## p6game\_new

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

https://github.com/lana-rodion/p6game\_new

Notes:

player

random Free Cell

the random Free Cell

using the direction
+ getAccessibleCells(cell, nbOfAccessCell)

+ createGrid(width, height) defines cell

coordinates, to push cells in columns

+ randomCell() to return random cell

with coordinates x and y, called

+ randomPlayers(player) to place

getAdjacentCells(cell) to verify if

random player in random cell, called

adjacent Cells and the cell of player

placement are not occupied by other

+ obstacles() inserts the obstacle in

+ weaponsArr() to place the weapon in

+ getAdjacentCells(cell) returns all the

concats accessibleCells array to return

+ this.obstacle = false; + this.player = null;

+ this.weapon = null;

+ isFree()

all cells accessible by the player

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

cases adjacent to a player cell

+ getAccessCellsAxis(cell, nbOfAccessCell, horizontal, axis) returns an array of the accessible cells

randomNumber(0, this.width)

and row with for loop

18 july 2020

## Game Notes: + init() to initialize the game by import Board from "./board.js"; creating the game grid, to place import { player1, player2 } from "./players.js"; players, to display accessible cells import { weapons } from "./weapons.is"; export default class Game + gamePlay() to manage the game turns and to display players description + this.turnToPlay = turnToPlay; + this.gameBoard = gameBoard; + playerActions(player, boardCell, cellsAround) to manage the different players actions: to move, to change weapons, to prepare the fight + gamePlay() + playerActions(player, boardCell, cellsAround) + prepareClash() to change the + prepareClash() appearance of the board before the + gongSound() fight + playersDescription(player) Board Player import Cell from "./cell.js" import { weapon1 } from "./weapons.js"; export default class Board + this name = name: + this.weapons = weapons; + this.nickname = nickname; + this.player1 = player1; + this.weapon = weapon1; + this.player2 = player2; + this.life = 100; + this.width = null; + this.currentCell = null; + this.height = null; + this.defense = false: + this.cells = []: export let player1 = new Player(name, nickname); export let player2 = new Player(name, nickname); + createGrid(width, height) + randomNumber(min, max) + move(newCell) + randomCell() + changeWeapon(player) + isPlayerAround(cellsAround) + players() + randomPlayers(player) + heroTarget(target) + obstacles() + heroDefense() + weaponsArr() + endGameModal() + randomFreeCell() + gongSound() + getAdjacentCells(cell) + gameOver() + cellExist(x, v) + scoreLife() + getAccessCellsAxis(cell, nbOfAccessCell, horizontal, axis) + fight(target) + getAccessibleCells(cell, nbOfAccessCell) + restart() Cell Weapon export default class Cell export let weapons = []; + this.x = x;+ this.name = name: + this.y = y; + this.element = element; + this.damage = damage: + this.nickname = nickname:

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

index.html

<script type="text/javascript" src="js/interface.js"></script>

app.js

<script type="module" src="js/app.js"></script>

import Game from "./game.js";

\$(document).ready(function() {

let game = new Game(true, true);

\$("body").fadeIn(2000);

game.init();

+ move(newCell) to move player and

+ changeWeapon(player) to exchange

the player weapon into the cell weapon

+ isPlayerAround(cellsAround) checks

change the previous cell property

if there is a player in cellsAround

+ heroTarget(target) to change the

appearance of the player who is a

to attack or defend

target in the fight and to hide buttons

+ heroDefense() to give the choice of

+ gameOver() to finish the game if one

player has not life points and to call

+ scoreLife() to calculate life points

+ fight(target) to manage the fight, to

modal of endGameModal()

count fight damages on click

*});* 

Notes: