ITS
TECNOLOGIE
DELL'INFORMAZIONE E
DELLA COMUNICAZIONE

Backend System Integrator

Unità Formativa: Programmazione web - front-end

Docente: Shadi Lahham

Titolo argomento: Progetto Balls

AIM

Create an application that displays interactive bouncing balls and a navigation panel to control their behavior

REQUIREMENTS

Structure and behavior

Create an application that allows users to create and interact with bouncing balls, similar to examples demonstrated in class and the included screenshots and with the following requirements:

- new balls should be generated when the use clicks and drags the mouse

- the speed of the balls should depend on the speed of the mouse movement

- the direction of the balls should depend on the direction of motion of the mouse

- the balls should bounce at the edges of the container

- scrollbars should not appear when the balls hit the edges of the container

- the ball movement should have some physicality (wight, gravity, speed)

- it should be possible to 'freeze' and 'resume' the animation

- the user should be able to choose the 'skin' of the balls, colors, images, etc

- the user should be able to make the balls bigger or smaller with keyboard shortcuts and onscreen buttons

- the user should be able to remove balls (e.g. by clicking on them or on a button)

- the user should be able to to increase/decrease the speed of balls with keyboard shortcuts and onscreen buttons

- the page should be page responsive, meaning that when the user changes the size of the page, the balls never go out of their container and all page controls are repositioned to fit the viewport

- add an onscreen counter that displays the current number of balls

Performance

performance is critical for an application of this type. Rather than using setTimeout or setInterval consider methods that produce smoother animations such as the requestAnimationFrame() documentation.

You can also find some examples here: Animating with javascript: from setInterval to requestAnimationFrame

Pay attention to **performance**!

- adding more balls should not have a big impact or slow down the application

- the application should not slow or stutter when switching browser tabs

Optional

You can add any other features that you think will make your application more interesting:

- making the application more game like, for example by adding rules

- addling backgrounds and images

- adding sounds or music

- generate balls of random sizes and colors

Important

- Variables should have meaningful names

- The code should be well documented

- The code should be well indented

- The data structures should be effective and the code should be efficient

- You need to be able to explain each line of your code

Bonus (extra points)

Bonus 1: Magnet

Right clicking and holding the mouse attracts all the balls to the position of the mouse

When releasing the mouse button the balls return to their original behavior

Bonus 2: Ball trails

Balls should show shadow trails behind them to show the previous positions they were in. Search for "mouse trails" images to understand what this means

Bonus 3: Gravity

Add a slider to adjust gravity and impact the ball movement. You can add other factors such as air resistance for example

Help

Plan your data structures and functions before starting

Use arrays, objects and any other structures you think are necessary

Use a config object to store all the application settings

IIFEs (recommended)

Use IIFEs to make your code in multiple javascript files scope-safe

Immediately invoked function expression

I Love My IIFE

Immediately-Invoked Function Expression (IIFE)

The many ways to write an IIFE

Compatibility

The project should be tested and work properly on: **Chrome, Firefox, IE11**

Compatibility with other browsers is a nice **bonus**

Libraries, frameworks and language features

Do not use any external libraries or frameworks.

You have to write all the code yourself.

If you use any language features not seen in class, they have to be justifiable, correct and tested.

- You may use **<canvas>** if you wish, however if you decide to use it, make sure that you study it in detail, write all the javascript code yourself, **do not use** any external libraries and also make sure that your code is optimized for performance

- Using **<canvas>** will not give you bonus points, but will help you think more about application performance

Documentation and validation

Comments and code documentation

- All HTML, CSS and JS files should contain comments and be well documented

- JSDoc header documentation for every file

- JSDoc documentation for every function

- CSS files (if used) should have a header and contain comments where needed

- HTML files should contain comments to indicate important sections

- Follow all the comments and documentation requirements in Appendix 01

Validation

HTML files should be validated https://validator.w3.org/

CSS files should be validated https://jigsaw.w3.org/css-validator/

JSDoc documentation (required)

- Generate a JSDoc documentation for your code and put it in a folder called /JSDoc

Readme

Include a readme.md file that includes at least the following sections

- Introduction / Project description
- Usage (how to set up, run and use the application)
- Configuration and technical characteristics
- Files and project structure
- Features delivered
    - feature 1: description of feature 1
    - feature 2: description of feature 2
    - feature 3: missing
- Bonuses delivered
    - Bonus 1: description of bonus 1
    - Bonus 2: missing
- Browser compatibility
    - IE11: tested and fully compatible

- Chrome v##.##.##: tested and partially compatible - feature x not working

- Opera v##.##.##: not tested or not functional

- External resources (Links and description of external resources such as JSON files, APIs, DBs, etc)

- License and contact information

- Authors: names, roles and team composition

- Changelog and version history

- Any other information that you think is important

note: "feature" means requirements or behaviors of this project as you have interpreted them from this document.

Screenshots

The attached screenshots are for reference only, develop your own style, look and feel

Your application should behave similarly to the screenshots provided