

---

**Main**

# **Software Requirements Specification**

**for**

## **Force Connector**

**Version 1.0 approved**

**Prepared by**

**Bethany Roberts, Brandon Shelton, Payton Long, Cole Dalfonso, &  
Casey Sanchez**

**Skywalker LLC**

**2/3/2021**

# Table of Contents

<b>Table of Contents</b>	<b>ii</b>
<b>Revision History</b>	<b>ii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 Intended Audience and Reading Suggestions	1
1.3 Product Scope	1
1.4 References	1
<b>2. Overall Description</b>	<b>1</b>
2.1 Product Perspective	1
2.2 Product Functions	2
2.3 User Classes and Characteristics	3
2.4 Operating Environment	4
<b>3. External Interface Requirements</b>	<b>5</b>
3.1 User Interfaces	5
<b>4. System Features</b>	<b>11</b>
4.1 Initializing the Application	11
4.2 Starting the Game	11
4.3 Reading the Rules	11
4.4 Exiting the Game	12
4.5 Music	12
4.6 Animation	12
4.7 Trophy	12
4.8 Difficulty	13
<b>5. Other Nonfunctional Requirements</b>	<b>13</b>
5.1 Performance Requirements	13
5.2 Safety Requirements	13
5.3 Security Requirements	13
5.4 Software Quality Attributes	14
5.5 Business Rules	14
<b>6. Other Requirements</b>	<b>14</b>
<b>Appendix A: Glossary</b>	<b>14</b>

## Revision History

Name	Date	Reason For Changes	Version

# **1. Introduction**

## **1.1 Purpose**

This SRS will define the functional and non-functional requirements for a simple game program of connect four named “Force Connector”. Connect four is a turn-based game in which players drop colored markers in an attempt to have four of their color pieces connected horizontally, vertically or diagonally.

## **1.2 Intended Audience and Reading Suggestion**

This document is intended for the contractor, testing team and developers who will implement the code and application of “Force Connector”.

## **1.3 Product Scope**

“Force Connector” shall provide a user interface for a single player game with a choice between two difficulty levels, and will allow the user to play against an AI computer. The game will start from a clear board and track movements until a winning move is made, at which point the application will display the outcome with a corresponding message, animation, and sound. The program shall also provide a count of consecutive games won in a single session and present the user with a corresponding “Trophy.”

## **1.4 References**

# **2. Overall Description**

## **2.1 Product Perspective**

Force Connector shall follow the basic rules of a traditional connect four game as outlined below.

### **Rules**

- Each game starts with a blank board.
- The player with black game pieces makes the first move. Player-one will be randomly selected.
- The user will alternate turns with the AI, each placing one game piece at a time
- The game is over when one player has aligned 4 of their color pieces either horizontally, vertically, or diagonally.

## 2.2 Product Functions

### Main Menu

Start game

- The program shall present user with a start game option.

Rules

- The program shall present user with a description of the rules

End session

- The program shall present the user with an exit button to end the program

### Game play

Board setup

- The program shall initialize with an empty 7x6 board grid

Playing the game

- The program shall show a game marker for each move
- The program shall preset the user with an exit to menu button at all times during game play
- The system shall present the user with a play again button upon a win or loss.

### Additional Functions

Difficulty level

- The program shall present the user with a difficulty selection after selecting start game and before game play begins

Win/loss display animation with sound

- The program shall play a corresponding sound, message and animation upon win or loss.

Trophy award

- The program shall keep a count of consecutive wins per session and display a corresponding award.

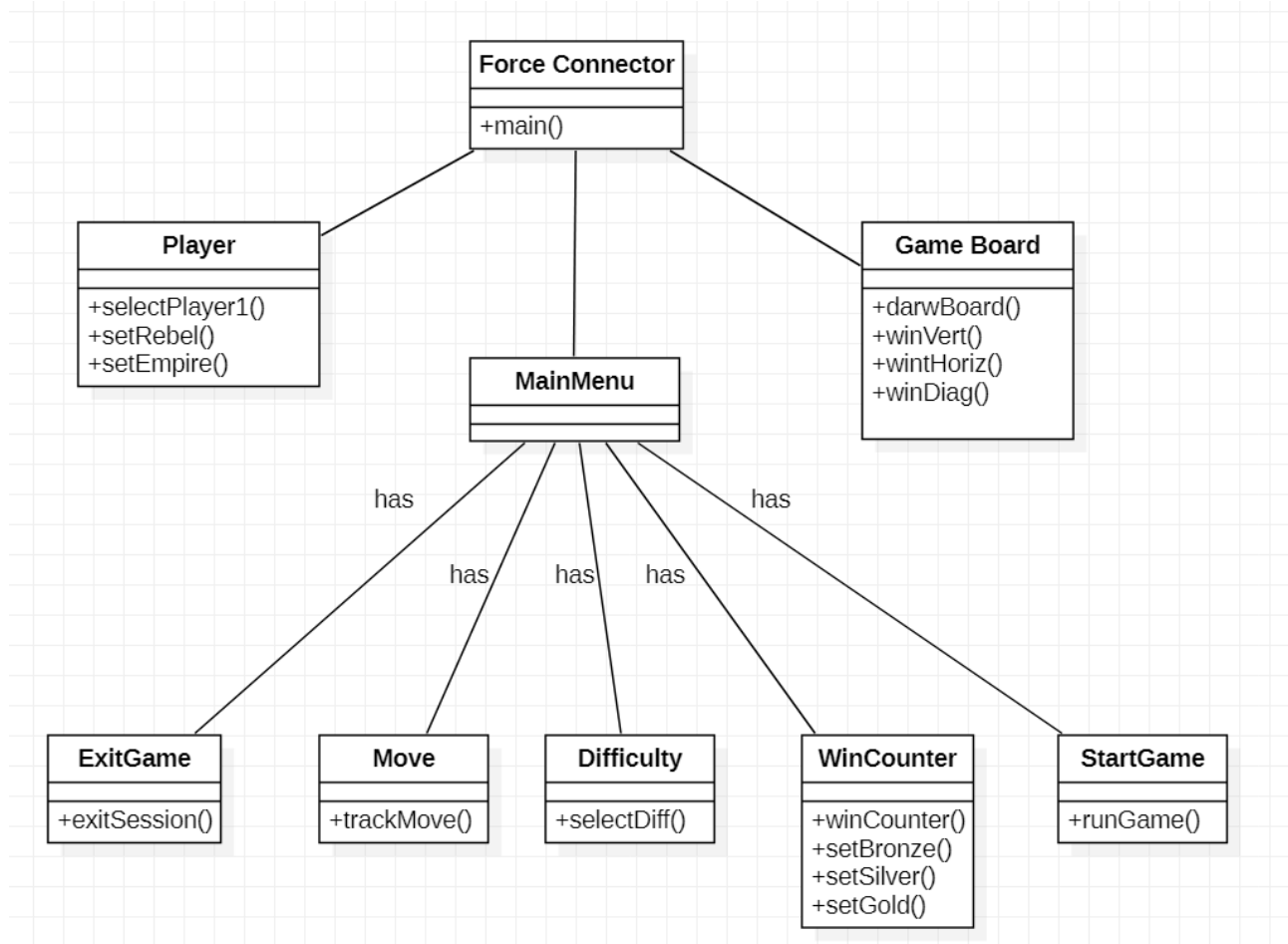


Figure 2.2 Product ER Class Diagram  
*The major features of Force Connector are shown in this ER diagram*

## 2.3 User Classes and Characteristics

A single player will be the only user for this program and should be able to perform the following functions layed out in figure 2.3

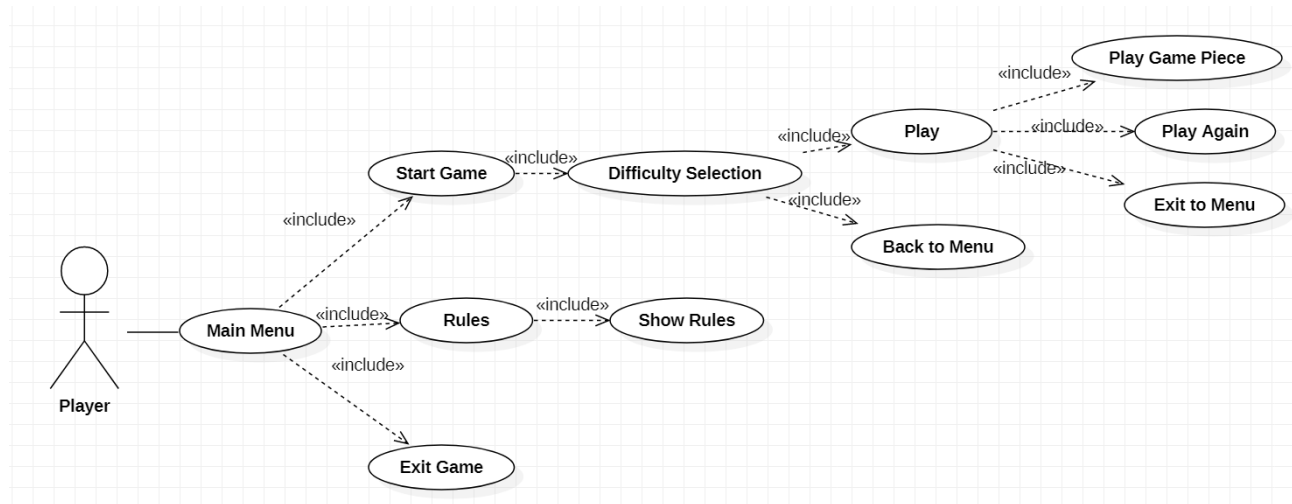


Figure 2.3  
*Force Connector Use Case is demonstrated in the chart above*

## 2.2 Operating Environment

Force Connector shall run in any Development Environment on either Windows or MAC OS.

## **3. External Interface Requirements**

### **3.1 User Interfaces**

-The program shall open with a main menu where players can start a new game, read the rules, or end the program.

-If the user selects 'Rules,' a detailed description of the rules shall be presented, with the options of beginning a game (by selecting the desired difficulty) or going back to the main menu.

-Upon selecting 'Start Game,' the system shall present the user with two difficulty levels ('Easy' or 'Hard').

-After difficulty selection, the program shall present a blank board and the game can begin. Once a game has begun, the user can click the top of any open column to place their piece, after which the AI shall place its piece. The game shall continue until either the user or the AI opponent has won.

-At the conclusion of a game the program shall play a sound and display a dialog box to indicate whether the player has won or lost, and a new game can begin or the player can return to the title screen.

- Main Menu (Figure 3.1.a)
- Rules (Figure 3.1.b)
- Select Difficulty (Figure 3.1.c)
- Game Interface Screen (Figure 3.1.d)
- Win/Loss Dialog (3.1.e)

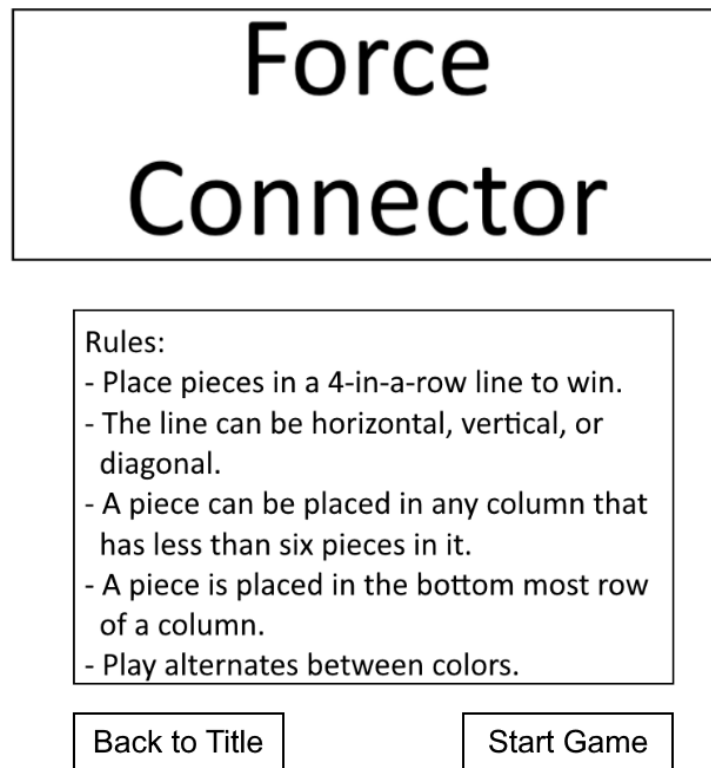


Team Skywalker, LLC

### **Figure 3.1.a (Main Menu)**

The interface that shall be displayed when the application starts. The system shall allow the user to navigate to Figure 3.1.c (Select Difficulty) by clicking "Start Game", to Figure 3.1.b (Rules) by clicking "Rules", and to terminate the system by clicking "exit".





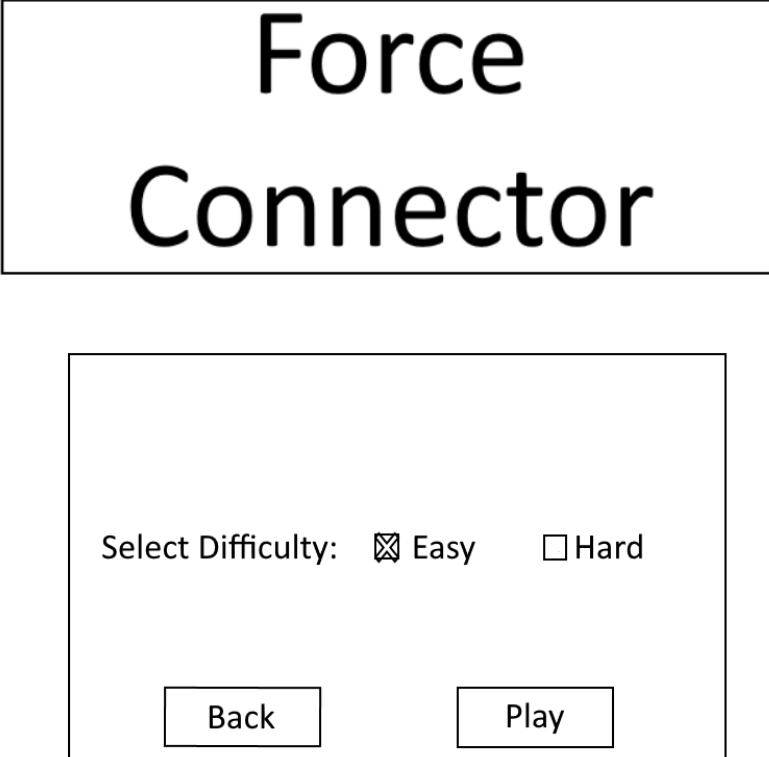
Team Skywalker, LLC

### Figure 3.1.b (Rules)

This interface shall be displayed if the user clicks the “rules” button from Figure 3.1.a (Main Menu).

The “back to title” button shall allow the user to return to Figure 3.1.a (Main Menu) when pressed.

The system shall go to Figure 3.1.c (Select Difficulty) if the user clicks “Start Game”.



The image shows a software interface for 'Force Connector'. At the top, the title 'Force Connector' is displayed in a large, bold, black font within a rectangular box. Below this, there is a section for selecting difficulty. It says 'Select Difficulty:' followed by two options: 'Easy' with a checked checkbox and 'Hard' with an unchecked checkbox. At the bottom of this section, there are two buttons: 'Back' and 'Play', each enclosed in a rectangular box.

# Force Connector

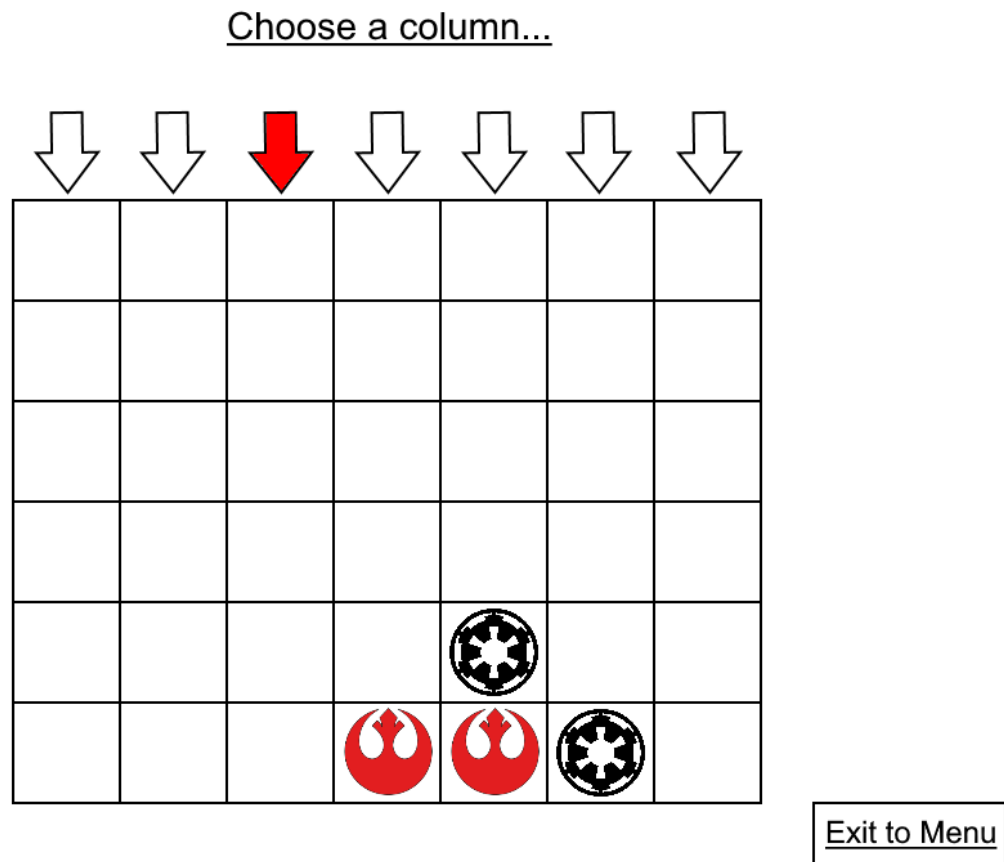
Select Difficulty: ☒ Easy ☐ Hard

Back Play

Team Skywalker, LLC

### Figure 3.1.c (Select Difficulty)

The system shall allow the user to select either “easy” or “hard” difficulty by clicking the box next to either option. By default, the “easy” box is checked. The system shall allow the user to click “play” to move to Figure 3.1.d (Game Interface) using the selected difficulty for the game. the system shall return to Figure 3.1.a (Main Menu) if the user clicks the “back” button.



**Figure 3.1.d (Game Interface)**

The system shall randomly decide to let the user place the first piece or let the AI place the first piece at the start of the game. The system shall place a piece at the bottom-most valid position of a column if the column is selected by the user. The system shall place the AI piece after each piece placed by the user if the game hasn't been won/lost. The system shall return to Figure 3.1.a (Main Menu) if the user clicks the "exit to menu" button.

**Figure 3.1.e (Win/Loss Dialogue)**

The system shall display the win dialogue if the user wins and the loss dialogue if the AI wins. The system shall play a fireworks animation if the user wins. The system shall clear the board and start a new game with the same difficulty if the user clicks the “play again” button. The system shall return to Figure 3.1.a (Main Menu) if the user clicks the “exit to main menu” button.

## **4. System Features**

### **4.1 Initializing the Application**

#### **4.1.1 Functional Requirements**

REQ 4.1.1 Upon initialization of application the user shall be presented with a main menu and a game title.

If the player selects "Start Game" Game continue to 4.2

If the player selects "Rules" continue to 4.3

If the player selects "Exit" continue to 4.5

### **4.2 Starting a Game**

#### **4.2.1 Description**

The game interface will present the player with a grid of 7 columns and 6 rows. When the player selects a column the circle will fall into that column. When there are 4 of the same colored circles in a row, that color wins. The system shall randomly select who goes first - the AI or player. If AI is selected it will immediately put their piece in a column.

#### **4.2.2 Functional Requirements**

REQ 4.2.2a The player shall be able to click any given column to place the circle in.

REQ 4.2.2b The system shall present a text box indicating who has won.

REQ 4.2.2c The system shall present a sound feature that plays music when there is a winner or loser.(continue to 4.5)

REQ 4.2.2d The system shall present an animation function when someone has won or lost.(continue to 4.6)

REQ 4.2.2e The system shall present a button to play a new game.

REQ 4.2.2f The system shall present a radio menu of game level difficulty.(continue to 4.8)

REQ 4.2.2g The system shall present a trophy on the page depending on amounts of wins. (continue to 4.7)

REQ 4.2.2g The system shall present a button to exit the game to the main menu.

### **4.3 Reading the Rules**

#### **4.3.1 Description**

In the rules section the system shall present the player with the rules page which presents the user with text that explains how the game works.

#### 4.3.2 Functional Requirements

REQ 4.3.2a The system shall present an exit button that returns the player back to the main menu.

REQ 4.3.2b The system shall present a button that allows the player to Start the Game.  
(continue from 4.2)

### 4.4 Exiting the Game

#### 4.4.1. Description

The exit button will exit the player from the game GUI.

#### 4.4.2 Functional Requirements

REQ 4.4.2a The player shall be returned to the main menu after pressing the exit button.

### 4.5 Music

#### 4.5.1. Description

Upon winning or losing the game, the system shall present an upbeat or sad noise.

#### 4.5.2 Functional Requirements

REQ 4.5.2a The system shall present a sad sound if the opponent wins the game.

REQ 4.5.2b The system shall present a happy sound if the opponent wins the game.

### 4.6 Animation

#### 4.6.1. Description

Upon winning or losing the game, the system shall present an animation.

#### 4.6.2 Functional Requirements

REQ 4.6.2a The system shall present fireworks if the player wins.

REQ 4.6.2b The system shall present a sad face if the player losses.

### 4.7 Trophy

#### 4.6.1. Description

There will be a bronze trophy for 1 win, silver trophy for 5 wins, and gold trophy for 10 wins.

#### 4.6.2 Functional Requirements

- REQ 4.6.2a    The system shall keep a counter of wins and display the color of the trophy accordingly.

## **4.8 Difficulty**

### **4.8.1. Description**

The system will present a radio button with two difficulty options - easy or hard.

### **4.8.2 Functional Requirements**

- REQ 4.8.2a    The system shall present a button to go back to the main menu.  
REQ 4.8.2b    The system shall present a button to start playing the game.

## **5. Other Nonfunctional Requirements**

### **5.1 Performance Requirements**

- REQ 5.1.1    Throughout development, the team must assure that the use of memory/data must not exceed set parameters in order to have an efficient game/software.
- REQ 5.1.2    The game/software must be able to perform on different operating systems. (See 5.4.2 for more)

### **5.2 Safety Requirements**

- REQ 5.2.1    Safety requirements that this game/software must be followed and ensured throughout the developmental process.
- REQ 5.2.2    The game/software must be able to keep the users data protected and be sure that in case of an error users personal data will not be accidentally manipulated or lost.
- REQ 5.2.3    Incorporate some type of warning to the user of the game/software if some sort of flashing, strobing light or visual effect is used to tell people that are sensitive to those effects are not harmed.

### **5.3 Security Requirements**

- REQ 5.3.1    Development team needs to make sure that the users personal data is not collected and stored, but if needed the team needs to make sure to bring the

user aware (opt in/opt out) of what/which data is being collected and how it will be used.

- REQ 5.3.2 Development team also needs to make sure that the game/software(s) files are secure and are unable to be edited by outside users and be sure that no malicious intent with the game/software can be unable to do so.

## 5.4 Software Quality Attributes

- REQ 5.4.1 Throughout the development process, code must be designed around being efficient, reusable, and maintainable in order to make communication between the development team accurately.
- REQ 5.4.2 The game/software must be able to be performed on multiple devices as well as different systems and should be tested by the development team before finalizing game/software.
- REQ 5.4.3 If the development process requires the use of API's the software must use reliable API's or libraries that are accessible to users and are able to be compatible across multiple devices.

## 5.5 Business Rules

- REQ 5.5.1 The development team will follow ethical business practice and respect values that the users of the software/game will be mindful of.

# 6. Other Requirements

## Appendix A: Glossary

"Player" & "User"	The one using the software/game
Development Team	Direct team creating the software/game
PVP	Player VS. Player
GUI	Graphical User Interface
System	The software/game as a whole