[Date]

[company name]

Game

[Document subtitle]

**Game Description**

Contents

[**Functional:** 1](#_Toc6471123)

[**Non-Functional:** 1](#_Toc6471124)

[**Use Case Diagram** 2](#_Toc6471125)

[**Use Case Description** 2](#_Toc6471126)

[**ID TRD TITLETRD1 Starting InterfaceTRD2 Registration Interface TRD3 Game Engine** 2](#_Toc6471127)

[**TRD2: Registration Interface** 3](#_Toc6471128)

[**TRD3: Game Engine** 3](#_Toc6471129)

[**TRD.1: Application Launched.** 3](#_Toc6471130)

[**TRD.2: User Starts the Game** 4](#_Toc6471131)

[**TRD.3: User Quits the Game** 4](#_Toc6471132)

[TRD2.1: **Choosing Human or CPU**. 5](#_Toc6471133)

[**TRD2.2: Register Players.** 5](#_Toc6471134)

[**TRD3.1: Initiate Game Engine.** 6](#_Toc6471135)

[**TRD3.2: Player Turns** 6](#_Toc6471136)

[**TRD3.3: Calculate scores.** 7](#_Toc6471137)

[**TRD3.4: Check game status** 7](#_Toc6471138)

# In this report we study and analysis the game requirement .We define the game functional and non-functional requirement . We are providing brief description about the game using use case diagram and use case description.

# **Functional:**

1.Must be called Light 15142.

 Must Allow Single or Multiple Players to Play.

a. LAN up to 10 players

b. Must adhere to the rules and regulations of the client/server model found in Light3004 visualization components.

 c)Must have player registration using Unique Names

4.Must not allow users to hold an already used Unique Name.

5.Must stop more than 3 users from joining a single game.

 New users/players to Instances exceeding 3 players in play, Must be allowed/forced to spectate.

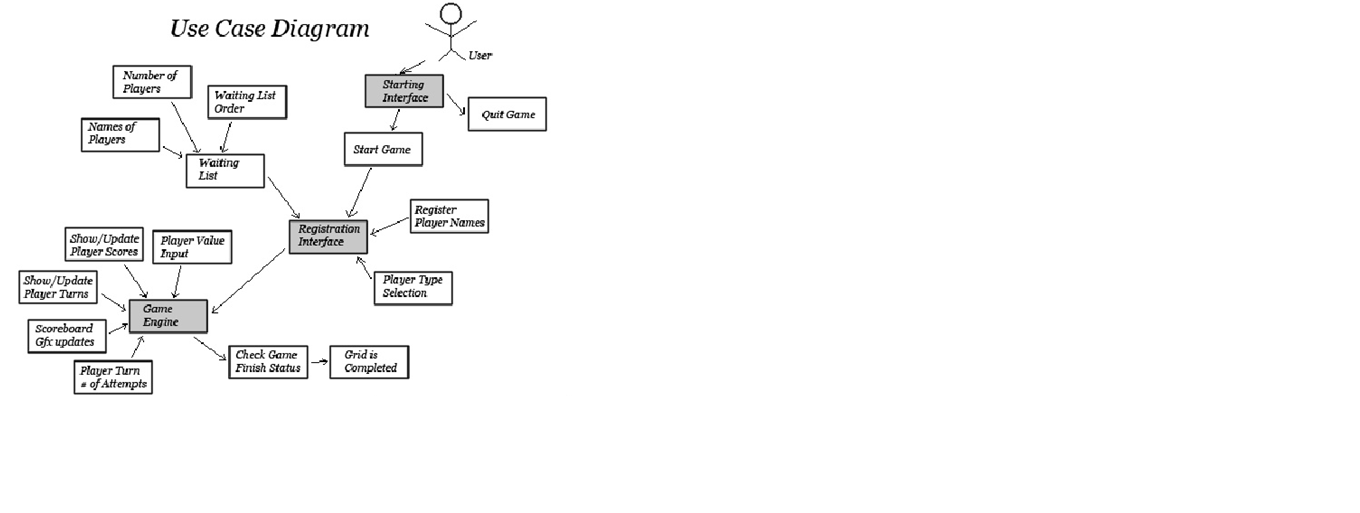
# **Non-Functional:**

1.A Impressive GUI

2.Timer Display

3.Score Display

# **Use Case Diagram**



# **Use Case Description**

## **ID TRD TITLETRD1 Starting InterfaceTRD2 Registration Interface TRD3 Game Engine**

TRD1: Starting Interface

**Requirements**

**Description**: The interface that prompts the user how to start.

**Actors**: User, System

**Pre-condition:** The user must initiate (run) the application.

**Triggering Event:** Automatic from start-up, run from the system. FR1, NFR1

**Sequence:** Application begins, interface loads. Uses secondary conditions 1.1, 1.2, 1.3

**Post Condition**: Game initiates the Registration Interface.

**Resulting Events:** Jump to TRD2 (Registration Interface).

**Alternative Scenarios: User quits the application. N/A**

## **TRD2: Registration Interface**

**Requirements**

**Description**: The interface allowing the user to register multiple players.

**Actors**: User, System

**Pre-condition:** Must have initiated the starting interface (TRD1).

**Triggering Event**: Players are to be registered and named here. FR2, FR3, FR4, FR5, FR6, NFR1

**Sequence**: Players are registered by name, type (CPU or human)and by the total number of players available. Uses secondary conditions 2.1, 2.2

**Post Condition:** Players are registered.

**Resulting Events**: Jump to TRD3 (Game Engine).

**Alternative Scenarios**: N/A

**Non-functional requirements:**-Impressive GUI NFR-1

## **TRD3: Game Engine**

**Requirements**

**Description**: The actual game of Tron that multiple people take turns to play, until the board is clear and there is a winner.

**Actors**: User, System

**Pre-condition**: Must have registered all players (TRD2).

**Triggering Event**: Players take turns clearing the grid.FR2, FR5-12, NFR1-4

## **TRD.1: Application Launched.**

**Requirements**

**Description:** The user launches the program from the PC.

**Actors**: User

**Pre-condition**: N/A

**Triggering Event:** Program launches.

See TRD.

**Sequence**: Program initiates.

**Post Condition**: Application is launched.

**Resulting Events**: Starting interface launches.

**Alternative Scenarios:** N/A

**Non-functional requirements:** See TRD.

See TRD.

## **TRD.2: User Starts the Game**

**Requirements**

**Description**: The player starts the game from the starting interface.

**Actors**: User

**Pre-condition:** Program is launched (TRD 1.1)

**Triggering Event**: Click to start the game.

See TRD.

**Sequence:** Launch the registration interface.

**Post Condition:** Brings up registration interface.

**Resulting Events:** Registration interface launches.

**Alternative Scenarios:** Quit game (TRD 1.3)

**Non-functional requirements:** See TRD.

See TRD.

## **TRD.3: User Quits the Game**

**Requirements**

**Description**: The player quits the game from the starting interface.

**Actors**: User

**Pre-condition:** Program is launched (TRD 1.3)

**Triggering Event**: Click to exit the game.

See TRD.

**Sequence**: Closes the program.

**Post Condition:** Application is closed.

**Resulting Events:** Application is closed.

**Alternative Scenarios:** Start game (TRD 1.2)

**Non-functional requirements:** See TRD.

See TRD.

## TRD2.1: **Choosing Human or CPU**.

**Requirements**

**Description:** The player chooses either human or CPU players.

**Actors**: User

**Pre-condition:** Registration screen is initiated.

**Triggering Event**: Human/CPU players are set.

See TRD2.

**Sequence:** Choose single/multiplayer game.

**Post Condition:** Set game type.

**Resulting Events:** Human/CPU players are set.

**Alternative Scenarios:** Naming, Registering Players.

**Non-functional requirements:** See TRD2.

See TRD2.

## **TRD2.2: Register Players.**

**Requirements**

**Description:** Type player names using the keyboard.

**Actors**: User

**Pre-condition**: Registration screen is initiated.

**Triggering Event:** Players are given unique names.

See TRD2.

**Sequence**: Players are registered.

**Post Condition:** Players must be registered.

**Resulting Events**: The game engine initiates.

**Alternative Scenarios**: Setting player types, number of players.

**Non-functional requirements:** See TRD2.

See TRD2.

## **TRD3.1: Initiate Game Engine.**

**Requirements**

**Description**: All GUI and gameplay elements are loaded into play.

**Actors**: System

**Pre-condition:** All players are properly registered.

**Triggering Event**: User registers players correctly.

See TRD3.

**Sequence**: All elements of the game are loaded into play.

**Post Condition:** Everything in the game board must be loaded.

**Resulting Events**: The game begins play.

**Alternative Scenarios**: N/A

**Non-functional requirements:** See TRD3.

See TRD3.

## **TRD3.2: Player Turns**

**Requirements**

**Description**: Determine turn of next player.

**Actors**: System

**Pre-condition**: Player must have finished a turn (or the game juststarted).

**Triggering Event:** Start game, finish a turn.

See TRD3.

**Sequence:** Game switches player turns.

**Post Condition**: Must have switched to the next player.

**Resulting Events**: The game continues under a different player.

**Alternative Scenarios:** N/A

**Non-functional requirements:** See TRD3.

## **TRD3.3: Calculate scores.**

**Requirements**

**Description**: Update scores of all the players.

**Actors**: System

**Pre-condition**: Player must have finished a turn.

**Triggering Event**: Start game, finish a turn.

**Sequence**: Game updates the player scores.

**Post Condition**: Scores must be properly updated.

**Resulting Events:** The game has updated all scores.

**Alternative Scenarios:** N/A

**Non-functional requirements:** See TRD3.

## **TRD3.4: Check game status**

**Requirements**

**Description**: Return yes/no if the light game is finished.

**Actors**: System

**Pre-condition:** Have the scores been recently updated?

**Triggering Event:** Checks and returns status of the board.

**Sequence**: If the game is complete, end the game.

**Post Condition**: Have checked the game at least once per pass.

**Resulting Events:** The game is either finished or not.

**Alternative Scenarios:** Game finished or not?

**Non-functional requirements:** See TRD3.