1. Carcassonne

Description:
Carcassonne is a game in which a single player attempts to build a map within the limits of a board using squared tiles. The map will be constituted by tiles representing roads, abbeys and cities. Each of these types of tiles give certain number of points to the player. The goal is for the player to build the map that scores the highest.

Requirements:
-The total number of tiles which the player can play is N^2 where N is the length of a side of the board.
- The board must be a 11x11 square.
- The player can put one (1) tile on the board per turn
- At the start of the game the player gets four (4) random tiles to select from and will be dealt a new tile each turn, replacing the previously played tile.
- The player will never know what tile will be the next.
- The player should be able to discard their hand every 5 turns
- The game is played until the player can no longer play a tile, either because there are no empty places or there are no valid placements for the tiles at hand.
- If the player cannot play a tile in their hand, the game will end unless the discard is available, in which case, the player can do a discard.
- This game uses von Neumann Neighborhoods for adjacency: A tile is said to be adjacent to another tile if they have a Manhattan distance of 1 unit from each other.
- A chain of tiles is a set of tiles where every tile is adjacent to at least one tile in the chain
- Road tiles score 1 point per tile and can only be placed in positions adjacent to road tiles, so the chain keeps getting larger. The game starts with a single road tile in the center.
- Abbey tiles score 1 point per tile surrounding the abbey tile, so the maximum score a player can get is 8 if an abbey tile is completely surrounded by other tiles. Abbeys must be placed adjacent to any other tile.
- City tiles score 2 points per tile and can only be placed adjacent to cities, roads and/or abbey tiles. City chains award an extra point for each city in the chain
- Tile dealing should not be completely random. Roads should be more likely to appear than cities, and abbeys should be scarce. No fifteen (15) turns can happen without an abbey and at least three (3) cities. How this is done is up to the implementor.

Bonus:
-You can add direction to road tiles, so each road tile indicates in which way must the road grow: right-left, up-down, four-way crossing, three-way crossing, or corner tile. The game must start with a four-way crossing road tile in the center
- Allow boards of different odd sizes.
- Allow different random tile generation algorithms

2. Othello

Description:
Othello is a game in which two players attempt to gain control of a square board by placing tokens strategically. Players take turns placing tokens to take control of the tokens.

Requirements:
- Each player will start the game with 32 tokens.
- The board is an 8x8 square.
- The game starts with the four (4) center spaces already in play, with two (2) white tokens and two (2) black tokens diagonal from each other.
- This game uses von Neumann Neighborhoods for adjacency: A token is said to be adjacent to another if they have a Manhattan distance of 1 unit from each other.
- A chain is a set of tokens where every token is adjacent to at least one token of the same color in the chain, in a single direction (either horizontally or vertically).
- The player with black tokens starts the game.
- Each turn a player can place one of its tokens adjacent to one of its opponent's tokens.
- The game is played until no movements are possible.
- It is possible that at the end of a game, the board isn't filled with tokens, considering the allowed movements per turn.
- When the placement of a token causes a chain of tokens of the other color to start and begin with the color of the placed token, all the tokens in the chain will change color.
- Every move must cause a chain to change color.
- Should a player lose all the tokens, the game will end and said player will lose the game.

Bonus:
- Allow diagonal chains
- Allow different board sizes (must be even).

3. Checkers

Code Sandbox Template: https://codesandbox.io/s/checkers-ui-0ps4u5

DO NOT MODIFY ANYTHING INSIDE THE "UI" FOLDER. WE WILL NOT PROVIDE SUPPORT FOR ANYTHING INSIDE THE "UI" FOLDER. ALL YOUR CODE MUST BE PLACED INSIDE THE "solucion" FOLDER AND INSIDE THE "events.ts" FILE.

Description:
This game will be a remake of the traditional Chinese checkers with a few changes.

Requirements:
-The board is an 8x8 square
-Each player will have 11 pawns and a queen.
-The initial board must place the pawns and queen as shown below:

X		X		X		X	
	X		X		X		X
X		X		X		X	
	X		X		X		X
X		X		X		X	
	X		X		X		X

- Queens start on opposite corners of the board
- Pieces can capture other pieces of opposite color by "jumping" diagonally over it
- To capture, a piece must be diagonal to the piece and have an empty space in the same direction, next to the piece being captured
- Pieces move diagonally towards the opposite player's side. The queen can move diagonally in any direction.
- Should a queen ever be captured, the owner of the queen must choose a new queen from among any of the idle pieces. A piece is considered "idle" if it is among the owned pieces furthest from the opposite side of the board.
-The player that places its queen on the first line of the opponent battlefield will win the game.

Bonus:
- Allow multiple jumps
- Force capture of pieces

4. Bowling calculator

Code Sandbox Template: https://codesandbox.io/s/bowling-alley-ui-eucr6q

DO NOT MODIFY ANYTHING INSIDE THE "UI" FOLDER. WE WILL NOT PROVIDE SUPPORT FOR ANYTHING INSIDE THE "UI" FOLDER. ALL YOUR CODE MUST BE PLACED INSIDE THE "solucion" FOLDER AND INSIDE THE "events.ts" FILE.

Description:
The task is to make a system for tracking scores in a bowling alley.

Requirements:

- A game of bowling consists of 10 turns
- Each player throws the bowling ball twice each turn
- The system must receive the number of pins that were knocked over in each throw, sequentially.
- The score is the total amount of pins a player has knocked over up to that turn
- A spare is when in any given turn a player knocks all ten pins on the second roll
- When a spare is achieved, the score for that round will be ten (10) plus the current score plus the number of pins knocked over in the next roll.
- A strike is when in any given turn a player knocks all ten pins on the first roll
- When a strike is achieved, the score for that round will be ten (10) plus the current score plus the number of pins knocked over in the next two rolls.
- The last round awards the player an extra roll if and only if the player achieves a spare or a strike. If a strike is achieved, the turn will not end, and an extra roll will be awarded. Should a strike or a spare be achieved in the extra roll, it will only be worth 10 points. Should a strike be achieved in the second roll, only the next roll (the extra roll) will be added to the score of that strike.
- Should three (3) consecutive strikes be achieved, an extra twenty (20) bonus points must be added. For every extra consecutive strike, ten (10) more bonus points are to be added.
- The score board must show a slash (/) on the second roll square should a spare be achieved, a capital X on the second roll square should a strike be achieved, and a double capital X when a strike grants bonus points.
- A gutter ball is when no pins are knocked over.
- Should there ever be four (4) consecutive gutter balls, ten (10) points should be deducted from the score.
- Six (6) consecutive gutter balls will disqualify a player and end their game. Their score will remain as it were prior to the last gutter ball. Other players may continue as normal.
- Ten (10) gutter balls will disqualify a player and end their game. Their score will remain as it were prior to the last gutter ball. Other players may continue as normal.

Bonus:

- Allow different scoring systems