Hellscape

Game Design Document

ZentiNext

1. Table of Content

1. Table of Content **1**

2. Game Concept **2**

2.1. Game Overview 2

3. Game Design **3**

3.1. Visual Design 3

3.1.1. Exterior Design 3

3.1.2. Interior Design 3

3.2. Control Design 4

3.3. Implementation Design 5

3.3.1 Scene Graph/ Game Models 5

3.3.1.1 Wall 5

3.3.1.2 Handle 5

3.3.1.3 Bricks 6

3.3.1.4 Ball 6

3.3.1.5 Life Panel 6

3.3.2. Game Architecture Design / Folder Structure 7

3.3.2.1 index 7

3.3.2.2 Game Assets 7

3.3.2.3 Game Controllers 7

3.3.2.4 Event Bus 8

3.3.2.5 Game Models/ Scene Subjects 8

4. Interaction **9**

4.1. Single player 9

4.2. Two Player 9

5. Game-Flow **10**

5.1. Wireframe Diagram 10

5.2. Menus 11

6. Rules **12**

6.1. Single Player 12

6.2. Two Player 12

2. Game Concept

2.1. Game Overview

Hellscape is a new two dimensional game which is based on ‘Squash’. Squash is a ball sport played by two (singles) or four players (doubles squash) in a four-walled court with a small, hollow rubber ball. The players must alternate in striking the ball with their racquet and hit the ball onto the playable surfaces of the four walls of the court.

Hellscape is a new and fun way of playing squash in a two dimensional area. It is played by one or two players in a three walled area with a ball. In addition to the game squash, this has bricks in the play area which can be destroyed by striking at it with the ball and has handles at the bottom to be used as a racket. The players must alternate in striking the ball with their handle and hit the ball onto the destroyable bricks in the play area.

The goal of the game is to obtain the highest score by destroying the bricks with minimum hits. Game is based on scores and It does not have an explicit winning rather It shows when the player surpassed the current highest score.

3. Game Design

3.1. Visual Design

3.1.1. Exterior Design

The overall design of the game is cartoony and it is made to give user experience as an arcade game. It has a 2D top view in the theme of ‘Desert’ with customizable player colours, initially its red and blue.

3.1.2. Interior Design

The interior design consists of three main parts, the bottom scoreboard area, the play area in the middle and the handle area at the bottom. The scores and the lives are shown below the handles while handles(one or two) are shown at the bottom. In the play area there’s a ball designed as a skull and Bricks designed as cactus which are distributed in the play area to be used as striking targets.

3.1.2.1 Background Texture

Scene Background may have a sand which will give look and feel of a desert.

3.1.2.2 Brick Texture

Bricks which are destroyed when ball hits may look like cactus so that it stimulates realworl scenarios.

3.1.2.3 Wall Texture

Wall may look like sand blocks.

3.1.2.4 Ball Texture

Ball texture may designed with transparency so that player can get dynamic colour in the runtime.

3.1.2.5 Handle Texture

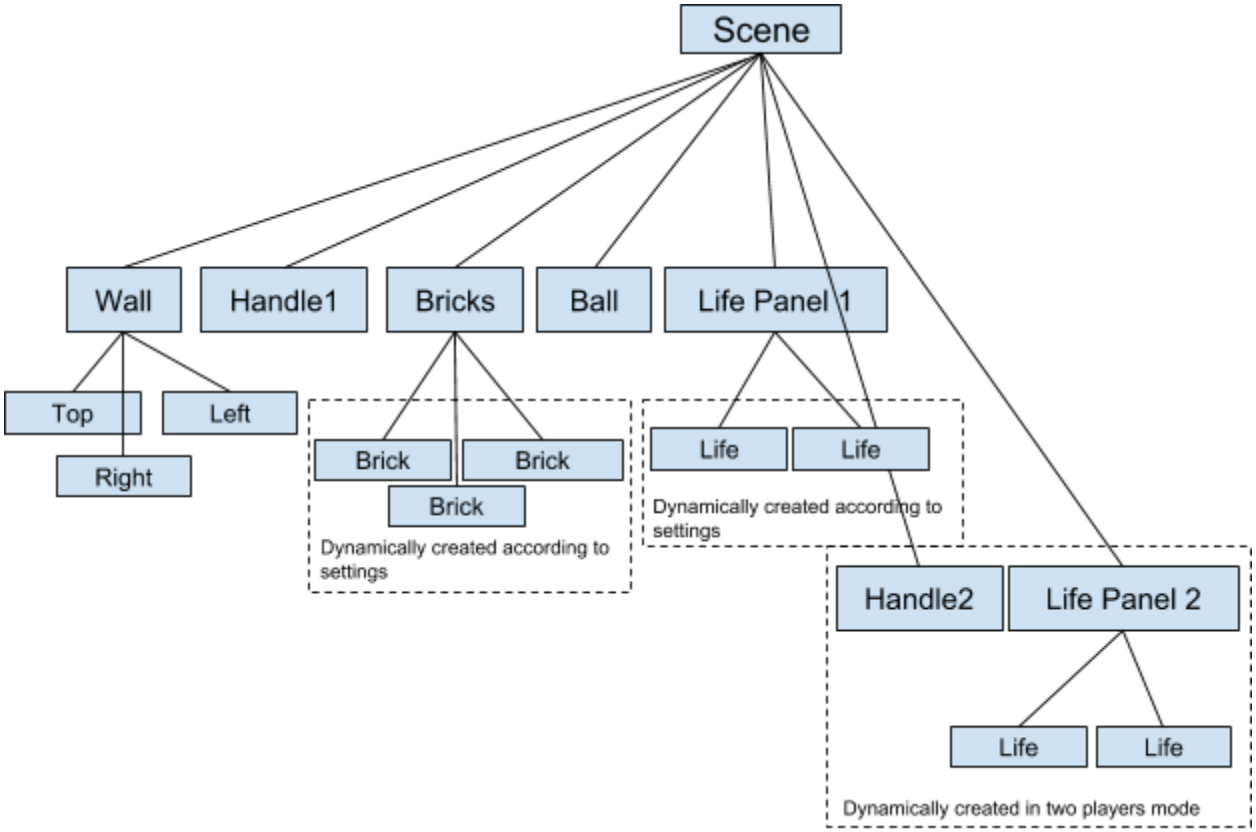
Player Handle also may designed with transparent area so that player can have customizable colour and change it in the runtime.

3.2. Control Design

Only keyboard is supported and Player handle is controlled by keyboard arrow keys. Two players on the same computer can be allowed by having controllers mapped to different keys. By default player1 controllers maped to left and right arrow keys. And for player2 “A” and “D” are mapped as left and right controllers. Players can change the controllers any given time.

3.3. Implementation Design

3.3.1 Scene Graph/ Game Models



3.3.1.1 Wall

Acts as an boundary of the game. It has three sides which stimulates a real world wall which bounce back if a ball hits it. Those three sides can be added as sub components to the wall.

3.3.1.2 Handle

Handle is the paddle/racket which is controlled by the player. Since this game has both one player and two-player modes, there can be up to two handles in this game. Handle1 is compulsory since we need at least one player playing this game. Handle2 will be created dynamically if the game mode is change to two players.

3.3.1.3 Bricks

Bricks are breakable components which is destroyed if a ball collide with it. Players will be able to score points if the ball owns by them collided with it. Those bricks generated and added to scene dynamically each time when player reset the game (play again). Players can dynamically change the number of bricks in the game by changing the settings.

3.3.1.4 Ball

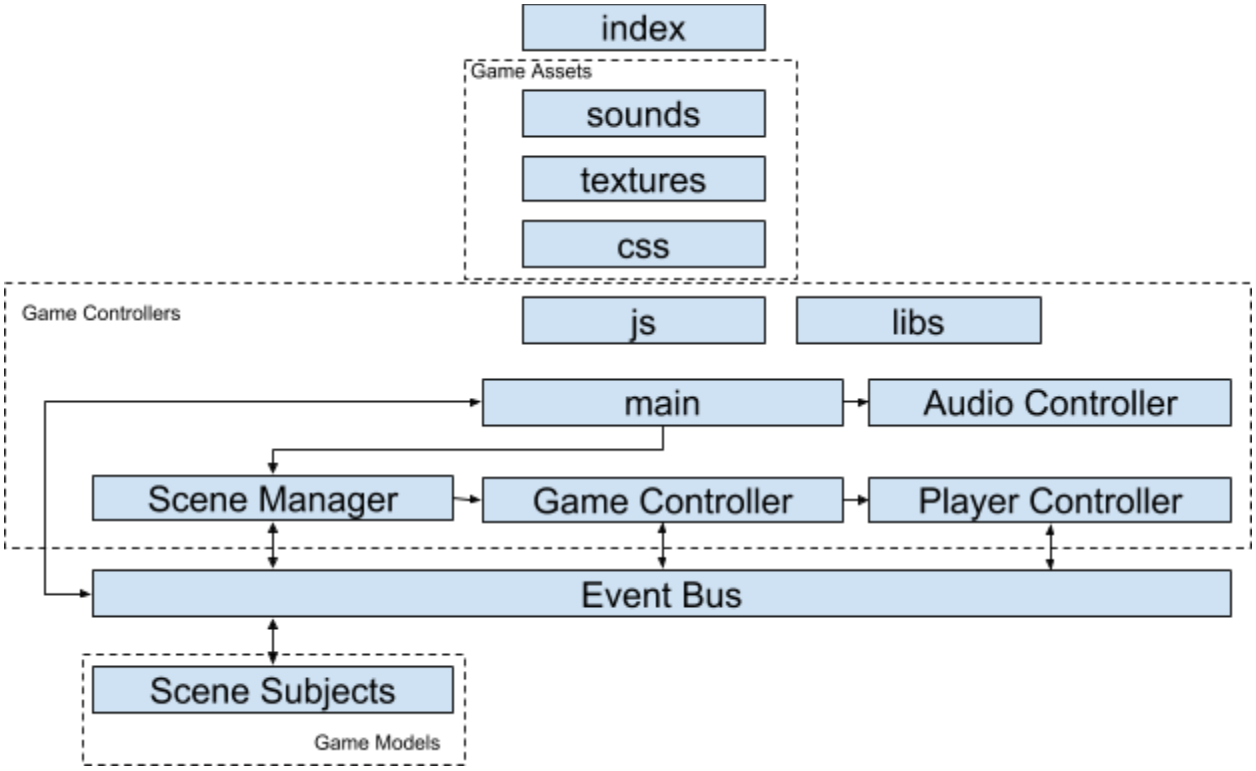
Ball is the main component of this game. It detects which components it collided with and control the flow of the game. It bounces over when hit a wall, brick or handle. Ball can be owned by either player 1 or player 2. By default it is owned by player 1 and ownership changes when it hits the owners handle in two player mode.

3.3.1.5 Life Panel

Life panel is a HUD (Head Up Display) which displays number of lives each player has. By default it has only player 1's Life panel and the LifePanel2 is created dynamically in two player mode. It changes it colour according to the player.

	Wall	**Brick**	**Handle**	**Ball**
Ball	**Collide**	**Damage & Collide**	**Collide**	**-**
Handle	**Collide**	**-**	**-**	
Brick	**-**	**-**		
Wall	**-**			

3.3.2. Game Architecture Design / Folder Structure



3.3.2.1 index

“index.html” file is at the root level and it is the entry point of the webpage and the game. It links all the css and javascript codes and contains the canvas which the game is rendered.

3.3.2.2 Game Assets

There are other 3 folders css, textures and sounds in the root level which are assets to the game. Css contains all the stylesheets and textures contains images which are used as textures in the game rendering. Sounds folder contains audio mp3 files which will be played in the game events.

3.3.2.3 Game Controllers

Other root level folder is js which contains all the controllers and models in the game.

“Main.js” file is the entry point to the javascript coding of the game. It can access DOM and contains SceneManager and AudioController. Main is responsible of creating SceneManager and passing canvas to it, attaching listeners to DOM events and starting the reder loop.

“SceneManager.js” is responsible for handling Three.js and it is independent from DOM. It creates scene, render and cameras. And also it creates all the models/scene subjects necessary for one player mode game and updates everything in each frame.

AudioController loads all the sound files and plays them when the game menu is opened, ball collided with another entity or when the ball is lost.

GameController controls the whole game. It contains number of bricks in the game and number of lives each player can have. It is responsible of creating the two players and managing the winning/losing conditions.

PlayerController controls the player movements by keeping keyboard mappings and keep track of their scores, number of bricks damages and number of lives lost.

3.3.2.4 Event Bus

Event bus is responsible of communication between each components in the game. It allows components to act independently without knowing each other.

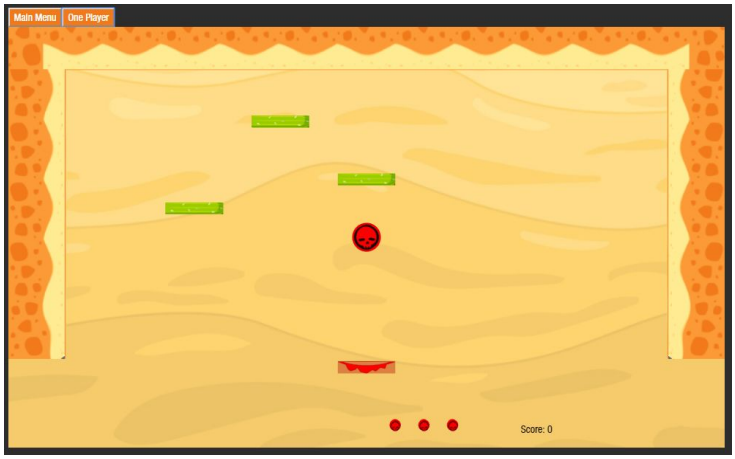
3.3.2.5 Game Models/ Scene Subjects

Game models/Scene Subjects are the entities/ components identified in the game. Scene Subjects were discussed in the previous section.

4. Interaction

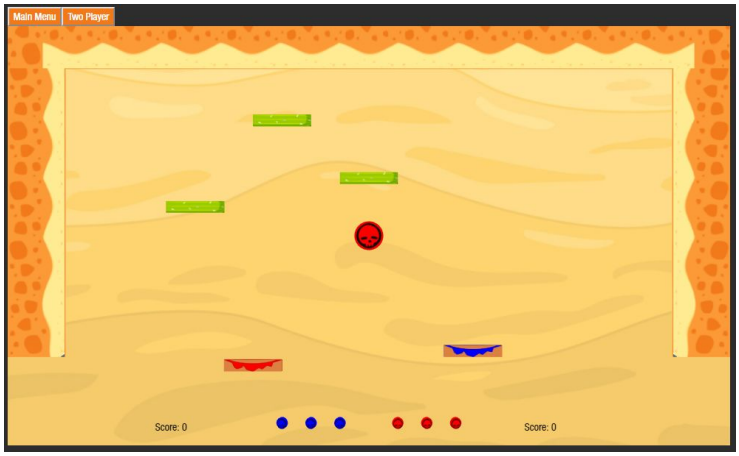
4.1. Single player

In single player game mode player can aim for current highest score. Player is playing against the previous record.



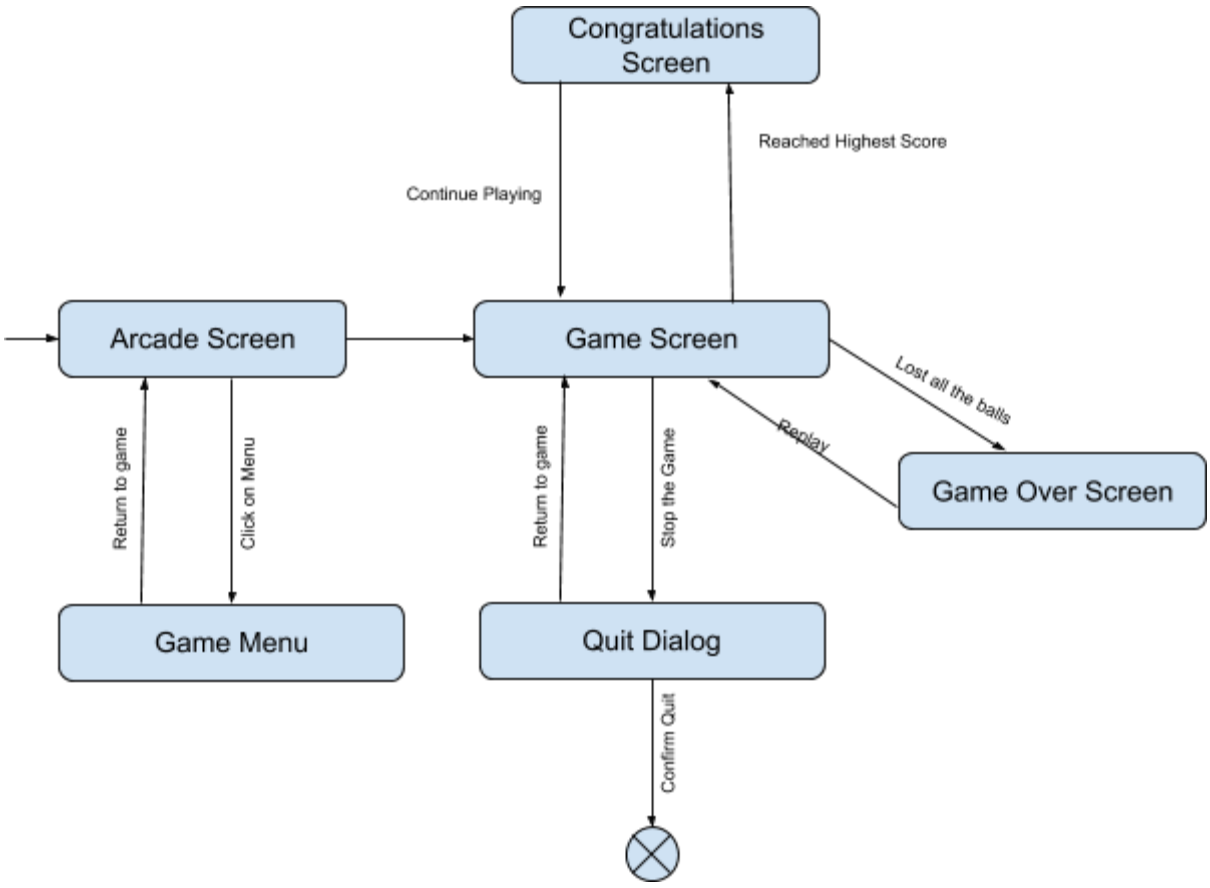
4.2. Two Player

In two player game mode, two players can play against each other aiming to score more than each other by destroying all the bricks.



5. Game-Flow

5.1. Wireframe Diagram



Game is rendered in the index page and all the menu buttons and player mode changing buttons are in the index page. Player can start the game by pressing enter. Player can change setting at any given time and if game settings are changed then the game will be restarted.

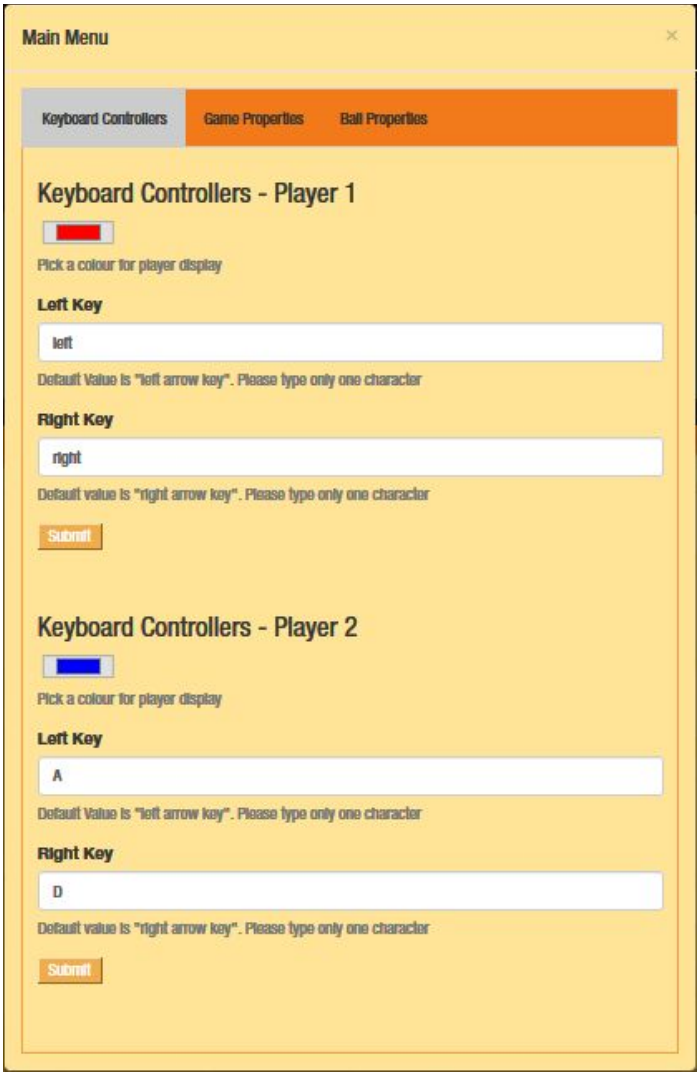
5.2. Menus

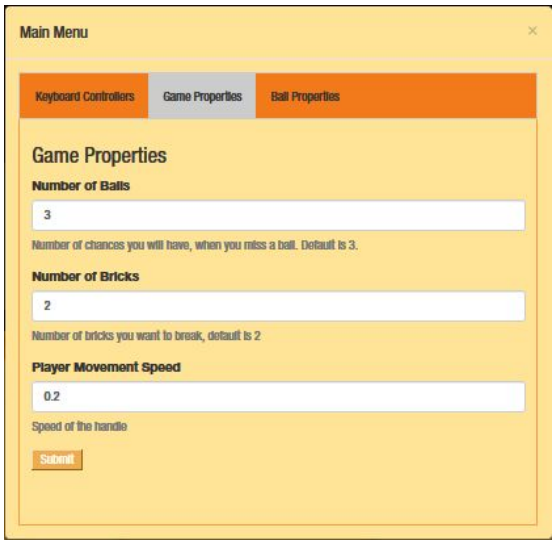
In the game menus there are three categories.

Keyboard controllers let users to map custom keyboard bindings to the player controller and also let player choose a colour for their handle, ball and lives.

Game controller contains number of bricks contains in the game, number of lives each player can have and the speed of the handle.

Ball Properties may contain the speed of the ball, force given when ball collided with other components and other relevant game physics attributes.





6. Rules

6.1. Single Player

- ❏ The handle is the only thing which can be controlled
- ❏ Marks can be added by striking at the bricks
- ❏ Dropping the ball by not catching it by the handle will lose a life
- ❏ To win the game the player has to destroy all the bricks in a given time using only three lives provided

6.2. Two Player

- ❏ The handle is the only thing which can be controlled
- ❏ Marks can be added by striking at bricks
- ❏ Dropping the ball by not catching it by the handle will lose a life
- ❏ Two players have to strike the ball one after the other
- ❏ Once the ball is hit by a handle, the striked player will get marks according to the bounces until the ball is hit by the opponent player
- ❏ To win the game a player has to score more than the opponent in a limited time