



---

# Project Management: Briscola Team

Bonora Riccardo - 892867 (Group Leader)

Boschiero Oscar - 891051

Federico Nori - 890184

Hokkanen Arttu Juhi Jalmari - 1000430

Surian Raoul - 886732

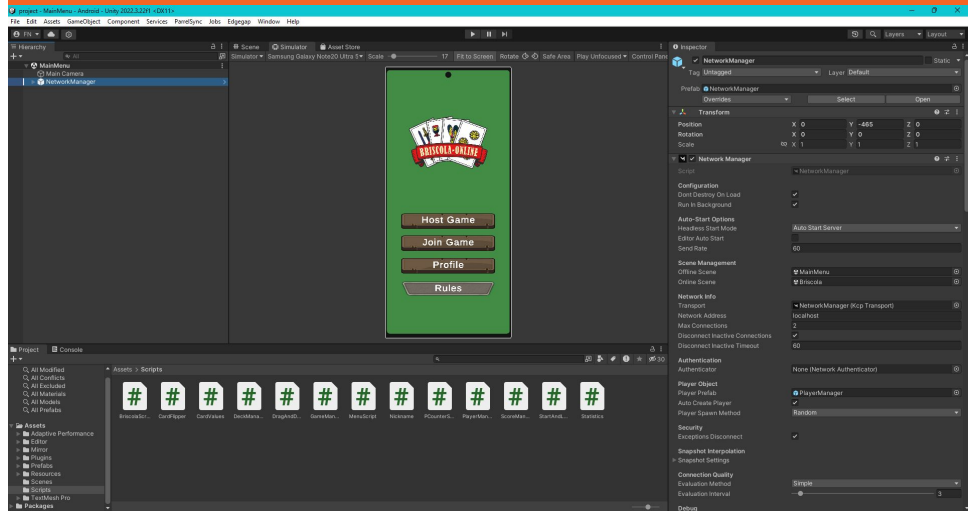
# Introduction

Briscola-Online is an app for Android phones created for enthusiasts of the classic card game Briscola. This application offers a platform where players can challenge each other in real-time in one-on-one mode.

The application not only allows you to play Briscola but also offers a range of additional features. Players can track their performance by recording wins, draws, and losses. Additionally, Briscola-Online allows you to customize your nickname.

For those who are new to the game or wish to refresh their memory, the app includes a section dedicated to the rules of Briscola, making it accessible to both beginners and veterans. With Briscola-Online, you can immerse yourself in one of Italy's most beloved card games and challenge your friends.

---



For the development of our Android app, we chose to use Unity, a cross-platform game engine that utilizes the C# programming language. We integrated the Mirror plugin to enable the development of multiplayer features within the application.

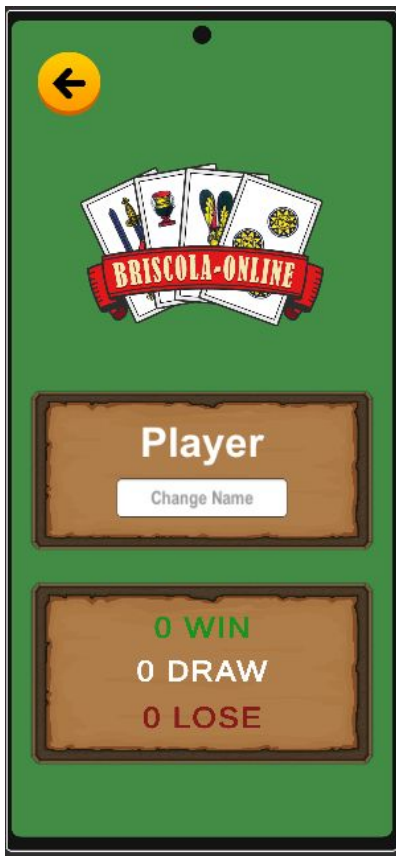
# Implemented features

- Host a game
- Join an existing game
- View profile (username and statistics)
- Displaying the rules

# Main Menu



# Profile section



# Rules section



# Implemented features

## Host Game

- Counter of connected players in the lobby
- Ability to start the game if both players are connected

## Join Game

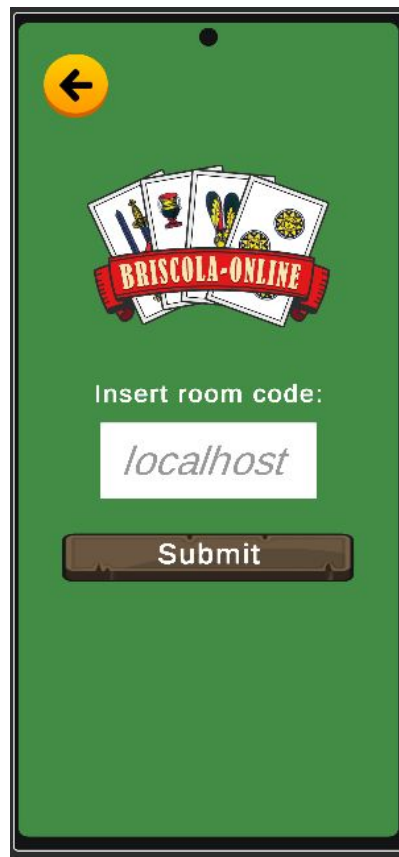
- Ability to join a game via code
- Ability to leave a game before it starts

---

## Host section



## Join section



# Implemented features

- Briscola game
- Show scores and current turn in the scoreboard during the game
- Show the number of remaining cards in the deck
- Show cards in each player hand
- Show the trump card for the current game
- Show end-game results



# Game start



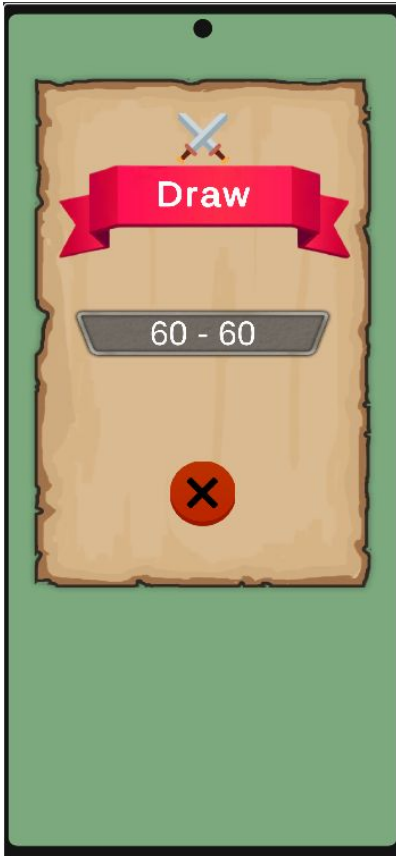
# Move done



## Winner screen



## Draw Screen



## Loser screen



# Tasks overview

We decided the number of tasks depending on the amount of work that was needed.

The first sprint we focused on understanding the core idea that we wanted to implement in our project.

The second and third sprint focused on the actual development of the application.

The final sprint we had fewer work points as we focused on refining the application.

---

✓	SCRUM-34	Identify the necessary graphic assets for the main menu inter...	DONE	🕒 Sprint 1	AH Arttu Hokkanen
✓	SCRUM-35	Do some sketches of what the game should be like	DONE	🕒 Sprint 1	AH Arttu Hokkanen
✓	SCRUM-37	Implement navigation between different menu options	DONE	🕒 Sprint 1	OB oscar boschiero
✓	SCRUM-40	Get github working	DONE	🕒 Sprint 1	R RB
✓	SCRUM-41	Create the match creation menu interface	DONE	🕒 Sprint 1	RS Raoul Surian
✓	SCRUM-43	create the interface for joining a match already existing	DONE	🕒 Sprint 1	RS Raoul Surian
✓	SCRUM-44	Implement logic for searching and joining a lobby	DONE	🕒 Sprint 1	FN FEDERICO NORI
✓	SCRUM-42	Implement logic for creating a lobby	DONE	🕒 Sprint 1	FN FEDERICO NORI
✓	SCRUM-36	Create the main menu user interface	DONE	🕒 Sprint 1	OB oscar boschiero
📌	SCRUM-68	As a player, I want to be able to create my lobby so that my f...	DONE		
📌	SCRUM-69	As a player, I want to be able to join an existing game so that...	DONE		

## Sprint #1

✓	SCRUM-55	Handle player disconnections	DONE	🕒 Sprint 2	FN FEDERICO NORI
✓	SCRUM-57	More asset research (Card PNG's and Sound effects)	DONE	🕒 Sprint 2	AH Arttu Hokkanen
✓	SCRUM-58	Implement deck system	DONE	🕒 Sprint 2	R RB
✓	SCRUM-59	Develop player hand management	DONE	🕒 Sprint 2	OB oscar boschiero
✓	SCRUM-63	Create the ppt for the second sprint	DONE	🕒 Sprint 2	RS Raoul Surian
✓	SCRUM-64	Organize the oral presentation for the second sprint	DONE	🕒 Sprint 2	RS Raoul Surian

## Sprint #2

✓	SCRUM-54	Synchronize player data	DONE	🕒 Sprint 3	FN FEDERICO NORI
✓	SCRUM-56	Implement the selection of the nickname (UI and Backend)	DONE	🕒 Sprint 3	RS Raoul Surian
✓	SCRUM-61	Develop game field and captured cards functionality	DONE	🕒 Sprint 3	RS Raoul Surian
✓	SCRUM-46	Creation of interfaces for the playing phase	DONE	🕒 Sprint 3	R RB
✓	SCRUM-60	Develop turn mechanics	DONE	🕒 Sprint 3	OB oscar boschiero
✓	SCRUM-45	Create the match interface for visualizing cards	DONE	🕒 Sprint 3	R RB
✓	SCRUM-39	Create a section for the game rules	DONE	🕒 Sprint 3	AH Arttu Hokkanen
📌	SCRUM-66	As a player, I want to be able to consult the rules	DONE		
📌	SCRUM-65	As a player, I want to be able to access the main menu of the...	DONE		
📌	SCRUM-67	As a player, I want to be able to choose my nickname	DONE		
📌	SCRUM-76	As a player, I want to be able to keep track of games won an...	DONE		

## Sprint #3

✓	SCRUM-51	Create interface to display final results and declare the winni...	DONE	🕒 Sprint 4	FN FEDERICO NORI
✓	SCRUM-52	Implement logic to calculate final scores at the end of the ga...	DONE	🕒 Sprint 4	R RB
✓	SCRUM-53	Add button or functionality to return to the main menu after ...	DONE	🕒 Sprint 4	RS Raoul Surian
📌	SCRUM-71	As a player, when I'm the owner of the lobby, I would like to ...	DONE		
📌	SCRUM-72	As a player, I want to be able to play Briscola	DONE		
📌	SCRUM-73	As a player, during the game, I would like the game to keep t...	DONE		
📌	SCRUM-74	As a player, I want final scores to be calculated at the end of ...	DONE		
📌	SCRUM-75	As a player, I want to be able to return to the main menu aft...	DONE		

## Sprint #4

# Burn-down charts

The burn-up graphs here overall show a steady completion of the work schedule, producing results that overall resolve the expected scope of the project.

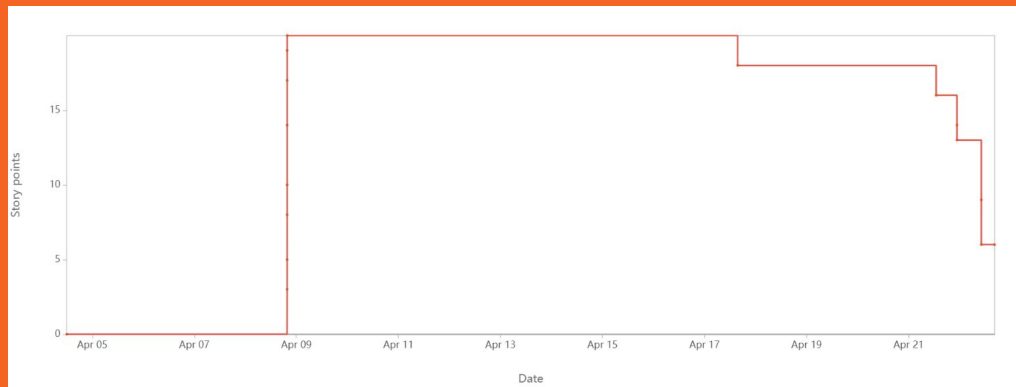
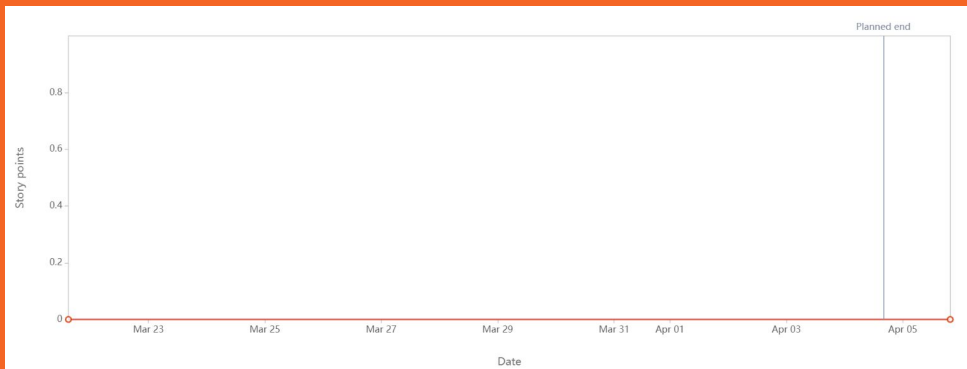
We had to resize the project a bit during the second sprint in order to ensure we produced a satisfactory result, but that allowed us to complete the project on time. In fact, the biggest problem was with the organization, even though for each sprint we tried to have at least three daily scrum meetings to achieve as much cohesion and organization as possible.

We would like to point out that the graph of the first sprint might be a little inaccurate as we were still unfamiliar with the Jira software.

---

# Jira burndown-chart

## Sprint #1



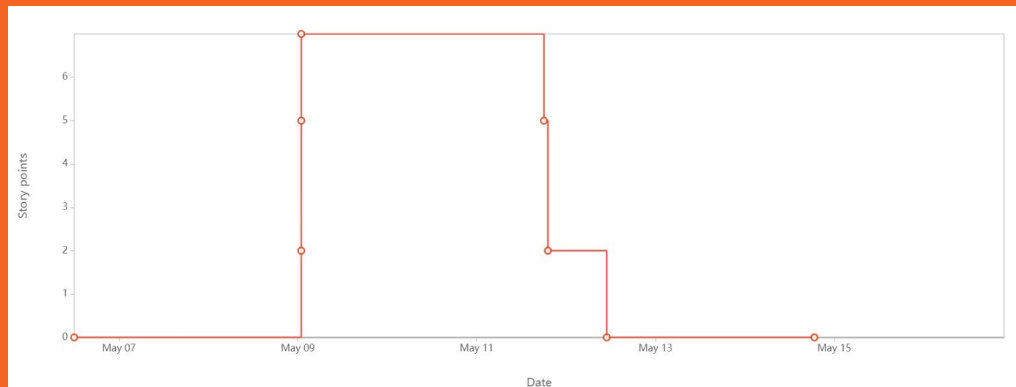
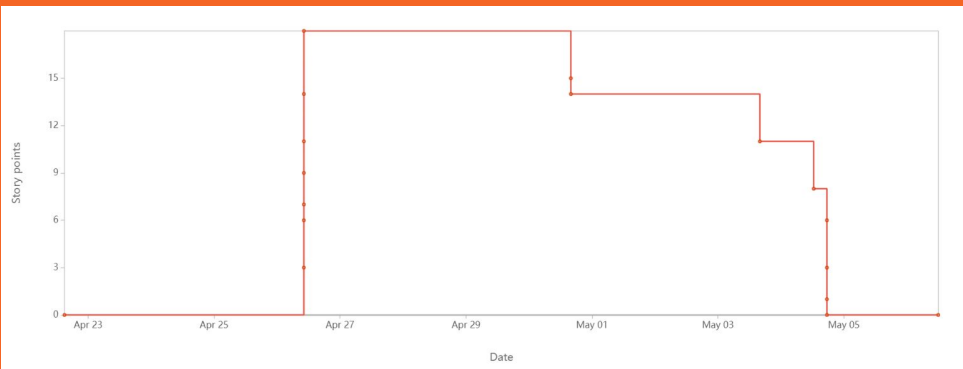
## Sprint #2

**Remaining work**  
Number of story points left to complete this sprint

**Guideline**  
Ideal burn rate

# Jira burndown-chart

## Sprint #3



## Sprint #4

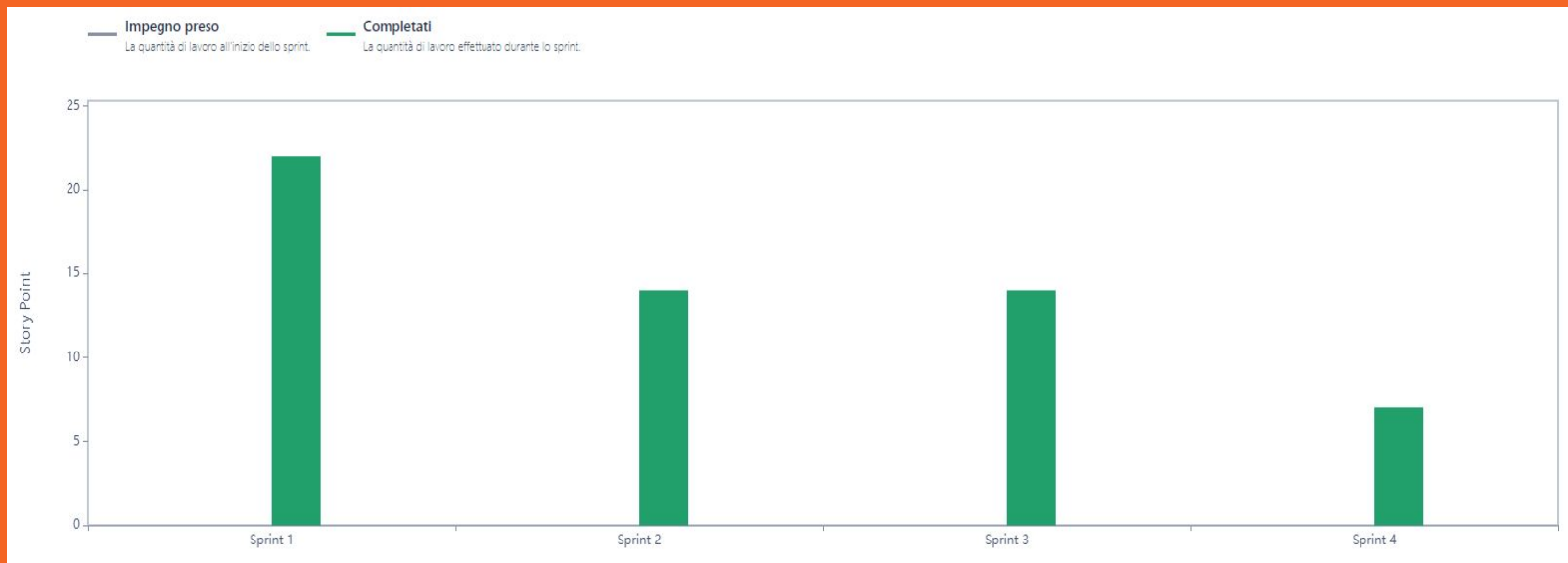
**Remaining work**  
Number of story points left to complete this sprint

**Guideline**  
Ideal burn rate

# Velocity

Here is the graph of the total amount of working days for each sprint.

Unfortunately, as you can also see from the graph, the commitment made is not present. In fact, since it was the first time we used Jira, we made mistakes in completing the sprints, and therefore it doesn't show up.





# Testing

The testing process has been a crucial element in software development and has occurred concurrently with the code implementation.

Developers were responsible for testing each implemented block of code. Subsequently, other team members were able to carry out tests on the interactions between the various components of the application.

Naturally, during the development phase of the application, several test cases did not pass on the first attempt. However, in the current version of the application, all test cases pass.

Our testing strategy was divided into two main phases, aimed at ensuring the reliability and robustness of the application.

## Unit Testing

Unit testing involves testing individual units of code to identify potential errors. This approach allows for easier identification and understanding of errors, confining them to specific portions of the application. It helps limit errors to interactions between individual units, simplifying the subsequent testing phase.

## Incremental Testing

This method is used to test the interaction between the various components or modules of an application. After Unit Testing, modules are gradually integrated into the larger system. At each step, their interaction with existing components is verified to ensure seamless integration.

---

Test Case	Precondition	Description	Input	Expected Output
T_01	-	Host of a "Briscola" game	Click on the "Host" button	Game hosted successfully and scene loaded correctly
T_02	A game is waiting for a player to join	Join a "Briscola" game	Click on the "Join" button, enter the code, and submit it	Game joined successfully
T_03	Being in a game that hasn't started yet	Leave a game that has not started yet	Click on the "Leave" button	Return to the main menu
T_04	Being the host of a game that hasn't started yet with two players	Start a game	Click on the "Start!" button	The game starts for both players
T_05	Game started	Shuffling the deck	-	Shuffled deck

Test Case	Precondition	Description	Input	Expected Output
T_06	Game started and deck shuffled	Dealing the cards	-	Cards dealt and present in the respective players' areas
T_07	Deal the first 3 cards per player	Show the trump card and the number of remaining cards	-	Trump card and the number of remaining cards displayed
T_08	Cards dealt	Moving one of your cards to the central area	Dragging	Card moved to the central area and unable to move it again
T_09	-	Turn rotation mechanism	-	Turns assigned and correctly respected by both players
T_10	One card from each player in the central area	Determination of the round winner	-	Cards removed from the central area and turn assigned correctly

Test Case	Precondition	Description	Input	Expected Output
T_11	Round terminated	Updating the score and current turn on the scoreboard	-	Score and current turn updated on the scoreboard
T_12	-	Synchronization of data between the two players	-	Data synchronized between the two players
T_13	All cards finished and hands played	Determination of the game winner	-	Final screen with results displayed
T_14	Results screen displayed	View of the winner, results, and return button to the main menu	-	Winner, results, and return to main menu button displayed correctly
T_15	Result screen displayed	Return to main menu	Click on the "X" button	Return to main menu successfully

Test Case	Precondition	Description	Input	Expected Output
T_16	Game ended	Saving player statistics changes	-	Changes saved successfully
T_17	-	Display profile section	Click on the "Profile" button	Profile section displayed
T_18	Profile section displayed	Display player statistics and username	-	Player statistics and username displayed
T_19	Profile section displayed	Changing player's username	Enter the new name in the text box and press enter	Username changed successfully
T_20	-	Display rules section	Click on the "Rules" button	Rules section displayed

# Source of Assets utilized

- **Button Set:**  
<https://assetstore.unity.com/packages/2d/gui/buttons-set-211824>
  - **Card shirts Lite:**  
<https://assetstore.unity.com/packages/2d/gui/card-shirts-lite-165698>
  - **Fantasy Wooden GUI:**  
<https://assetstore.unity.com/packages/2d/gui/fantasy-wooden-gui-free-103811>
  - **Win, Draw and Lose Red Ribbon:**  
[https://pngtree.com/freepng/conratulation-red-ribbon-with-confetti-and-golden-crown-winner-victory-vector\\_5604196.html](https://pngtree.com/freepng/conratulation-red-ribbon-with-confetti-and-golden-crown-winner-victory-vector_5604196.html)
  - **Simple Button Set:**  
<https://assetstore.unity.com/packages/2d/gui/icons/simple-button-set-01-153979>
  - **Cards:**  
<https://www.hiclipart.com/free-transparent-background-png-clipart-vhfso>
  - **Crossed Swords:**  
<https://emojipedia.org/crossed-swords>
  - **Logo:**  
<https://www.hiclipart.com/free-transparent-background-png-clipart-skrzu>
  - **Mirror:**  
<https://assetstore.unity.com/packages/tools/network/mirror-129321>
-

---

# End of presentation

Thank you for your attention!!

---