Game Documentation

• Overview

This JavaScript code constitutes a simple game with the following modules:

1. Bullet.js
2. BulletController.js
3. Box.js
4. Player.js
5. Game.js
6. StartGame.js

• Usage

- ❖ The game is initiated by creating an instance of the **Game** class.
- ❖ The initial setup is handled by creating an instance of the **StartGame** class.
- ❖ The game can be played by controlling the player with arrow keys and shooting with the spacebar.

Bullet.js

Class: Bullet

This class represents a bullet in the game.

Properties:

- **canvas**: The HTML canvas element.
- **x**: X-coordinate of the bullet.
- **y**: Y-coordinate of the bullet.
- **velocity**: Speed of the bullet.
- **bulletColor**: Color of the bullet.
- **width**: Width of the bullet.
- **height**: Height of the bullet.

Methods:

- **draw(theContext)**: Draws a single bullet on the canvas.

BulletController.js

Class: BulletController

This class manages the bullets in the game.

Properties:

- **bullets**: An array to store bullet instances.
- **timeTillNextBulletAllowed**: Countdown timer for controlling bullet firing.
- **canvas**: The HTML canvas element.
- **maxBulletsAtTime**: Maximum number of bullets allowed on the canvas.
- **bulletColor**: Color of the bullets.
- **score**: Player's score.

Methods:

- **draw(theContext)**: Draws all the bullets on the canvas.
- **shoot(x, y, velocity, timeTillNextBulletAllowed)**: Initiates the shooting of a bullet.
- **checkBulletCollisions(boxes, cols, rows, leftEdge, rightEdge)**: Checks for collisions between bullets and boxes.
- **destroySurroundingBoxes(boxes, index, cols, leftEdge, rightEdge)**: Destroys surrounding boxes of a red box.

Box.js

Class: Box

This class represents a box in the game.

Properties:

- **canvas**: The HTML canvas element.
- **x**: X-coordinate of the box.
- **y**: Y-coordinate of the box.
- **type**: Type of box (green/red).
- **width**: Width of the box.
- **height**: Height of the box.
- **image**: Image object for box appearance.
- **isDestroyed**: Flag to track whether the box is destroyed.

Static Methods:

- **createBoxes(canvas, numRows, numCols, gap)**: Creates an array of boxes for the game.
- **drawBoxes(boxes, theContext)**: Draws all the boxes on the canvas.
- **moveDown(boxes, theContext, velocity)**: Moves the boxes down on the screen.

Methods:

- **draw(theContext)**: Draws a single box on the canvas.

Player.js

Class: Player

This class represents the player in the game.

Properties:

- **rightPressed**: Flag indicating if the right arrow key is pressed.
- **leftPressed**: Flag indicating if the left arrow key is pressed.
- **shootPressed**: Flag indicating if the shoot key is pressed.
- **canvas**: The HTML canvas element.
- **velocity**: Speed of player movement.
- **bulletController**: Instance of **BulletController**.
- **x**: X-coordinate of the player.
- **y**: Y-coordinate of the player.
- **width**: Width of the player.
- **height**: Height of the player.
- **image**: Image object for player appearance.

Methods:

- **draw(theContext)**: Draws the player and initiates shooting if shoot key is pressed.
- **update()**: Updates the state of the player.
- **handleKeyDown(event)**: Handles keydown events for player movement and shooting.
- **handleKeyUp(event)**: Handles keyup events for player movement and shooting.

Game.js

Class: Game

This class orchestrates the game logic.

Properties:

- **notStart**: Flag indicating if the game has started.
- **leftEdge**: Array containing indices of left edges in each row.
- **rightEdge**: Array containing indices of right edges in each row.
- **rows**: Number of rows in the game.
- **cols**: Number of columns in the game.
- **gap**: Gap between boxes.
- **canvas**: The HTML canvas element.
- **theContext**: Canvas rendering context.
- **backGround**: Image object for the game background.
- **remainingTime**: Remaining time for the game.
- **fallingSpeed**: Speed of falling boxes.
- **velocity**: Speed of player and bullet movement.
- **timerInterval**: Interval for updating the timer.
- **playerBulletController**: Instance of **BulletController**.
- **player**: Instance of **Player**.
- **boxes**: Array of box instances.
- **gameLoop**: Interval for the main game loop.

Methods:

- **setupEventListeners()**: Sets up event listeners for key events.
- **handleKeyDown(event)**: Handles keydown events for the player.
- **handleKeyUp(event)**: Handles keyup events for the player.
- **startGame()**: Initiates the main game loop.
- **update()**: Updates the game state.
- **draw()**: Draws the game elements on the canvas.
- **checkFallingBoxes()**: Checks if any box has reached the bottom.
- **checkHittingAllBoxes()**: Checks if all boxes are destroyed.
- **winGame()**: Displays a win message and resets the game.
- **loseGame()**: Displays a lose message and resets the game.
- **updateLocalStorage()**: Updates player scores in local storage.
- **popUpMessage()**: Displays a welcome message for returning players.

StartGame.js

Class: StartGame

This class handles the initial setup of the game.

Properties:

- **nameStorage**: Array to store player names and scores.
- **playButton**: Play button element.
- **startButton**: Start button element.
- **queryString**: Query string from the URL.
- **level**: Game level obtained from the URL.
- **userName**: Input field for entering the player's name.

Methods:

- **handlePlayButtonClick()**: Handles the play button click event.
- **handleStartButtonClick()**: Handles the start button click event.