

## PDF Document for Project 9- Nand2Tetris Course

### Serving:

- Liram Kreiner: [liram.kreiner@post.runi.ac.il](mailto:liram.kreiner@post.runi.ac.il) 323866012
- Yoav Cohen Nehemia: [yoav.cohennehemia@post.runi.ac.il](mailto:yoav.cohennehemia@post.runi.ac.il) 323100107

### Concept / idea of our game:

Backgammon is a two player strategy game recreated in Jack, capturing the essence of this timeless classic. Our implementation features an interactive and visually engaging digital board, dynamic dice rolls, and smooth piece movement. Players can compete to move their pieces off the board, leveraging strategy and luck while adhering to traditional Backgammon rules. The game also includes real-time win detection and intuitive player interactions for seamless gameplay.

### System Architecture: Explanation for every Jack file

#### **Main.jack**

- **Purpose:** The central controller of the game. Initializes the game components, manages the game flow, and orchestrates interactions between modules like the board, players, and zones.
- **Main Logic:** Starts the game loop, monitors player actions, and ensures the game's progression according to the rules of Backgammon.

#### **Board.jack**

- **Purpose:** Represents the game board and handles interactions between columns, pieces, and zones.
- **Main Logic:** Manages board updates, validates moves, and interacts with other components like `Piece` and `Column`.

#### **Backgammon.jack**

- **Purpose:** Encodes the rules and logic of Backgammon.
- **Main Logic:** Implements rules for dice rolls, piece movement, capturing, and scoring, ensuring fair gameplay.

#### **Column.jack**

- **Purpose:** Represents a column on the Backgammon board.
- **Main Logic:** Handles piece placement, movement, and interactions such as capturing opponents' pieces.

#### **Piece.jack**

- **Purpose:** Represents individual game pieces.
- **Main Logic:** Encodes piece attributes (e.g., color, position) and handles movement and rendering on the board.

**Dice.jack**

- **Purpose:** Simulates dice rolls for gameplay.
- **Main Logic:** Uses the `Random.jack` module to generate numbers, simulating a dice roll, and outputs results to the screen.

**Random.jack**

- **Purpose:** Generates random numbers for use in dice rolls.
- **Main Logic:** Implements a random number generator (RNG) for producing unbiased dice outcomes.

**EatZone.jack**

- **Purpose:** Represents zones where captured pieces are placed.
- **Main Logic:** Tracks and displays captured pieces based on their color.

**WinZone.jack**

- **Purpose:** Represents zones where pieces are moved once they complete their path.
- **Main Logic:** Handles the arrangement of pieces in the win zone and determines when a player has won.

**Arrow.jack**

- **Purpose:** Manages arrows used for navigation or selection.
- **Main Logic:** Draws and updates arrow positions, guiding player actions during gameplay.

**ListofPiece.jack**

- **Purpose:** Implements a linked list for managing collections of pieces.
- **Main Logic:** Tracks multiple pieces in a column or zone and provides utility functions for list manipulation.

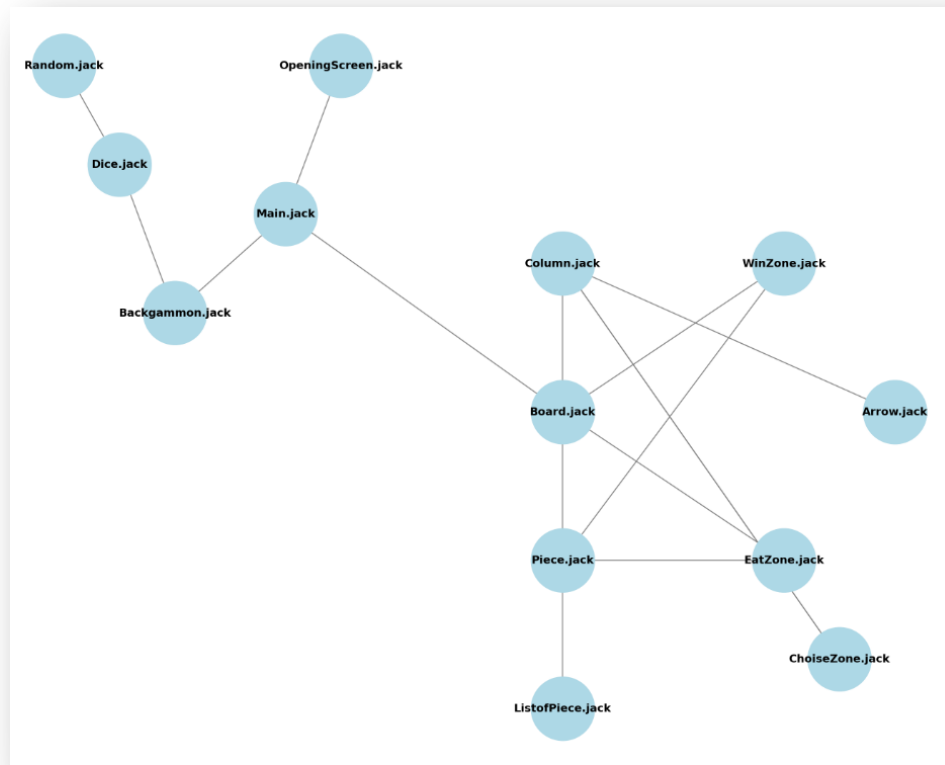
**ChoiseZone.jack**

- **Purpose:** Handles zones where players can make decisions.
- **Main Logic:** Manages player choices and highlights selected options on the screen.

**OpeningScreen.jack**

- **Purpose:** Displays the initial screen of the game, introducing the players to Backgammon.
- **Main Logic:** Renders the opening visuals, provides options to start or quit the game, and waits for user input to proceed to the main gameplay.

## System Architecture Diagram:



### Motivation:

We chose to create Backgammon because it is a timeless classic that combines strategy and luck, making it one of the most engaging two player games. Growing up, we both enjoyed playing Backgammon with friends and family, and we wanted to recreate that nostalgic experience while applying the programming concepts we've learned in the Nand2Tetris course. This project allowed us to combine our technical skills with creative design, resulting in a rewarding challenge. Additionally, we believe that games like Backgammon foster social interaction and bring people together, making them an excellent choice for a collaborative and interactive experience.

Throughout the development of our Backgammon game, we utilized Git to manage our workflow and collaborate efficiently. Using Git allowed us to maintain a clean version history, resolve conflicts seamlessly, and merge our contributions effectively.

### Link to our Video:

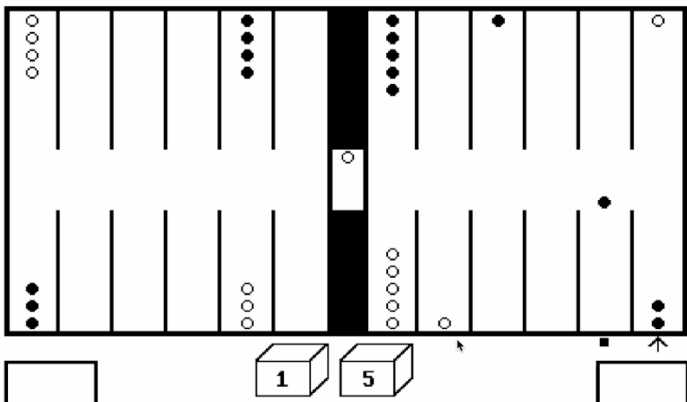
Video link- [https://drive.google.com/drive/folders/13-7FE3BK7akZQIDt-5aVi1x4CfQ5Ps0i?usp=drive\\_link](https://drive.google.com/drive/folders/13-7FE3BK7akZQIDt-5aVi1x4CfQ5Ps0i?usp=drive_link)

Before starting the game

Please enter two 5 digit numbers

First: ■

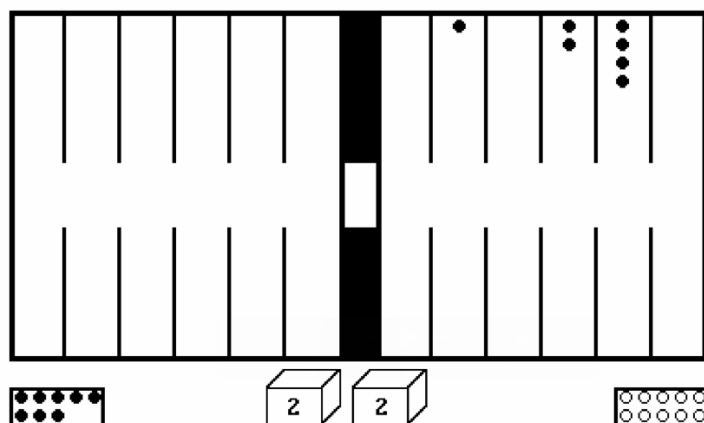
Backgammon in Jack language  
By Yoav Cohen and Liram Kreiner



Turn:  
Black

Keys:  
enter-pick  
esc-cancel  
<,>-move  
d-roll  
w-take  
out  
space-end  
turn

Congrats, White Player Won The Game!



Turn:  
White

Keys:  
enter-pick  
esc-cancel  
<,>-move  
d-roll  
w-take  
out  
space-end  
turn