Escola Superior de Tecnologia e Gestão

Licenciatura em Engenharia Informática

Escola Superior de Tecnologia e Gestão

Instituto Politécnico de Viana do Castelo

Tecnologias Multimédia

2023/2024

Balloon Popper

João Lima Araújo Nº 24682

Índice

Lista de figuras

Introdução

Este relatório apresenta o desenvolvimento e implementação de um jogo 2D de estourar balões com Phaser 3, no âmbito da unidade curricular de Tecnologias Multimédia. O Balloon Popper é um jogo de ação e reflexos rápidos, onde os jogadores devem estourar balões de várias cores que sobem pelo ecrã, enquanto evitam bombas que podem reduzir suas vidas. O jogo começa com balões subindo lentamente, mas à medida que o jogador avança, a velocidade e a quantidade de balões aumentam, tornando o jogo mais desafiador.

Objetivo e instruções do jogo

O objetivo do Balloon Popper é simples: estourar o máximo de balões possível para ganhar pontos, enquanto se evita estourar as bombas. Os jogadores devem colocar o cursor em cima do balão e pressionar a tecla 'E' para estourá-los, e cada balão de cor diferente oferece uma quantidade específica de pontos, por exemplo, os balões azuis dão 2 pontos, os balões dão 3 pontos, os balões roxos dão 5 pontos e os balões brancos dão 10 pontos. Além disso, estourar balões consecutivos de diferentes cores pode formar combos, aumentando significativamente a pontuação do jogador.

Desenvolvimento do jogo

Criação dos balões

```
function createBlueBalloon(scene) {
    const x_blue = Phaser.Math.Between(50, 750);
    const blue_balloon = scene.balloons.create(x_blue, 600, 'blue_balloon');
    blue_balloon.setVelocityY(-100);
    blue_balloon.setInteractive();
    blue_balloon.setScale(0.2);
}

function createRedBalloon(scene) {
    const x_red = Phaser.Math.Between(50, 750);
    const red_balloon = scene.balloons.create(x_red, 600, 'red_balloon');
    red_balloon.setVelocityY(-200);
    red_balloon.setInteractive();
    red_balloon.setScale(0.15);
}

function createPurpleBalloon(scene) {
    const x_purple = Phaser.Math.Between(50, 750);
    const purple_balloon = scene.balloons.create(x_purple, 600, 'purple_balloon');
    purple_balloon.setVelocityY(-250);
    purple_balloon.setInteractive();
    purple_balloon.setScale(0.2);
}

function createWhiteBalloon(scene) {
    const x_white = Phaser.Math.Between(50, 750);
    const white_balloon = scene.balloons.create(x_white, 600, 'white_balloon');
    white_balloon.setVelocityY(-300);
    white_balloon.setInteractive();
    white_balloon.setScale(0.2);
```

Figura 1 - Criação de balões

Spawn dos balões

```
function setupBalloons() {
    this.time.addEvent({
        delay: 750,
        callback: function () {
            if (!gameOver && gameStarted) createBlueBalloon(this);
        },
        callbackScope: this,
        loop: true
    });

    this.time.addEvent({
        delay: 2000,
        callback: function () {
            if (!gameOver && gameStarted && this.timer > 10) {
                createRedBalloon(this);
            }
        },
        callbackScope: this,
        loop: true
    });

    this.time.addEvent({
        delay: 2500,
        callback: function () {
            if (!gameOver && gameStarted && this.timer > 20) {
                createPurpleBalloon(this);
            }
        },
        callbackScope: this,
        loop: true
    });

    this.time.addEvent({
        delay: 3500,
        callback: function () {
            if (!gameOver && gameStarted && this.timer > 30) {
                createWhiteBalloon(this);
                createRedBalloon(this);
            }
        },
        callbackScope: this,
        loop: true
    });
}
```

Figura 2 - Spawn de balões

Os balões “spawnam” no ecrã em intervalos de tempo regulares e são controlados por variáveis booleanas, por exemplo o balão azul aparece no ecrã a cada 750ms. Quanto mais aumenta o tempo de jogo, mais balões vão aparecer no ecrã.

Adicionalmente, vão também aparecer bombas no ecrã a cada 5000ms.

```
function setupBomb() {
    this.time.addEvent({
        delay: 5000,
        callback: function () {
            if (!gameOver && gameStarted) createBomb(this);
        },
        callbackScope: this,
        loop: true
    });
}
```

Figura 3 - Spawn da bomba

Estourar balões

```
function setupKeyboard() {
    this.eKey = this.input.keyboard.addKey(Phaser.Input.Keyboard.KeyCodes.E);
    this.rKey = this.input.keyboard.addKey(Phaser.Input.Keyboard.KeyCodes.R);

    this.input.keyboard.on('keydown-E', function (event) {
        if (gameOver) return; // Não faz nada se o jogo terminou
        const pointer = this.input.activePointer;
        this.balloons.children.iterate(function (balloon) {
            if (balloon && balloon.getBounds().contains(pointer.x, pointer.y)) {
                showExplosion.call(this, balloon.x, balloon.y); // Adiciona a explosão
                if (balloon.texture.key === 'blue_balloon') {
                    balloon.destroy();
                    popSound.play();
                    updateScore.call(this, 2);
                    popBalloon(this, 'blue_balloon');
                } else if (balloon.texture.key === 'red_balloon') {
                    balloon.destroy();
                    popSound.play();
                    updateScore.call(this, 3);
                    popBalloon(this, 'red_balloon');
                } else if (balloon.texture.key === 'purple_balloon') {
                    balloon.destroy();
                    popSound.play();
                    updateScore.call(this, 5);
                    popBalloon(this, 'purple_balloon');
                } else if (balloon.texture.key === 'white_balloon') {
                    balloon.destroy();
                    popSound.play();
                    updateScore.call(this, 10);
                    popBalloon(this, 'white_balloon');
                } else if (balloon.texture.key === 'bomb') {
                    balloon.destroy();
                    bombSound.play();
                    loseLife.call(this);
                }
            }
        }, this);
    }, this);
    var scene = this;
    this.rKey = this.input.keyboard.addKey(Phaser.Input.Keyboard.KeyCodes.R);

    // Reiniciar jogo
    this.rKey.on('down', function () {
        this.scene.restart();
    }, this);
}
```

Figura 4 - Função para estourar balões

Esta função é responsável por estourar os balões, ela verifica se a tecla 'E' está a ser pressionada e se o cursor está em cima do balão, caso esteja a função vai verificar a cor do balão e conforme a cor vai atribuir a pontuação. Sempre que estourar uma bomba perde-se uma vida. Para reiniciar o jogo basta clicar no 'R'.

Guardar pontuação

```
function saveScore(score) {
    let scores = JSON.parse(localStorage.getItem('topScores')) || [];
    scores.push(score);
    scores.sort((a, b) => b - a);
    scores = scores.slice(0, 10); // Mantém apenas os top 10 scores
    localStorage.setItem('topScores', JSON.stringify(scores));
}
```

Figura 5 - Guardar pontuação

A função saveScore armazena a pontuação do jogador no localStorage do browser, organizando as pontuações em ordem decrescente e mantendo apenas as 10 melhores.

Função update

```
function update() {
    // Verifica se algum balão saiu do ecrã
    this.balloons.children.iterate(function (balloon) {
        if (balloon && balloon.y < 0) {
            if (balloon.texture.key !== 'bomb') {
                loseLife.call(this);
                loseSound.play();
            }
            balloon.destroy();
        }
    }, this);
}
```

Figura 6 - Função update

A função update verifica se algum balão saiu do ecrã iterando sobre todos os balões no grupo balloons. Se um balão estiver fora da tela e não for uma bomba, é chamada a função loseLife e o som loseSound é reproduzido, indicando a perda de uma vida. Em seguida, o balão é destruído, garantindo que ele seja removido do jogo.

User Interface

Menu principal

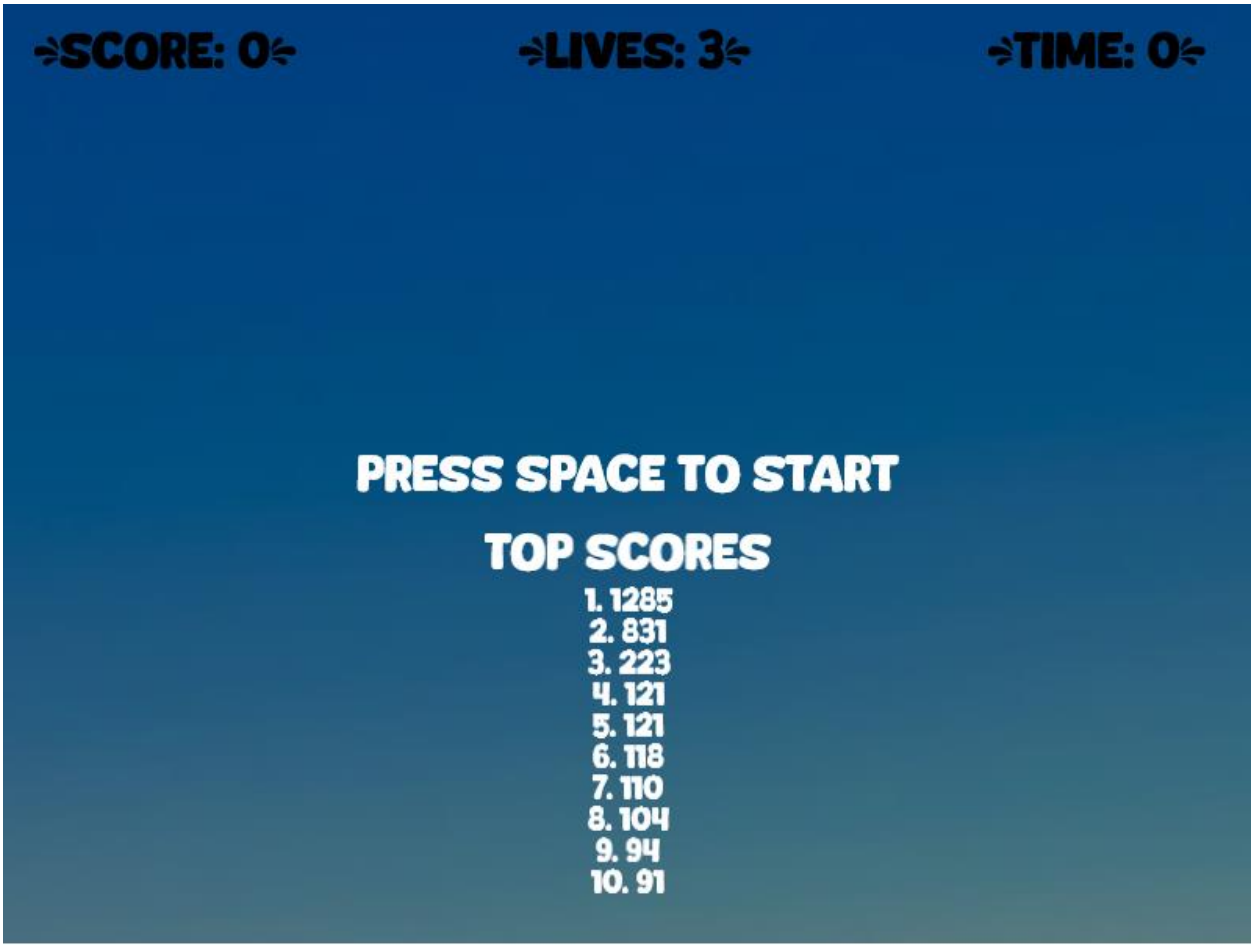

Figura 7 - Menu principal

Neste ecrã é possível observar o top 10 pontuações. Para iniciar o jogo basta pressionar na tecla espaço.

Explosão do balão

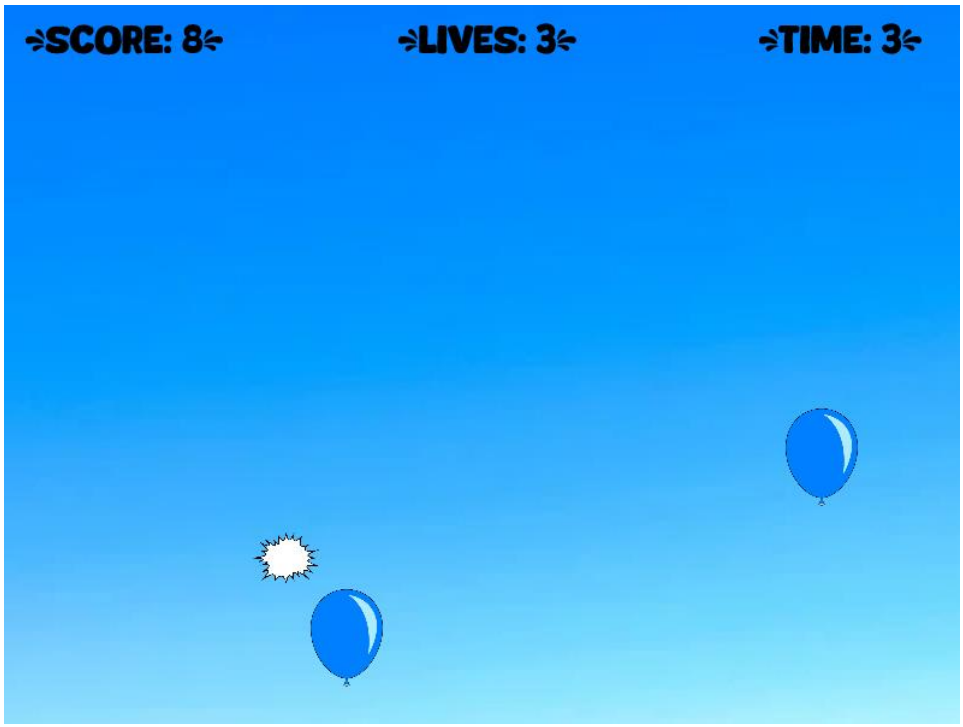

Figura 8 - Explosão do balão

Combo

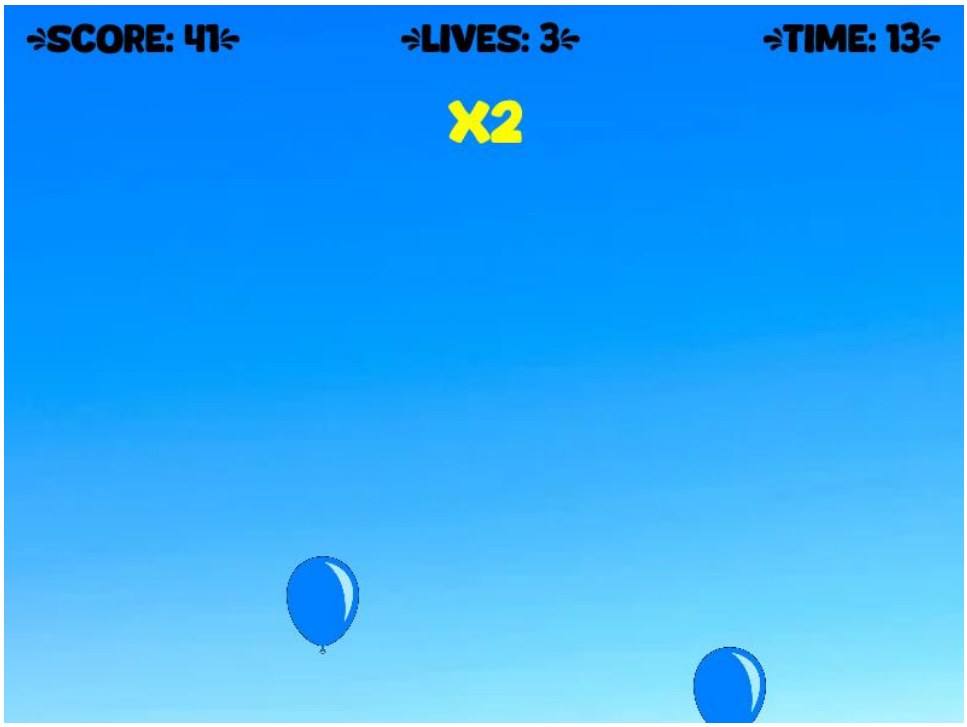

Figura 9 – Combo

Explosão da bomba

Figura 10 - Explosão da bomba

Game over

Figura 11 - Game over

Conclusão

O desenvolvimento do jogo Balloon Popper foi interessante. O objetivo do jogo é estourar balões de diferentes cores, evitando bombas, para acumular a maior pontuação possível. A implementação do sistema de combos foi particularmente difícil, exigindo uma lógica complexa para identificar e registar uma sequência de cores diferentes. Adicionalmente, a persistência dos scores no localStorage apresentou desafios ao ordenar as pontuações e limitar aos 10 melhores scores, garantindo a competitividade do jogo. Apesar das dificuldades, o projeto proporcionou uma aplicação prática de conceitos teóricos aprendidos sobre phaser3. Em suma, Balloon Popper reforçou as minhas competências técnicas e criativas.