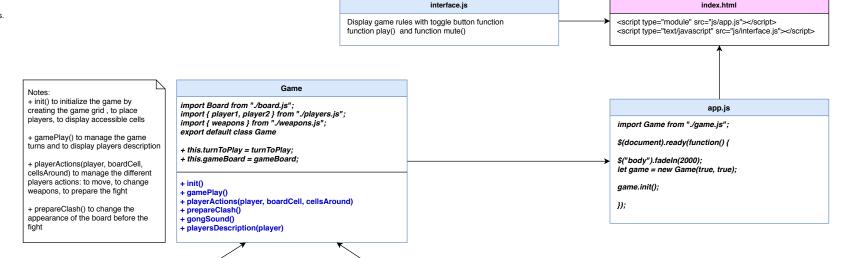
p6game new

Turn-based JS board game. Diagram Class and methods.

https://github.com/lana-rodion/p6game_new

18 july 2020



Notes:

- + createGrid(width, height) defines cell coordinates, to push cells in columns and row with for loop
- + randomCell() to return random cell with coordinates x and y, called randomNumber(0, this.width)
- + randomPlayers(player) to place random player in random cell, called getAdjacentCells(cell) to verify if adjacent Cells and the cell of player placement are not occupied by other player
- + obstacles() inserts the obstacle in random Free Cell
- + weaponsArr() to place the weapon in the random Free Cell
- + getAdjacentCells(cell) returns all the cases adjacent to a player cell
- + getAccessCellsAxis(cell, nbOfAccessCell, horizontal, axis) returns an array of the accessible cells using the direction
- + getAccessibleCells(cell, nbOfAccessCell) concats accessibleCells array to return all cells accessible by the player

+ isFree() checks if this cell is not occupied by an obstacle or a player

Board

import Cell from "./cell.js" export default class Board

- + this.weapons = weapons;
- + this.player1 = player1;
- + this.player2 = player2;
- + this.width = null;
- + this.height = null; + this.cells = []:
- + createGrid(width, height)
- + randomNumber(min, max)
- + randomCell()
- + players()
- + randomPlayers(player)
- + obstacles()
- + weaponsArr() + randomFreeCell()
- + getAdjacentCells(cell)
- + cellExist(x, v)

+ this.x = x;

+ isFree()

- + getAccessibleCells(cell, nbOfAccessCell)

+ getAccessCellsAxis(cell, nbOfAccessCell, horizontal, axis) Cell export default class Cell + this.y = y; + this.element = element; + this.obstacle = false; + this.player = null; + this.weapon = null;

Player

import { weapon1 } from "./weapons.js";

- + this name = name:
- + this.nickname = nickname;
- + this.weapon = weapon1;
- + this.life = 100;
- + this.currentCell = null;
- + this.defense = false:

export let player1 = new Player(name, nickname); export let player2 = new Player(name, nickname);

- + move(newCell)
- + changeWeapon(player) + isPlayerAround(cellsAround)
- + heroTarget(target)
- + heroDefense()
- + endGameModal()
- + gongSound()
- + gameOver() + scoreLife()
- + fight(target)
- + restart()

Weapon

export let weapons = [];

- + this.name = name:
- + this.damage = damage:
- + this.nickname = nickname:

export let weapon1 = new Weapon(name, damage, nickname)

Notes:

- + move(newCell) to move player and change the previous cell property
- + changeWeapon(player) to exchange the player weapon into the cell weapon
- + isPlayerAround(cellsAround) checks if there is a player in cellsAround
- + heroTarget(target) to change the appearance of the player who is a target in the fight and to hide buttons
- + heroDefense() to give the choice of to attack or defend
- + gameOver() to finish the game if one player has not life points and to call modal of endGameModal()
- + scoreLife() to calculate life points
- + fight(target) to manage the fight, to count fight damages on click