

Checkers Computer Software Application  
System Design Description

Version 2.0

November 2, 2022

The Systems Squad

Team Members:

Kylie Hall  
Brittany Brenneman  
Xan Weatherholtz

## Table of Contents

<b>1. Introduction</b>	<b>3</b>
1.1 Purpose	3
1.2 Scope	3
1.3 Definitions, Acronyms, and Abbreviations	3
1.4 Document References	4
1.5 System Overview	4
<b>2. Data Design</b>	<b>4</b>
2.1 Internal Data Structures	4
2.2 Data Flow Diagrams	7
2.3 Data Dictionary	8
<b>3. Architecture Design</b>	<b>8</b>
3.1 Program Structure	8
<b>4. Interface Design</b>	<b>9</b>
4.1 Internal System Interfaces	9
4.2 External System Interfaces	9
4.3 User Interfaces	10
<b>5. Procedural Design</b>	<b>11</b>
5.1 Logic Design	11
<b>6. Miscellaneous</b>	<b>13</b>

## Revision History

Date	Version	Description	Author(s)
October 12, 2022	1.0	Initial Version	Kylie Hall, Brittany Brenneman, Xan Weatherholtz, & Isabella Woel-Popovich
November 2, 2022	2.0	Revision of Initial Version	Kylie Hall, Brittany Brenneman, and Xan Weatherholtz

# 1. Introduction

## 1.1 Purpose

The Software Design Description (SDD) is an extensive design of the software to allow the software team to understand what is to be built as well as what are the expectations of the software design. The SDD will describe a design model including data design, architectural design, interface design, and procedural design. The design concepts of the SDD will be communicated to the customers, users, and the software engineering team to ensure the software is being built as required.

The Data Design describes the overall structure of the software's data structures and data objects that will be used. The relationship between data objects will dictate the choice of data structures. It will utilize a level 0 and level 1 data flow diagram to organize the input, processing, and output among the modules.

The Architecture Design Documents all modules that will be used in the software design to create an understanding of the hierarchical format of the program structure. The transformation mapping method will show distinct boundaries between input and output data.

The Interface Design describes all internal, external, and human interfaces in the software. The internal and external interfaces will be based on the information given in the analysis model.

The Procedural Design utilizes graphical, tabular, and textual notations to describe the structure of all programming concepts. This will be a blueprint for all software engineering work on the program where pseudocode will be the medium used to describe the full design of the software.

## 1.2 Scope

The Software Design Document will specifically describe the data design (Section 2) including all major data structures, where each data structure will be specifically defined and all functions described, along with external file structures. It will also describe the architectural design (Section 3) of the software including the hierarchy of the modules present in the system. The interface design (Section 4) will describe the internal and external interface design including the human interface. Finally, the procedural design (Section 5) will describe the structured programming concepts of the program which will form the basis of all subsequent work on the program during construction.

## 1.3 Definitions, Acronyms, and Abbreviations

Acronym	Meaning
DLL	Dynamic Link Library
SDD	Software Design Description

## 1.4 Document References

Document Title	Version	Date	Author(s)

There are no document references within this document at this time.

## 1.5 System Overview

The software system being developed is a standard or classic checkers game application that requires two users to play from one computer and will require the user to use a mouse to select each move. It will automatically display an 8x8 frame size with 24 disk-shaped autogenerated playing pieces placed on the user's designated starting side. The system will include two sets of 12 playing pieces, one set being red and the other set is black for each user to be able to monitor which pieces belong to them. Additionally, both users must input their names which will determine which set they will be assigned, while also serving as a way to keep the score of any pieces conquered throughout the duration of each game. The first user to input their name will be assigned black and the second user will be assigned red. Furthermore, the user assigned to black will go first and start the game. The game will end once a player has collected all of the other player's pieces, or if a player decides to quit the game the winner is decided based on which player has the most pieces left or collected. The system will offer a one undo per move made for each player and a reset option to clear the score and names to begin a new match. The application will be delivered in a hard copy format through a flash drive where no installation is required and must be compatible with Windows DLLs.

## 2. Data Design

The data design will describe all data objects and structures that will be used in the software, including a data flow diagram to display how the data are related to each other, how they will be stored, and the function of the data structures. All major data structures and external file structures will be addressed.

### 2.1 Internal Data Structures

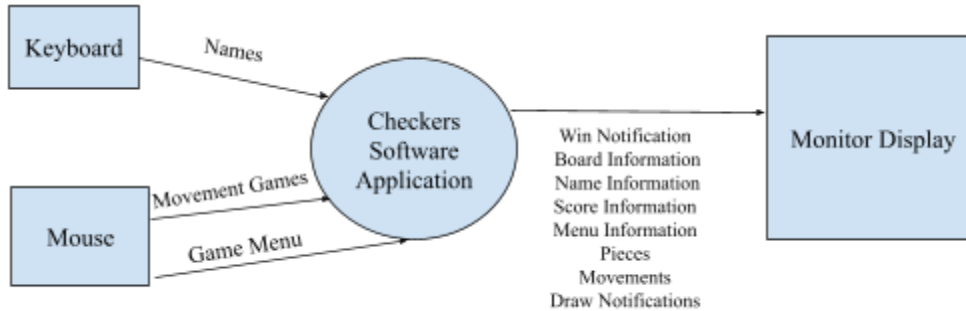
Data Item	Data Type	Description	Example	Size	Validation
nameBlack	String	Stores first user name playing black	John Doe	20 characters	Length > 0
nameRed	String	Stores second user name playing red	Jane Doe	20 characters	Length > 0
board	Array (Integer)	2D array including layout of 8x8 board, -1	{0,2,0,2,0,2,0,2,-1,0,-1,0,-1,0,-1,0,0,-1,0,-1,0,-1,0,-1}	8x8 grid, 8 rows, with length of 8	-

		for red pieces, 1 for black pieces, -2 for king red pieces, 2 for king black pieces and 0 for unoccupied	0,0, 0, 0, 0, 0,0,0, 0,0, 0, 0, 0, 0, 0, 1,0,1,0,1,0,1,0, 0,1,0,1,0,1,0,1, -2,0,-2,0,-2,0,-2,0}		
blackTurn	Boolean	Keeps track if it is black's turn to move	true	1	If true, highlight nameBlack on board
redTurn	Boolean	Keeps track if it is red's turn to move	false	1	If true, highlight nameRed on board
scoreRed	Integer	Stores number of games the red player has won	3	-	-
scoreBlack	Integer	Stores number of games the black player has won	0	1024	-
capturedRed	Integer	Keeps track of red player's score/ captured pieces	2	12	If 12, nameRed has won
capturedBlack	Integer	Keeps track of black player's score/ captured pieces	1	12	If 12, nameBlack has won
followMouse	Boolean	Keeps track if the game piece is following mouse or not	false	1	-
UndoLast	Boolean	Allow one undo per turn taken	false		
name1Txt	Object	Text object displays first player name	-	-	-
name2Txt	Object	Text object displays second	-	-	-

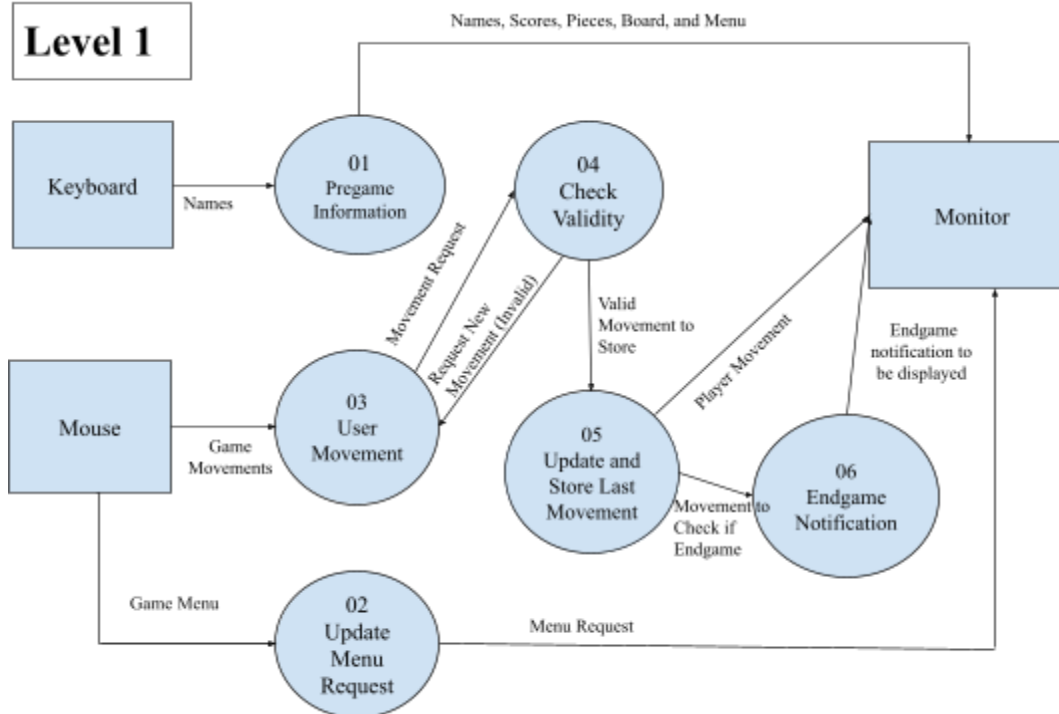
		player name			
scoreRedTxt	Object	Text object that displays second player score	-	-	-
scoreBlackTxt	Object	Text object that displays first player score	-	-	-
messageTxt	Object	Text object to display messages to user	-	-	-
redPrefab	Object	Object of red pieces	-	-	-
blackPrefab	Object	Object of black pieces	-	-	-
enterText	Object	Text object to allow text input from the user	-	-	-
button	Object	Button object allowing the user to change screens where text on the button will change with function needed	-	-	-

## 2.2 Data Flow Diagrams

### Level 0



### Level 1



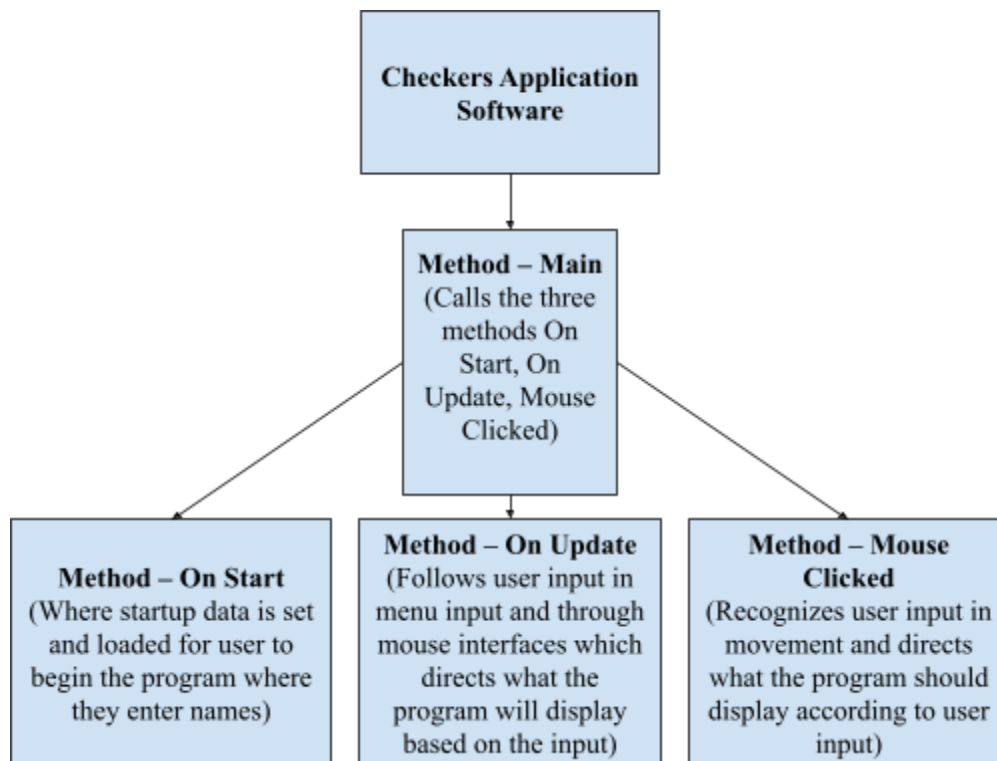
## 2.3 Data Dictionary

The Checkers Computer Software Application does not use a database. Therefore, this section is not applicable. Please refer to section 2.1 *Internal Data Structures* for information on data structures.

## 3. Architecture Design

The architecture design of the software will illustrate how all of the modules in the interface are related to each other in a hierarchical structure. Overall, this is the entire design of the software system with a focus on good architectural design using coupling (no two modules depend completely on each other), abstraction (seeing modules in full, not detailed), hierarchy (logical modules stem from others), and partitioning (logically grouping modules together).

### 3.1 Program Structure





## 4. Interface Design

The interface design describes internal and external program interfaces as well as the design of the human interface. Internal and external interface designs are based on the information obtained from the analysis model.

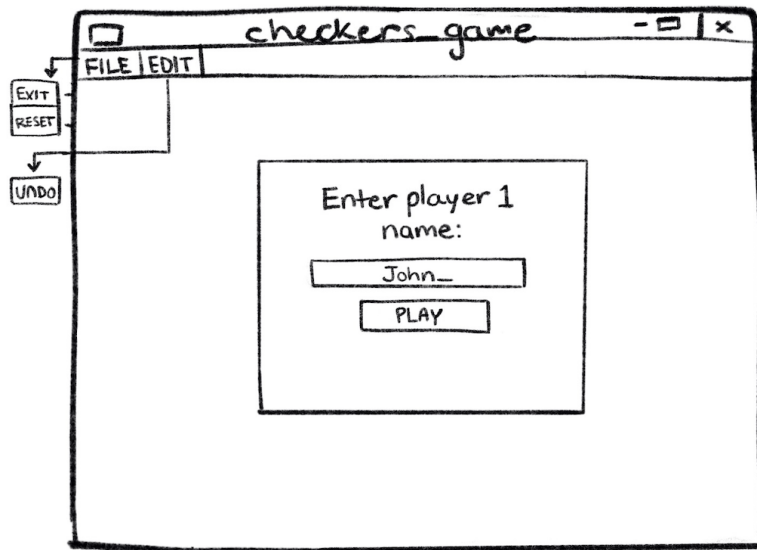
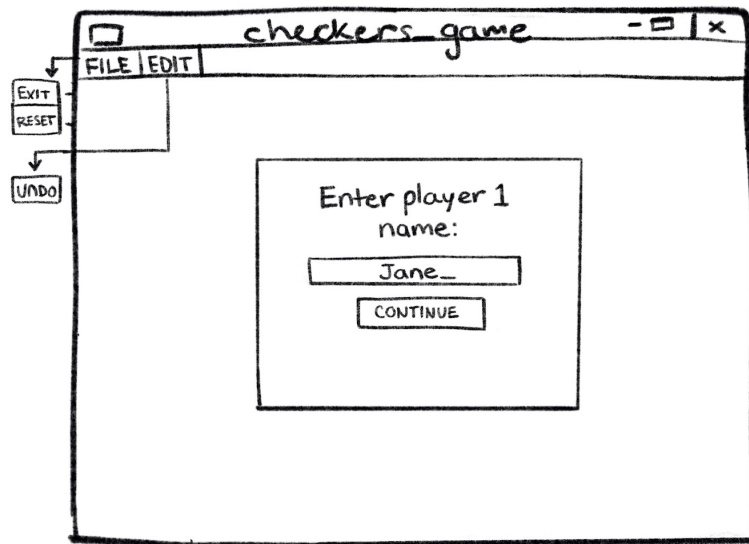
### 4.1 Internal System Interfaces

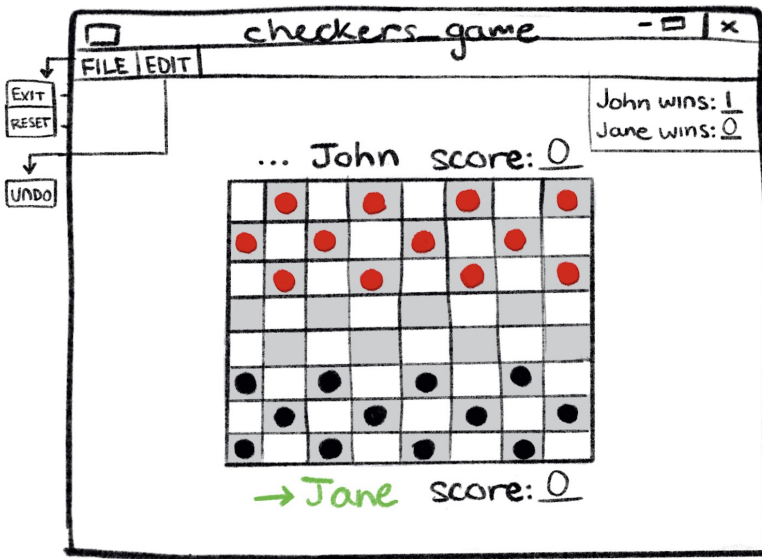
The software will not utilize any internal system interfaces, therefore this section is not applicable.

### 4.2 External System Interfaces

The software will not utilize any external system interfaces, therefore this section is not applicable.

### 4.3 User Interfaces





---

## 5. Procedural Design

The procedural design utilizes graphical, tabular, and textual notations to describe the structure of the software. This design is the blueprint for software engineers which is given in the form of pseudocode to represent the logic of the program.

### 5.1 Logic Design

#### METHOD - Main

- Call On Start method
- Call On Update method
- Call Mouse Clicked method

#### METHOD - On Start

- Set capturedRed to 0
- Set capturedBlack to 0
- Set blackTurn to true
- Set redTurn to false
- Set followMouse to false

- Create 12 redPrefabs and 12 blackPrefabs and place them in correct starting positions on the board (this will be a for loop within a for loop, also known as a nested for loop)
- Display message “Black Player - please enter your name”

#### **METHOD - On Update**

- If followMouse is true
  - Set the piece that has been clicked to the same position as the mouse
- If restart button clicked
  - If nameRed or nameBlack won last round
    - Reset game board where the loser is now playing the black pieces, set nameRed to nameBlack and nameBlack to nameRed.
  - Else if the last round was a draw or no winner was declared
    - Reset the game board where each player continues playing as the color they currently are
- If undo button clicked
  - Pop the spot that was selected from the undo stack
- If quit button clicked
  - Close application

#### **METHOD - Mouse Clicked**

- If button clicked once
  - Store name entered into text input field in nameBlack
  - Clear text input field
  - Display message “Red Player - please enter your name”
- If button clicked again
  - Store name entered into text input field in nameRed
  - Clear text input field
  - Hide text input field
  - Hide button
- Else if checkers piece clicked
  - If there is not a checkers piece currently being moved (if followMouse is false)
    - If checkers piece clicked correlates with whose turn it is
      - If the checkers piece is a king
        - If checkers piece has adjacent spots available to go OR capture spots available (with king rules)
          - Let the player move the checkers piece to valid square (set followMouse to true)
        - Else display message “you can’t move this piece”
      - Else (if the checkers piece is NOT a king)
        - If checkers piece has adjacent spots available to go OR capture spots available (with non-king rules)

- Let the player move the checkers piece to valid square (set followMouse to true)
  - Else display message “you can’t move this piece”
  - Else display message “it is not your turn”
- Else (if there is a checkers piece currently being moved (if followMouse is true))
  - Display message “there’s already a piece there”
- Else if board clicked
  - If there is a piece currently being moved (if followMouse is true)
    - If the board is clicked in a valid move spot
      - If that move spot is a capture spot
        - Place the piece down (set followMouse to false)
        - Remove the piece from the other team that has been captured
        - If red captured add 1 to captureRed, if black captured add 1 to captureBlack
        - If there is another capture spot available
          - Let the same player move again
      - Else (if that move spot is an adjacent spot)
        - Place the piece down (set followMouse to false)
        - Change whose turn it is
      - Else (if the board is clicked in an invalid move spot)
        - Display message “you can’t put that piece there”
    - Else (if there is not a piece currently being moved (if followMouse is false))
      - Display message “please click on a valid piece to move”
    - Else (if capturedRed or capturedBlack equals 12)
      - Display message “Congratulations! (nameRed or nameBlack) won this round!”
      - Allow the nameRed and nameBlack to switch colors where the nameRed will now play black and nameBlack to play red
      - Increment score
      - Reset capture counters
    - Else (if capturedRed and capturedBlack equals 11)
      - Display message “This round is a draw! Neither player earns a point!”
      - Allow the nameRed and nameBlack players to continue playing as the same colors

---

## 6. Miscellaneous

There are no miscellaneous items at this time.