Askar Alisher, 220103194, 17-P(Backend)
Bekkalieva Ainara, 220103093, 17-P(Project Manager)
Zhalgas Daniyar, 220103080, 17-P(Frontend)
Sairambay Nurbol, 220103096, 17-P(Testing)
Toibayev Nursultan, 220103044, 17-P(AI integrator)
Sabit Ulan, 220103244, 17-P(Frontend)

# *Chooser game*

## Overview

**Chooser Game** is a party-friendly web app that randomly selects one participant from a group using a fun and engaging touch-based interface. Once a player is selected, the game can assign a task based on difficulty. It's perfect for quick party games, icebreakers, or making spontaneous decisions among friends.

**Website:** hosted on localhost

## Problem Statement:

1. Why do I need the Chooser project?
Because in group games and events, there is often a problem with organizing the selection of participants or tasks.
2. Why does this problem occur in group games?
Because the process of selecting participants or assignments is often subjective or requires a lot of time and effort to organize.
3. Why is the process of selecting participants or assignments subjective or time-consuming?
Because this is often done using manual random selection, which can lead to dishonesty or difficulty in control.
4. Why can random manual selection lead to dishonesty or embarrassment?
Because it is impossible to manually guarantee that each selection will be random, and this process may also be inefficient for large groups.
5. Why does the Chooser project solve this problem?
Chooser offers an automatic and fair way to randomly select participants or assignments, ensuring a fast and unbiased process for groups of people.
***Conclusion***:
The Chooser project is necessary because it simplifies the process of randomly selecting participants and tasks in groups, increasing its fairness, speed and transparency.

## Objective

The goal of the project is to create an interactive and entertaining game for groups of people. Chooser is an application for groups of friends, in which participants perform a variety of tasks, randomly selected. Users can also formulate individual requests to generate tasks using artificial intelligence.

## Features

- **Touch-to-Join Interface**: Players simply place a finger on the screen to join.
- **Random Selection**: One player is randomly chosen from the group.
- **Task Mode**: Assigns fun tasks to the selected player.
- **Difficulty Settings**: Choose from Easy, Medium, or Hard levels.
- **Timer**: Configurable countdown for task completion.
- **Minimal UI**: Clean dark-mode interface with easy navigation.
- **AI Custom tasks:** Sending request to AI.

## Technologies Used

| Name | Technology |
|---|---|
| Frontend | Vanilla JavaScript, HTML5 Canvas, Bootstrap 5 |
| Backend Integration | REST API with JSON |
| Storage | LocalStorage for user settings |
| Deployment | Web-based with Service Worker support |

| | |
|---|---|
| Additional | Pointer Events API, Web Audio API |
| Backend | Java 21, Spring Boot 3.4.5 |
| Database | PostgreSQL |
| Build Tool | Maven |
| Containerization | Docker, Docker Compose |
| Testing | JUnit, Spring Test |
| AI Integration | Cohere API for custom task generation |
| Prerequirsites | • Java 21 or higher<br>• Maven 3.6+<br>• Docker and Docker Compose (optional, for containerized deployment)<br>• PostgreSQL (if running without Docker) |

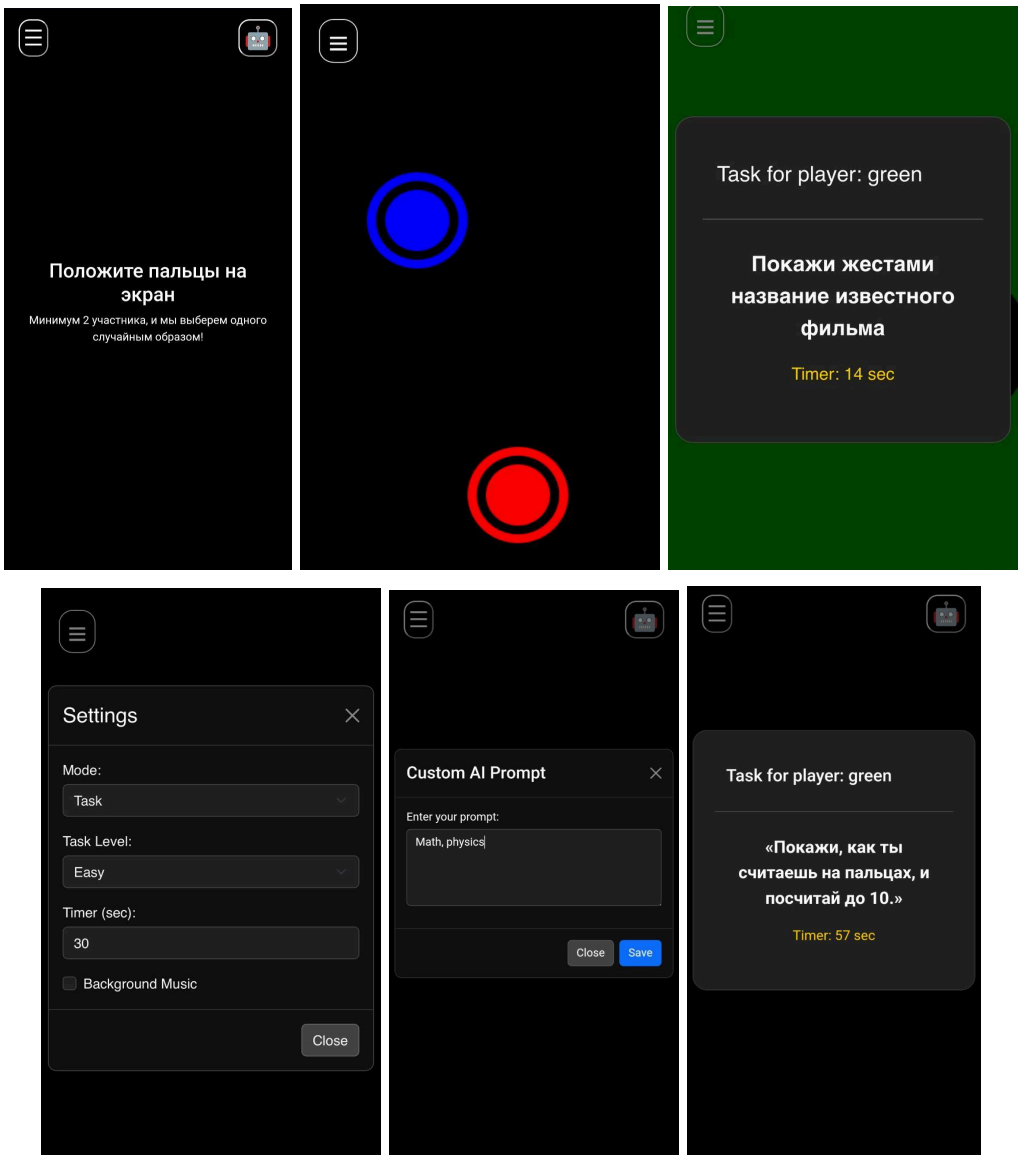
## How to Use

**Step-by-Step Instructions**

1. **Open the Website**
    - Visit the site
2. **Gather Players**

- At least **two people** should be present.

3. **Place Fingers on the Screen**
    - Each player touches the screen with one finger. Colored circles will appear (e.g., blue, red).
4. **Wait for Random Selection**
    - The app will **randomly pick one finger**, assigning it a color.
5. **Task Mode (Optional)**
    - Enable **"Task Mode"** from the menu.
    - Choose a **difficulty level** (Easy, Medium, Hard).
    - Set a **timer** (e.g., 30 seconds).
6. **Perform the Task**
    - The selected player receives a task (e.g., “Сделай смешное лицо” – *Make a funny face*).
    - Complete the task before the timer runs out.
7. **Custom AI Prompt:**
    - Sending request to AI to get task
8. **Play Again**
    - Lift your fingers and try another round!




## Settings Description

| Setting | Description |
|---|---|
| Mode | Select between random chooser or tack mode |
| Task Level | Adjusts the difficulty of the tasks(Easy, Medium, Hard) |

| | |
|---|---|
| Timer | Countdown timer for task execution |
| Background Sound | Enable or disable background sound |

## Frontend Interactions

## API Endpoints

**Start a game session:**

This endpoint starts a gaming session by selecting a random player or task to complete.

```
POST /api/v1/start
```

**Random Mode Example**:

- In this example, players with IDs 1, 2, 3, and 4 participate in the game.
- game Option.mode is set to RANDOM, which means that the system will randomly select one of the players.

```
{
    "playerIds": [1, 2, 3, 4],
    "gameOption": {
        "mode": "RANDOM"
    }
}
```

**Response**:

- SessionID is the identifier of the game session. In this case, null, because no session is created in random selection mode.
- choosenPlayerId is the ID of the selected player. In this example, the player with ID 3 is selected.
- task — a task, if there is one. In this case, since the mode is RANDOM, there is no task (null is passed).

```
{
    "sessionId": null,
    "choosenPlayerId": 3,
    "task": null
}
```

**Task Mode Example**:

- Here, players with IDs 1, 2, 3, and 4 also participate in the game.
- game Option.mode is set to TASK, which means that in addition to randomly selecting a player, a task will also be assigned.
- The Task Level is set to EASY, which means that the task will be easy.

```
{
    "playerIds": [1, 2, 3, 4],
    "gameOption": {
        "mode": "TASK",
        "taskLevel": "EASY"
    }
}
```

**Response**:

- SessionID is the identifier of the game session. In this case, a unique session ID has been generated.
- choosenPlayerId is the ID of the selected player. In this example, the player with ID 2 is selected.
- task — a task that is assigned to the selected player. In this case, the task is: "Sing a verse of a children's song" with an EASY difficulty level.

```
{
    "sessionId": "some-session-id",
    "choosenPlayerId": 2,
    "task": {
        "text": "Спой куплет детской песни",
        "level": "EASY"
    }
}
```

**Custom AI task example:**

This request demonstrates the creation of a custom task for the game, where users can interact with artificial intelligence (AI) to perform specific tasks. In this example:

- playerid: A list of the IDs of the participants who will participate in the task (in this case, these are the players with IDs 1, 2, 3 and 4).
- game Option: an object with game settings.

```
{
    "playerIds": [1, 2, 3, 4],
    "gameOption": {
        "mode": "TASK",
        "taskLevel": "MEDIUM",
        "prompt": "что-то связанное с танцами"
    }
}
```

## Testing:

| Test ID | Description | Preconditions | Test steps | Expected result | Actual result | Status | Priority |
|---|---|---|---|---|---|---|---|
| TC_001 | Checking if players have been added to the game via the Touch-to-Join interface. | The device is connected to the Internet. | 1. Open a website.<br>2. Each player places a finger on the screen.<br>3. Confirm that a colored circle appears for each player. | A colored circle (eg blue, red) appears for each player. | A colored circle (eg blue, red) appears for each player. | Pass | Pass |
| TC_002 | Checking the random selection of a participant. | Several participants in the game. | 1. Once players have joined, click the "Select Participant" button. 2. Confirm that one of the participants has been randomly selected. | One player is selected at random and assigned a task. | One player is selected at random and assigned a task. | Pass | Pass |
| TC_003 | Testing the operation of the task mode with a choice of difficulty level. | Players have joined, a task has been selected. | 1. Activate "Task Mode" via the menu.<br>2. Select the task difficulty level (Easy, Medium, Hard).<br>3. Set the timer (e.g. 30 seconds).<br>4. Confirm that the selected difficulty level affects the task type. | The task is assigned depending on the selected difficulty level. | The task is assigned depending on the selected difficulty level. | Pass | Pass |
| TC_004 | Testing the operation of the task request function via AI. | The player wishes to use AI to generate the task. | 1. Send a request to the AI to receive a task.<br>2. Confirm that the AI generates an appropriate task.<br>3. Check that the task is logical and feasible. | AI generates a task that can be completed. | AI generates a task that can be completed. | Pass | Pass |