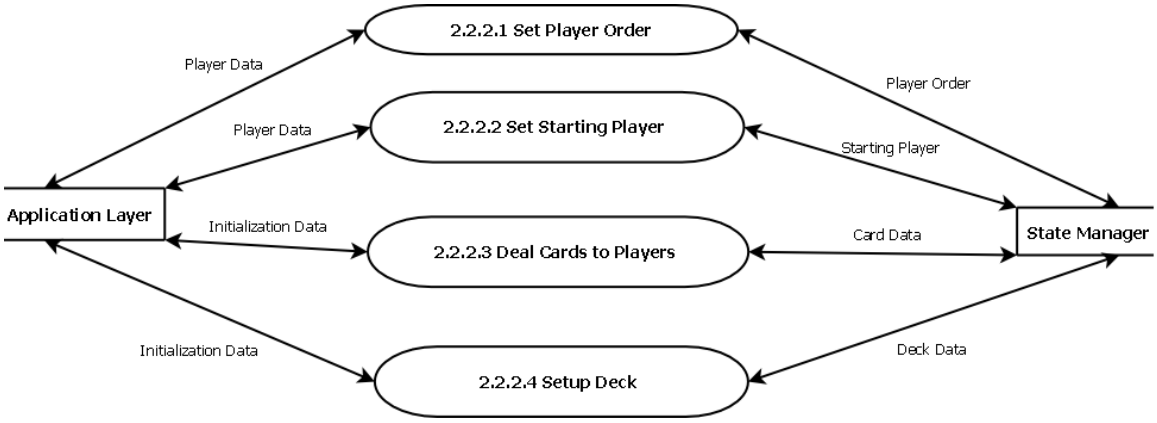

Figure 7: 2.2.2 Set Up Game State

The application layer communicates user input as well as derived parameters to the state manager in order to initialize the overall state of the game. Each of these sub-processes represents simply getting and setting data and do not need to be expanded.

3.13 Level 2: 2.2.3 Make Move

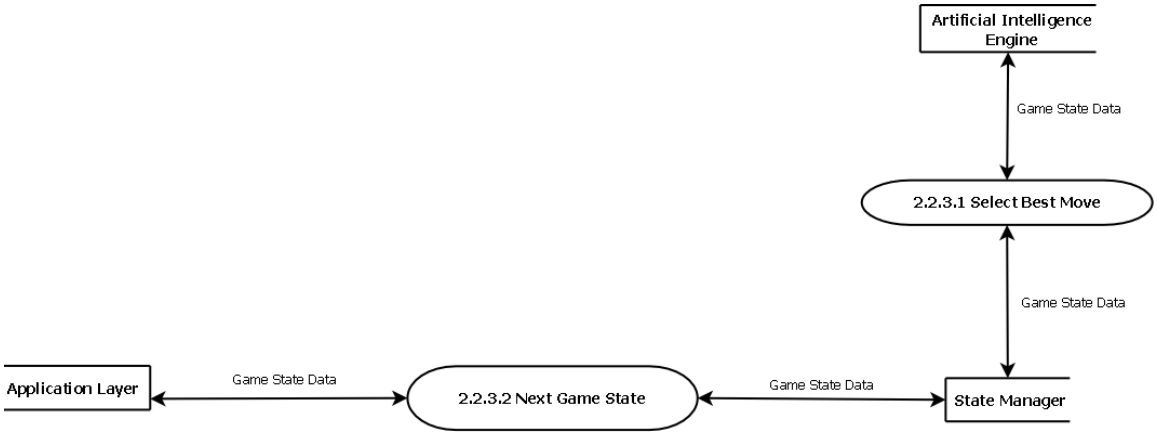

Figure 8: 2.2.3 Make Move

The artificial intelligence [12] engine will govern the reasoning exhibited by and decisions made by each automated player. This component will pull all necessary information from the state manager and then select a move to be executed. The artificial intelligence engine communicates the decision with the state manager.

3.14 Level 2: 2.2.3.1 Select Best Move

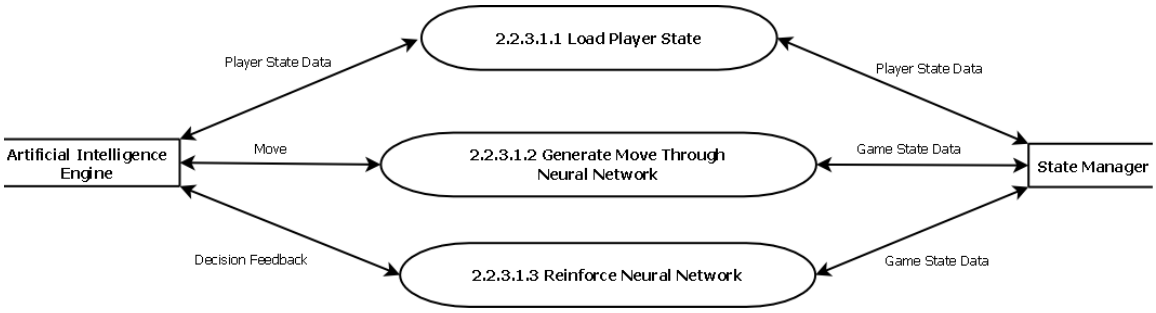

Figure 9: 2.2.3.1 Select Move

The processes shown here form the prominent function of the artificial intelligence [12] engine. Here an artificial neural network [13] generates a move to be made, followed by reinforcement being applied to the connection strengths in the network.

3.15 Level 0: 3. Request Performance Statistics

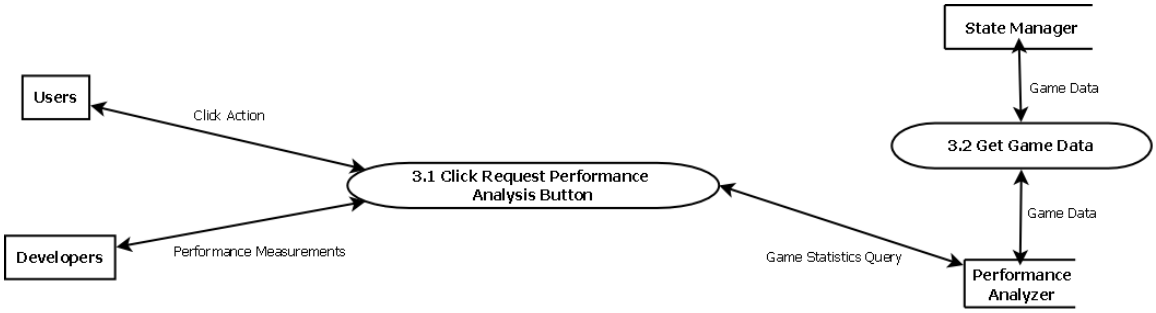

Figure 10: 3. Request Performance Statistics

This process governs the user requesting performance statistics for the game just played. Performance measurements will include data on the number of hands won by each player, the hand each player had and the amount bet, and the average time taken per move for each player.

3.16 Level 1: 3.2 Request Game Data for Performance Statistics

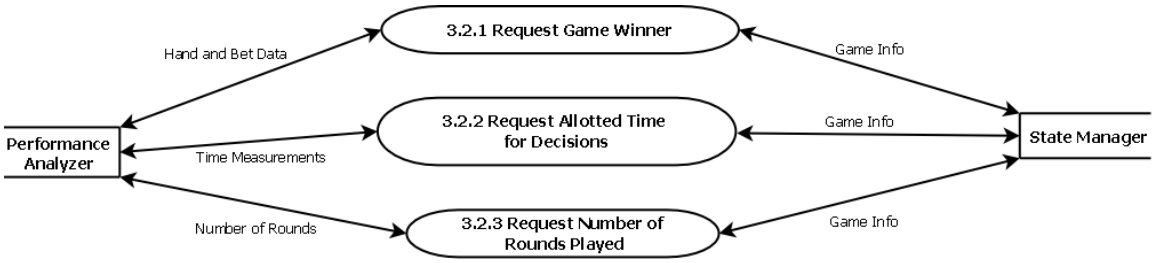

Figure 11: 3.2 Request Game Data for Performance Statistics

The performance analyzer will request state data and allotted time for each decision made in the poker game. This data will be communicated with the application layer in order to provide the user with performance statistics for the game just played.

Part IV
Scenario

A scenario is a story about how a user will interact with the system. This particular scenario describes how a user could use the second release application.

4 Scenario

Mike had a chance to try out the poker application. The application had an empty card table with a large "Start" button in the center with a brief description of how the parallel and sequential players worked in the middle. He clicked on the "Start" button, and a popup appeared. The popup had settings on it, such as the number of players and the number of games. Mike set the number of parallel players to one, the number of sequential players to one, and the number of human players to one. Player 1 was a parallel player, Player 2 was a sequential player, and Player 3 was a human player. On the same popup, he set the number of games to one. Once he was done setting his preferences, he clicked the "OK" button.

When the popup closed, the screen had place set for Players 1 and 2 on the left and right sides of the screen respectively. The deck was on the top of the screen, and Mike's place was at the bottom of the screen. All players, including Mike, had a pile of chips next to their hands. Some of Player 1's chips moved into the middle of the table, and Player 1's total amount of money went from $500 to $490. The running total of the pot, which was displayed in the middle of the screen, went up to $10. Then Player 2 contributed $5 into the pot. As Mike watched, he noticed that the game status bar on the bottom of the screen stated that the big blind was $10 and the small blind was $5. The game status bar then displayed the message: "Players have placed pre-flop bets." Then, cards were dealt to all three players.

Three pairs of cards were displayed on the screen. Players 1 and 2 had face-down cards. However, Mike's cards were face-up. He had a five of clubs and a king of hearts. Next to Mike's hand and chips were four buttons: "Call," "Raise," "Check," and "Fold." He noticed the game status bar changed to "Player 1 called. It took Player 1 8.254 ms to make this move." as some of Player 1's chips went to the pot. Player 2 also called. The game status bar also noted that Player 2 called and how long it took for Player 2 to make that move. It was Mike's turn, and he wanted to call. So, he clicked the "Call" button. Five of his chips went into the pot, and his amount of money went down by $25. Players 1 and 2 checked. The game status bar noted this. Mike checked by clicking "Check."

Then, three cards showed up face up on the top of the screen next to the deck. They were the dealer's cards. The cards were: a three of diamonds, a six of clubs, and a queen of spades. As Player 1's chips went into the pot, the game status bar displayed: "Player 1 raised $20. It took Player 1 7.345 ms to make this move. The current bet it $30." Player 2 raised the bet another $20, as the game status bar noted. Mike wanted to stay in the game, so he called by clicking the "Call" button. It was Player 1's turn, and so Player 1 called. Player 2 checked. Finally, Mike checked by clicking "Check."

A fourth card showed up face up on the top of the screen next to the other three dealer's cards. It was a four of diamonds. Player 1 raised the bet by $50, as specified by the game status bar. Player 2 called. Since Mike did not think he had a chance of winning, he decided to fold. So, he

clicked the "Fold" button. His cards on the screen flipped over, and the buttons to call, check, raise, and fold became grayed out. Player 1 checked, and the status bar updated correspondingly.

The fifth card was then displayed on the top of the screen. It was a ten of clubs. Player 1 raised $50. Player 2 called. Then, Players 1 checked. Mike watched as the game status bar updated and the chips went into the middle of the table.

Since the game was over, it was time to show cards. Both Player 1's cards and Player 2's cards flipped over to show the face value. Player 1 had a five of diamonds and a queen of clubs. Player 2 had a three of spades and a two of spades. All the chips in the pot moved to Player 1's side, and Player 1's money total updated. Also, the game status changed to "Player 1 wins." Mike watched as all of this happened.

Then, the statistics information at the bottom of the screen updated. These statistics included measures of efficiency such as the average time to make each move and the ratio of the parallel player's speed to the sequential player's speed. Mike viewed the statistics on the parallel and sequential players. In this case, the parallel player was more efficient. Mike enjoyed working with the poker application, and he decided to play again.

Part V
Screen Shots

5 Screen Shots

The images below represent the screens that the user would encounter when interacting with the application.

5.1 Player Preferences

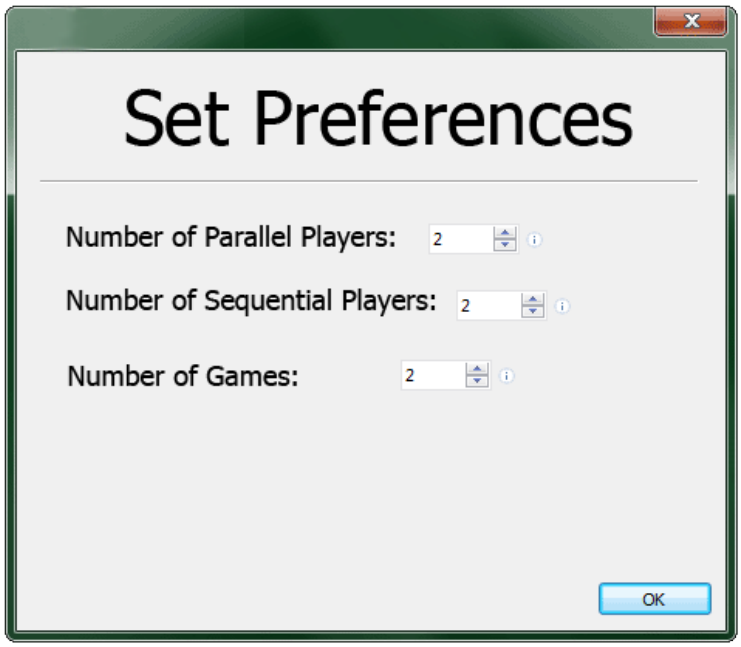

Figure 12: Player Preference Selection

In this screen, the user sets numerous preferences. The player selects how many players they would like in the poker game. Once the user selects the number of players, options for the number of players show up on the preference screen. In this example, the user has selected two parallel players and two sequential players. Finally, the user can select the number of games the players will play. Clicking on "OK" will signify to the system that the user has completed preference selection. Throughout the screen, there are numerous circles with an "i" in them. Clicking on the button provides users with contextual help on what the setting and how to use it.

5.2 In-Play Game Screen

Figure 13: In-Play Game Screen

This screen represents what the game screen would be during play. There would be two cards on the left and two cards on the right for each player. Also, each player would have chips with a corresponding amount of money. The dealer's cards are on the top of the screen. There is a status bar that updates the user with the game state and any relevant statistics from the last move. The game status bar would usually display messages such as "Player 1 raised $50. It took Player 1 6.342 ms to make this move."

5.3 Final Game Screen

Figure 14: Final Game Screen

In this screen, the user can see the final results of the game. The screen shows which player won the game. In addition, there are numerous measurements of efficiency displayed on the bottom of the screen. This way, the players' efficiency can be easily compared by the user.

Part VI
Decision Tree

6 Decision Tree

The decision tree below displays how the system can determine the winner after any round in the game. The process is simplified to two players, but the same decision rules can be applied to more players.

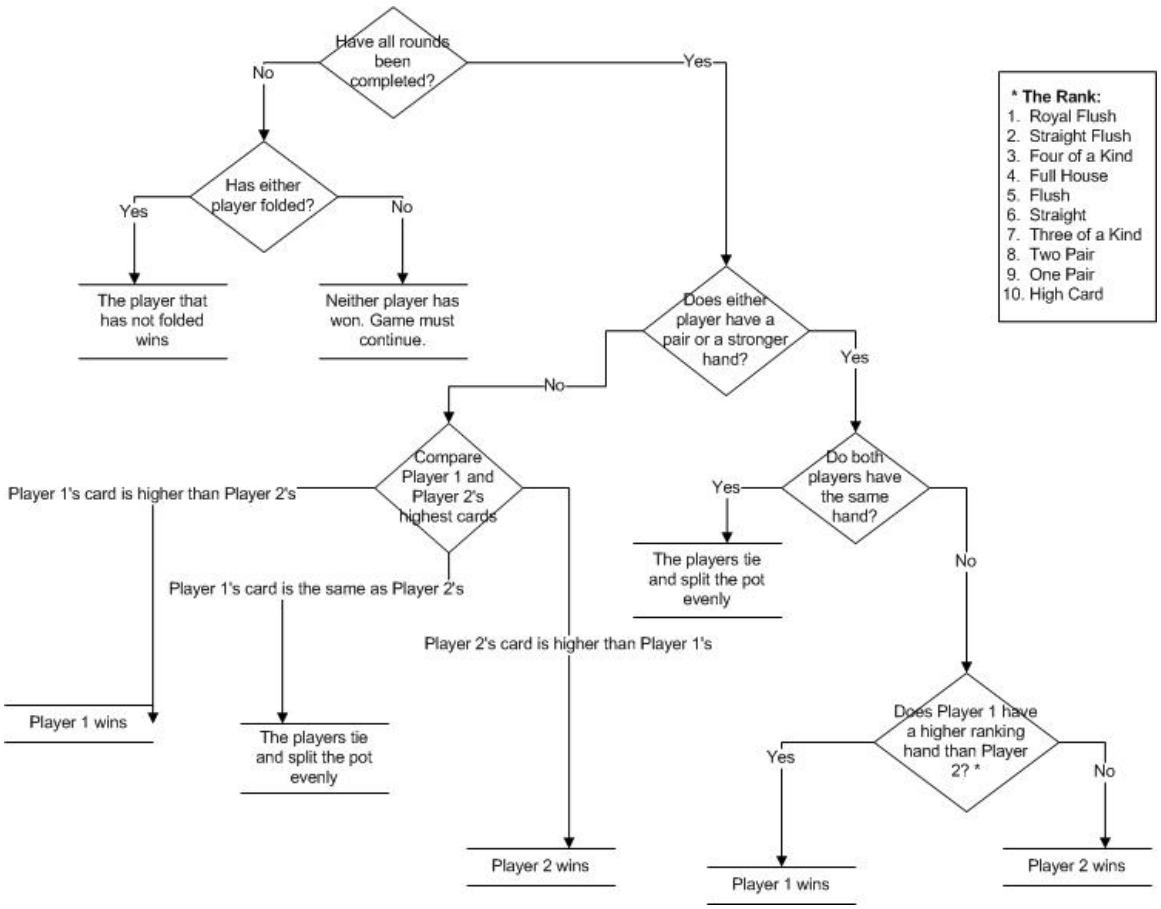

Figure 15: Winner Decision Tree

If all the simulated players have not completed all rounds, the game will have to continue until one player folds. If all rounds have been completed, then the system will first analyze each player's hand to see if it is a pair or stronger hand. If neither player's hand satisfies this constraint, the player with the highest card wins. If the player's highest cards are equal, the players tie and split the pot. If either player has a hand with a pair or a stronger hand, the hands are assessed based on the rankings specified in the image. The player with the highest ranking hand wins. If the players have hands with equal rankings, they tie and split the pot.

Part VII
Appendix

7 System Features Scale Key

Status - The status of a feature pertains to the progress or current state of the feature during the planning and design stages of the system specified by the project.

Priority - Priority here refers to the importance of the development of a feature based on its significance to the overall system. Of the features listed, their priorities range from medium to critical levels. The critical features are labeled as such because they are necessary for the proper functionality intended of the application as specified by the client.

Effort - The effort attribute of a feature describes the amount of time and resources that need to be designated to the development of that feature based on the level of priority and overall importance to the structure of the system. The critical features compose the base functionality of the entire application and represent the goal of the project as a whole. All of the high priority features are integral to the proper functioning of the application as well, but are not quite of dire importance.

Risk - This attribute represents the likelihood that a feature will cause unintended results, such as cost overruns or delays. A low risk represents a lack of potential for undesirable events.

Stability - The stability of a feature reflects the probability that the feature will change or the team's understanding of the feature will change throughout the development of the feature. The stability of a feature relates to the risk that feature presents. If a feature causes unintended results, it will very likely be unstable and need to be redesigned.

Target Release - This records the intended version of the product or system in which the feature will first appear and usually reflects the priority of and effort delegated to that particular feature.

Assigned To - Refers to who within the development team will be responsible for the implementation of the feature. Given that we have four members on this project team, each with capabilities maybe not seen in the others, the features must be distributed for development to the most capable person or persons within the team.

Rationale - The reason or reasons for intended implementation of a feature can be required by the system, mandated by the client, or just a feature decided upon to better the user experience in using the system.

8 Glossary

- C# - an object-oriented language that can be used to create a wide variety of applications, services, and tools [5]
- Microsoft Corporation - A multinational corporation that has five business segments: client, server and tools, online services business, Microsoft business division, and entertainment and devices division [6]
- .NET Framework - A platform build by Microsoft that provides a common set of APIs and a consistent programming model [4]
- Parallel Computing - using multiple resources simultaneously to solve a computation problem [1]
- Sequential Computing - solving a computational problem with sequential processing [2]
- Texas Hold 'Em Poker - A popular card game with multiple players. Players can use any combination of five community cards in combination with their own cards to create the best hand according to numerous rules. [14]
- Visual Studio - an integrated development environment produced by Microsoft [7]

References

[1] Blaise Barney. What is Parallel Computing?, 2009. https://computing.llnl.gov/tutorials/parallel_comp/#Whatis.

[2] Barney Blaise. Overview: Sequential Programming, 1995. http://www.hku.hk/cc/sp2/workshop/html/parallel-intro/ParallelIntro.html#sequential%20programming.

[3] Pogo Games. No Limit Texas Hold'Em, 2009. http://casino-games.pogo.com/games/no-limit-texas-holdem-poker.

[4] Microsoft. .NET Framework Overview, 2008. http://www.microsoft.com/net/Overview.aspx.

[5] Microsoft. Getting Started with Visual C#, 2009. http://msdn.microsoft.com/en-us/vcsharp/dd919145.aspx.

[6] Microsoft. Microsoft, 2009. http://www.microsoft.com/en/us/default.aspx.

[7] Microsoft. Microsoft Visual Studio, 2009. http://www.microsoft.com/visualstudio/en-us/default.mspx.

[8] Microsoft. Windows, 2009. http://www.microsoft.com/WINDOWS/.

[9] PokerListings.com. Official Texas Holdem Rules and Game Play, 2009. http://www.pokerlistings.com/poker-rules-texas-holdem.

[10] PokerStars.com. The World's Largest Poker Site, 2009. http://www.pokerstars.com/.

[11] Wikipedia. Application Programming Iinterface, 2009. http://en.wikipedia.org/wiki/Application_programming_interface.

[12] Wikipedia. Artificial Intelligence, 2009. http://en.wikipedia.org/wiki/Artificial_intelligence.

[13] Wikipedia. Artificial Neural Networks, 2009. http://en.wikipedia.org/wiki/Artificial_neural_network.

[14] Wikipedia. Texas Hold 'Em, 2009. http://en.wikipedia.org/wiki/Texas_hold_%27em.

Index