

LABORATORI MULTIMÈDIA

# PRÀCTICA 2

## ATARI BREAKOUT

Pau Serra  
Liuqihao Qiu  
Laura Martínez

Grup P200-G103-4

30/05/2024

# ÍNDEX

SETMANA 1.....	2
SETMANA 2.....	4
SETMANA 3.....	10
SETMANA 4.....	17
SETMANA 5.....	25

## SETMANA 1

### IMPLICACIÓ DE CADA INTEGRANT

Noms dels integrants del grup	Valoració
Liquihao Qiu	0
Laura Martínez	1
Pau Serra	1

### TASQUES REALITZADES

Número de tasca	Descripció de la tasca
Tasca 1	Inspecció del codi

## EVIDÈNCIES

### Tasca 1: Inspecció del codi

No tenim cap evidència ja que l'únic que hem realitzat des de la sessió de presentació de la pràctica ha estat una mica d'inspecció del codi per mirar d'entendre bé l'estruccuració del mateix i el funcionament de les diferents classes/objectes així com les funcions i mètodes de cada una.

## SETMANA 2

### IMPLICACIÓ DE CADA INTEGRANT

Noms dels integrants del grup	Valoració
Liquihao Qiu	0
Laura Martínez	2
Pau Serra	2

### TASQUES REALITZADES

Número de tasca	Descripció de la tasca
Tasca 1	Implementació del moviment de la pala
Tasca 2	Implementació dels rebots de la pilota amb les parets
Tasca 3	Generació d'array de totxos
Tasca 4	Generació del mur
Tasca 5	Selecció de nivell (per codi)

## EVIDÈNCIES

Tasca 1: Implementació del moviment de la pala

```
const joc = this;
$(document).on("keydown", function(e){
    if (e.which === joc.key.LEFT.code) {
        console.log("Left arrow key pressed");
        joc.pala.mou(-5, 0);
    } else if (e.which === joc.key.RIGHT.code) {
        console.log("Right arrow key pressed");
        joc.pala.mou(5, 0);
    }
});
$(document).on("keyup", function(e){
    // Handle keyup event if needed
});
```



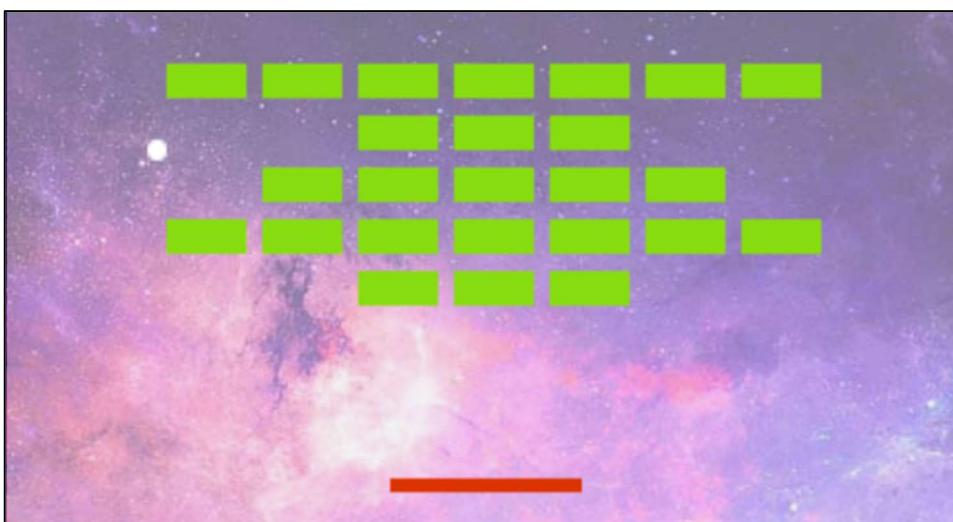
## Tasca 2: Implementació dels rebots de la pilota amb les parets

```

//Xoc lateral superior
if(traectoria.puntB.y - this.radi < 0){
    exces = (traectoria.puntB.y - this.radi)/this.vy;
    this.posicio.x = traectoria.puntB.x - exces*this.vx;
    this.posicio.y = this.radi;
    xoc = true;
    this.vy = -this.vy;
}
//Xoc lateral esquerra
if(traectoria.puntB.x - this.radi < 0){
    exces = (traectoria.puntB.x - this.radi)/this.vx;
    this.posicio.x = this.radi;
    this.posicio.y = traectoria.puntB.y - exces*this.vy;
    xoc = true;
    this.vx = -this.vx;
}
//Xoc lateral dret
if(traectoria.puntB.x + this.radi > joc.amplada){
    exces = (traectoria.puntB.x + this.radi - joc.amplada)/this.vx;
    this.posicio.x = joc.amplada - this.radi;
    this.posicio.y = traectoria.puntB.y - exces*this.vy;
    xoc = true;
    this.vx = -this.vx;
}
//Xoc lateral inferior
if(traectoria.puntB.y + this.radi > joc.alcada){
    exces = (traectoria.puntB.y + this.radi - joc.alcada)/this.vy;
    this.posicio.x = traectoria.puntB.x - exces*this.vx;
    this.posicio.y = joc.alcada - this.radi;
    xoc = true;
    this.vy = -this.vy;

    //alert("Has perdit");
}

```



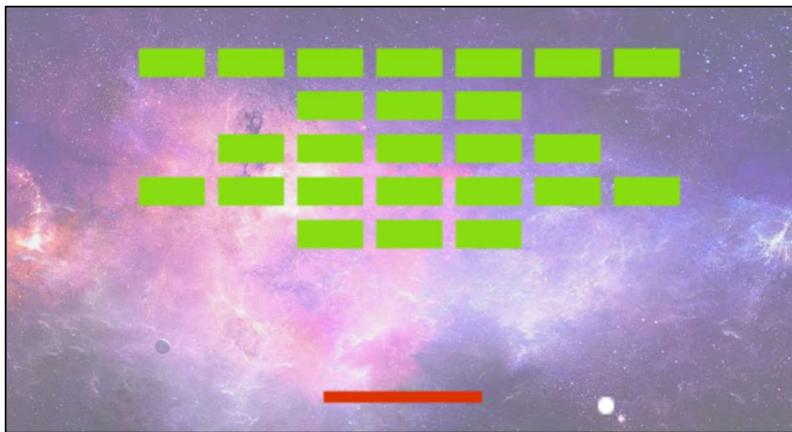
### Tasca 3: Generació d'array de totxos

```
generaMur(){
    var margin = 15;
    var x = 5;
    var y = 15;
    var totxoTemp;
    for(var i = 0; i < this.nivells[this.dificultat].totxos.length; i++) {
        for(var j = 0; j < this.nivells[this.dificultat].totxos[i].length; j++) {
            if(this.nivells[this.dificultat].totxos[i].charAt(j) === 'a') {
                totxoTemp = new Totxo(new Punt(x + margin, y), 25, 10, this.nivells[this.dificultat].color);
                this.arrayTotxos.push(totxoTemp);
            }
            x = x + 15 + margin;
        }
        y = y + margin;
        x = 5;
    }
    console.log(this.arrayTotxos);
    console.log(y);
}
```

```
▼ Array(25) [ ]
▶ 0: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 1: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 2: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 3: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 4: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 5: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 6: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 7: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 8: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 9: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 10: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 11: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 12: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 13: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 14: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 15: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 16: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 17: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 18: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 19: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 20: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 21: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 22: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 23: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ 24: Totxo {amplada: 25, alcada: 10, tocat: false, posicio: Punt, color: '#8D1'}
▶ length: 25
▶ [[Prototype]]: Array(0)
```

**Tasca 4:** Generació del mur

```
draw(ctx){  
    this.arrayTotxos.forEach(element => {  
        element.draw(ctx);  
    });  
}
```

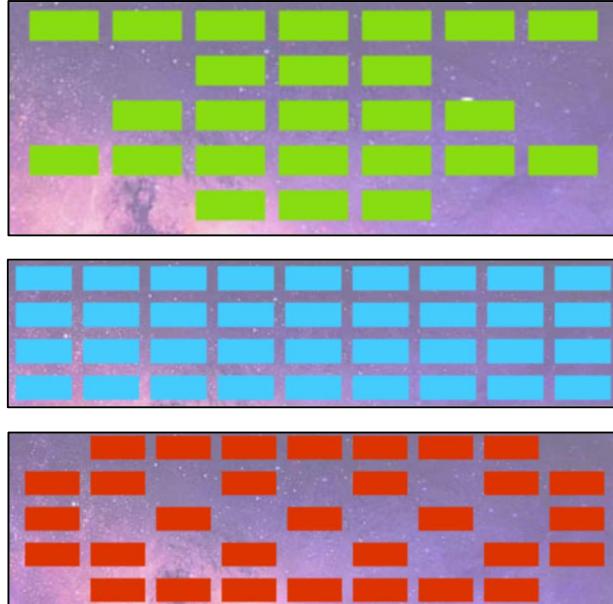


### Tasca 5: Selecció de nivell (per codi)

```
class Mur {
    constructor(dificultat, arrayTotxos) {
        this.dificultat = dificultat;
        this.arrayTotxos = arrayTotxos;
    }

    generaMur(){
        var margin = 15;
        var x = 5;
        var y = 15;
        var totxoTemp;
        for(var i = 0; i < this.nivells[this.dificultat].totxos.length; i++) {
            for(var j = 0; j < this.nivells[this.dificultat].totxos[i].length; j++) {
                if(this.nivells[this.dificultat].totxos[i].charAt(j) === 'a') {
                    totxoTemp = new Totxo(new Punt(x + margin, y), 25, 10, this.nivells[this.dificultat].color);
                    this.arrayTotxos.push(totxoTemp);
                }
                x = x + 15 + margin;
            }
            y = y + margin;
            x = 5;
        }
        console.log(this.arrayTotxos);
        console.log(y);
    }
}
```

```
defineixNivells(){
    this.nivells=[
        {
            color: "#4CF", // blau cel
            totxos:[
                "aaaaaaaaaa",
                "aaaaaaaaaa",
                "aaaaaaaaaa",
                "aaaaaaaaaa",
            ]
        },
        {
            color: "#8D1", // verd
            totxos:[
                " aaaaaaa ",
                "   aaa   ",
                "  aaaa  ",
                " aaaaaaa ",
                "   aaa   ",
            ]
        },
        {
            color: "#D30", // vermell
            totxos:[
                " aaaaaaa",
                "aa a a aa",
                "a a a a a",
                "aa a a aa",
                " aaaaaaa ",
            ]
        }
    ];
}
```



## SETMANA 3

### IMPLICACIÓ DE CADA INTEGRANT

Noms dels integrants del grup	Valoració
Liquihao Qiu	1
Laura Martínez	2
Pau Serra	2

### TASQUES REALITZADES

Número de tasca	Descripció de la tasca
Tasca 1	Correcció del moviment de la pala
Tasca 2	Implementació del rebot de la pilota amb els totxos
Tasca 3	Implementació del rebot de la pilota amb la pala
Tasca 4	Desaparició dels totxos en contactar amb la pilota
Tasca 5	Addició d'un temporitzador

## EVIDÈNCIES

### Tasca 1: Correcció del moviment de la pala

Moviment de la pala més fluid i control del moviment perquè no surti dels marges del canvas.

```

class Pala {
    constructor(puntPosicio, amplada, alcada){
        this.amplada = amplada;
        this.alcada = alcada;
        this.posicio = puntPosicio;
        this.velocitat = 2;
        this.vx = 0;           // Initialize vx to 0
        this.color = "#D30";
    }

    update() {
        this.mou(); // Call the movement method in the update method
    }

    draw(ctx) {
        ctx.save();
        ctx.fillStyle = this.color;
        ctx.fillRect(this.posicio.x, this.posicio.y, this.amplada, this.alcada);
        ctx.restore();
    }

    mou() {
        if(this.posicio.x + this.vx > 0 && this.posicio.x + this.vx < joc.amplada - this.amplada) {
            this.posicio.x += this.vx; // Update position with vx
        }
    }
}

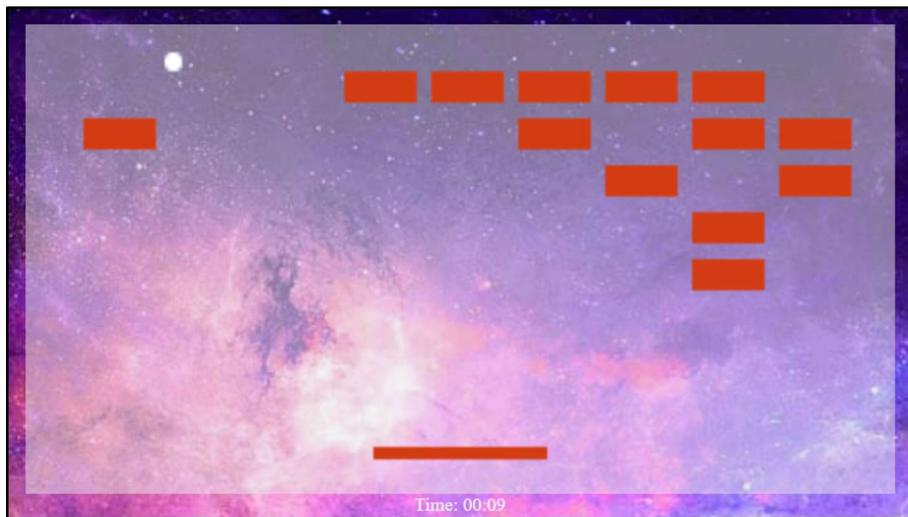
var key = {
    LEFT: { code: 37, pressed: false },
    RIGHT: { code: 39, pressed: false }
};

$(document).on("keydown", function(e){
    if (e.which === key.LEFT.code) {
        console.log("Left arrow key pressed");
        key.LEFT.pressed = true;
        key.RIGHT.pressed = false;
        joc.pala.vx = -joc.pala.velocitat; // Update vx to move left
    } else if (e.which === key.RIGHT.code) {
        console.log("Right arrow key pressed");
        key.RIGHT.pressed = true;
        key.LEFT.pressed = false;
        joc.pala.vx = joc.pala.velocitat; // Update vx to move right
    }
});

$(document).on("keyup", function(e){
    if (e.which === key.LEFT.code && key.LEFT.pressed) {
        console.log("Left arrow key released");
        key.LEFT.pressed = false;
        if (key.RIGHT.pressed) {
            joc.pala.vx = joc.pala.velocitat;
        } else {
            joc.pala.vx = 0;
        }
    } else if (e.which === key.RIGHT.code && key.RIGHT.pressed) {
        console.log("Right arrow key released");
        key.RIGHT.pressed = false;
        if (key.LEFT.pressed) {
            joc.pala.vx = -joc.pala.velocitat;
        } else {
            joc.pala.vx = 0;
        }
    }
});

```

## Tasca 2: Implementació del rebot de la pilota amb els totxos



```

for (let i = 0; i < arrayTotxos.length; i++) {
    let totxo = arrayTotxos[i];
    var objInterseccioTotxo = this.interseccioSegmentRectangle(trajectoria, totxo);

    if (objInterseccioTotxo != undefined) {
        switch (objInterseccioTotxo.vora) {
            case 'superior':
                this.posicio.x = objInterseccioTotxo.pI.x;
                this.posicio.y = objInterseccioTotxo.pI.y - this.radi;
                xoc = true;
                this.vy = -this.vy;
                break;
            case 'inferior':
                this.posicio.x = objInterseccioTotxo.pI.x;
                this.posicio.y = objInterseccioTotxo.pI.y + this.radi;
                xoc = true;
                this.vy = -this.vy;
                break;
            case 'esquerra':
                this.posicio.x = objInterseccioTotxo.pI.x - this.radi;
                this.posicio.y = objInterseccioTotxo.pI.y;
                xoc = true;
                this.vx = -this.vx;
                break;
            case 'dreta':
                this.posicio.x = objInterseccioTotxo.pI.x + this.radi;
                this.posicio.y = objInterseccioTotxo.pI.y;
                xoc = true;
                this.vx = -this.vx;
                break;
        }
        arrayTotxos.splice(i, 1);
        i--;
    }
}

```

```

interseccioSegmentRectangle(segment, rectangle){

    //1r REVISAR SI EXISTEIX UN PUNT D'INTERSECCIÓ EN UN DELS 4 SEGMENTS
    //SI EXISTEIX, QUIN ÉS AQUEST PUNT
    //si hi ha més d'un, el més ajustat
    let puntI;
    let distanciaI;
    let puntIMin;
    let distanciaIMin = Infinity;
    let voraI;

    //calcular punt d'intersecció amb les 4 vores del rectangle
    //necessitem coneixer els 4 segments del rectangle
    let puntSuperiorEsquerra = new Punt(rectangle.posicio.x - this.radi, rectangle.posicio.y - this.radi);
    let puntSuperiorDreta = new Punt(rectangle.posicio.x + rectangle.amplada + this.radi, rectangle.posicio.y - this.radi);
    let puntInferiorEsquerra = new Punt(rectangle.posicio.x - this.radi, rectangle.posicio.y + rectangle.alçada + this.radi);
    let puntInferiorDreta = new Punt(rectangle.posicio.x + rectangle.amplada + this.radi, rectangle.posicio.y + rectangle.alçada + this.radi);

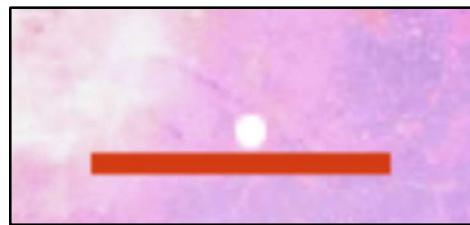
    let segmentVoraSuperior = new Segment(puntSuperiorEsquerra, puntSuperiorDreta);
    let segmentVoraInferior = new Segment(puntInferiorEsquerra, puntInferiorDreta);
    let segmentVoraEsquerra = new Segment(puntSuperiorEsquerra, puntInferiorEsquerra);
    let segmentVoraDreta = new Segment(puntSuperiorDreta, puntInferiorDreta);
    //2n REVISAR SI EXISTEIX UN PUNT D'INTERSECCIÓ EN UN DELS 4 SEGMENTS
    //SI EXISTEIX, QUIN ÉS AQUEST PUNT
    //si hi ha més d'n, el més ajustat

    //vora superior
    puntI = segment.puntInterseccio(segmentVoraSuperior);
    if (puntI) {
        //distància entre dos punts, el punt inicial del segment i el punt d'intersecció
        distanciaI = Punt.distanciaDosPunts(segment.puntA, puntI);
        if (distanciaI < distanciaIMin) {
            distanciaIMin = distanciaI;
            puntIMin = puntI;
            voraI = "superior";
        }
    }
    //vora inferior
    puntI = segment.puntInterseccio(segmentVoraInferior);
    if(puntI) {
        distanciaI = Punt.distanciaDosPunts(segment.puntA, puntI);
        if(distanciaI < distanciaIMin) {
            distanciaIMin = distanciaI;
            puntIMin = puntI;
            voraI = "inferior";
        }
    }
    //vora esquerra
    puntI = segment.puntInterseccio(segmentVoraEsquerra);
    if(puntI) {
        distanciaI = Punt.distanciaDosPunts(segment.puntA, puntI);
        if(distanciaI < distanciaIMin) {
            distanciaIMin = distanciaI;
            puntIMin = puntI;
            voraI = "esquerra";
        }
    }
    //vora dreta
    puntI = segment.puntInterseccio(segmentVoraDreta);
    if(puntI) {
        distanciaI = Punt.distanciaDosPunts(segment.puntA, puntI);
        if(distanciaI < distanciaIMin) {
            distanciaIMin = distanciaI;
            puntIMin = puntI;
            voraI = "dreta";
        }
    }
}

//Retorna la vora on s'ha produït la col·lisió, i el punt (x,y)
if(voraI){
    return {pI: puntIMin, vora: voraI};
}

distancia = function(p1,p2){
    return Math.sqrt((p2.x-p1.x)*(p2.x-p1.x)+(p2.y-p1.y)*(p2.y-p1.y));
}

```

**Tasca 3:** Implementació del rebot de la pilota amb la pala

```
//Xoc amb la pala
var objInterseccioPala = this.interseccioSegmentRectangle(trajectoria, this.pala);

if (objInterseccioPala != undefined) {
    switch (objInterseccioPala.vora) {
        case 'superior':
            this.posicio.x = objInterseccioPala.pI.x;
            this.posicio.y = objInterseccioPala.pI.y;
            xoc = true;
            this.vy = -this.vy;
            break;
        case 'inferior':
            break;
        case 'esquerra':
            this.posicio.x = objInterseccioPala.pI.x - this.radi;
            this.posicio.y = objInterseccioPala.pI.y;
            xoc = true;
            this.vx = -this.vx;
            break;
        case 'dreta':
            this.posicio.x = objInterseccioPala.pI.x + this.radi;
            this.posicio.y = objInterseccioPala.pI.y;
            xoc = true;
            this.vx = -this.vx;
            break;
    }
}
```

#### Tasca 4: Desaparició dels totxos en contactar amb la pilota



```

for (let i = 0; i < arrayTotxos.length; i++) {
    let totxo = arrayTotxos[i];
    var objInterseccioTotxo = this.interseccioSegmentRectangle(trajectoria, totxo);

    if (objInterseccioTotxo != undefined) {
        switch (objInterseccioTotxo.vora) {
            case 'superior':
                this.posicio.x = objInterseccioTotxo.pI.x;
                this.posicio.y = objInterseccioTotxo.pI.y - this.radi;
                xoc = true;
                this.vy = -this.vy;
                break;
            case 'inferior':
                this.posicio.x = objInterseccioTotxo.pI.x;
                this.posicio.y = objInterseccioTotxo.pI.y + this.radi;
                xoc = true;
                this.vy = -this.vy;
                break;
            case 'esquerra':
                this.posicio.x = objInterseccioTotxo.pI.x - this.radi;
                this.posicio.y = objInterseccioTotxo.pI.y;
                xoc = true;
                this.vx = -this.vx;
                break;
            case 'dreta':
                this.posicio.x = objInterseccioTotxo.pI.x + this.radi;
                this.posicio.y = objInterseccioTotxo.pI.y;
                xoc = true;
                this.vx = -this.vx;
                break;
        }
        arrayTotxos.splice(i, 1);
        i--;
    }
}

```

**Tasca 5:** Addició d'un temporitzador

Time: 00:22

```
startTimer() {
    this.startTime = Date.now();
    const joc = this;
    this.timerInterval = setInterval(function () {
        joc.updateTimer();
    }, 1000);
}

updateTimer() {
    this.elapsedTime = Math.floor((Date.now() - this.startTime) / 1000);
    const minutes = Math.floor(this.elapsedTime / 60);
    const seconds = this.elapsedTime % 60;
    $('#timer').text('Time: ' + this.pad(minutes) + ':' + this.pad(seconds));
}

pad(number) {
    return number < 10 ? '0' + number : number;
}

stopTimer() {
    clearInterval(this.timerInterval);
}
```

## SETMANA 4

### IMPLICACIÓ DE CADA INTEGRANT

Noms dels integrants del grup	Valoració
Liquihao Qiu	1
Laura Martínez	2
Pau Serra	2

### TASQUES REALITZADES

Número de tasca	Descripció de la tasca
Tasca 1	Implementació del menú
Tasca 2	Canvi de fons aleatori
Tasca 3	Velocitat de la pala progressiva
Tasca 4	Velocitat de la bola progressiva
Tasca 5	Comptador d'inici
Tasca 6	Implementació del sistema de vides (en procés)

## EVIDÈNCIES

### Tasca 1: Implementació del menú



```
<div id="menu">
  <div class="titles_menu">
    <h1>ATARI BREAKOUT</h1>
    <br>
    <h3>There are 3 different types of games! Choose one</h3>
  </div>
  <div class="difficulty">
    <button onclick="difficulty(0, event)">EASY</button>
    <button onclick="difficulty(1, event)">MEDIUM</button>
    <button onclick="difficulty(2, event)">HARD</button>
  </div>
  <button class="play" onclick="startGame()" id="play">PLAY</button>
  <div id="error_difficulty" style="display: none; color: red; font-size: 17px;">Please select a difficulty</div>
</div>
```

```
function startGame() {
  if(checkButtons()) {
    //("entra a startGame()");
    var level = nivell;
    $('#menu').hide();
    $('#principal').show();
    beginGame();
    countdown(() => {
      joc = new Joc(myCanvas, ctx, level);
      joc.inicialitza();
      animacio();
    });
  }
}
```

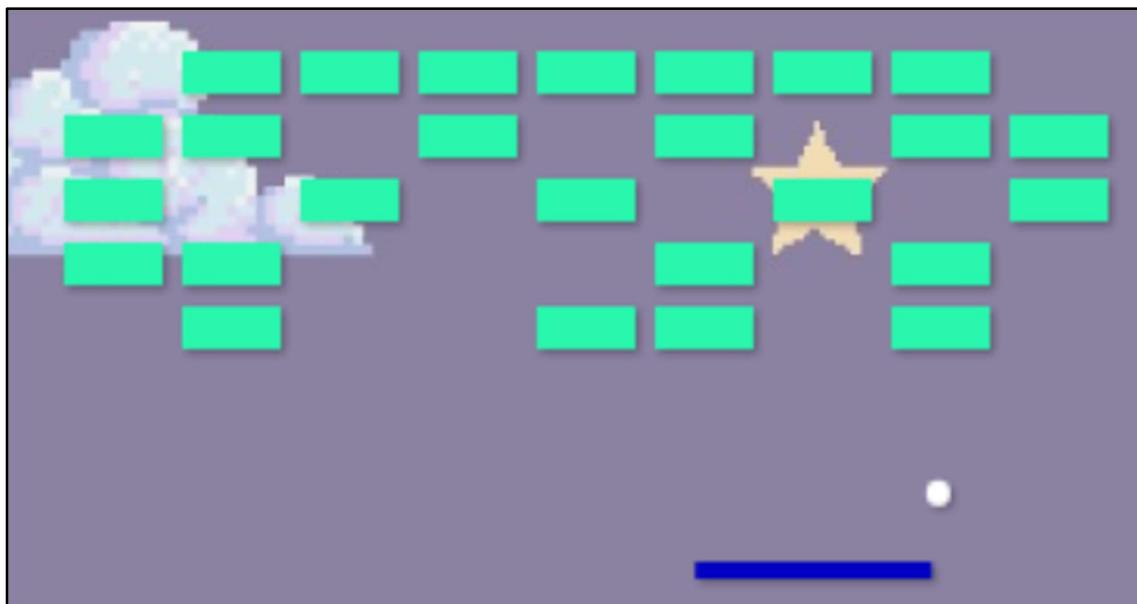
**Tasca 2:** Canvi de fons aleatori



```
function beginGame() {
    let backgrounds = ["images/fons1.jpg", "images/fons2.jpg", "images/fons3.jpg"];
    let randomBackground = backgrounds[Math.floor(Math.random() * 3)];

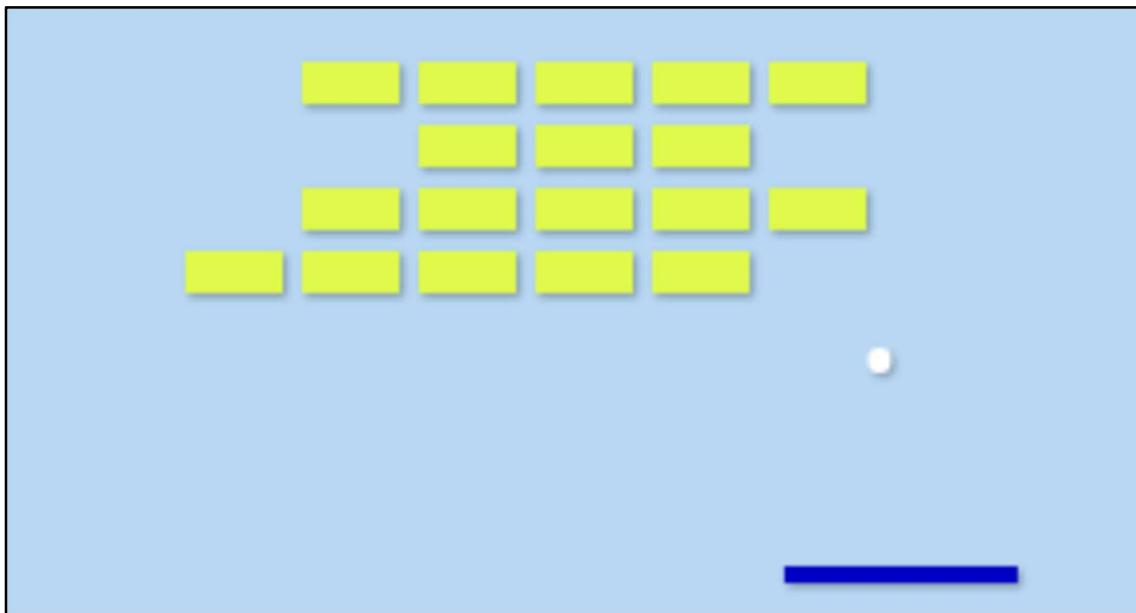
    $("body").css({
        "background": "url('" + randomBackground + "') no-repeat center center fixed",
        "background-size": "cover",
    });
}
```

**Tasca 3:** Velocitat de la pala progressiva



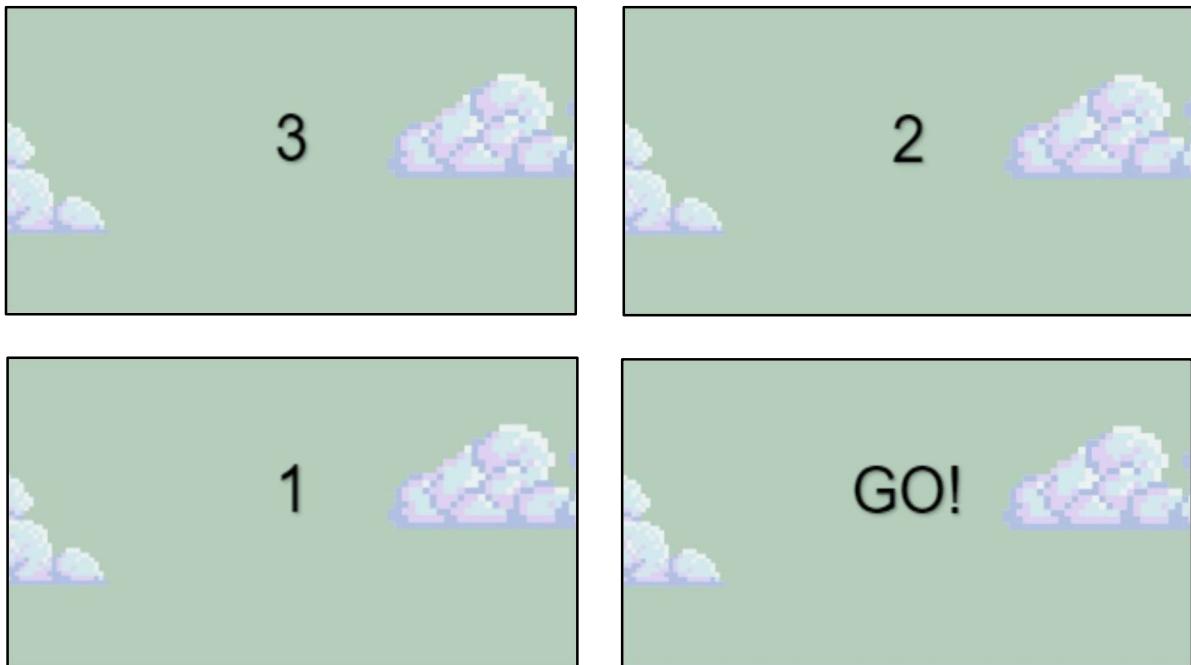
```
update() {  
    this.mou();  
    if(this.velocitat <= 2) {  
        this.velocitat = this.velocitat + 0.0001;  
    }  
}
```

**Tasca 4:** Velocitat de la bola progressiva



```
if (this.v >= 1) {  
    this.v = this.v - 0.0002;  
}
```

### Tasca 5: Comptador d'inici



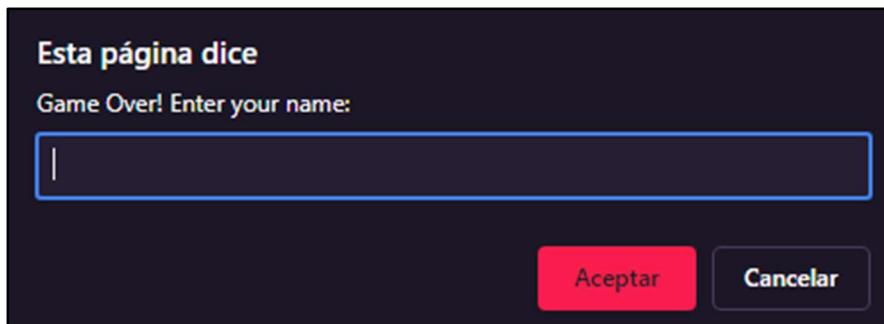
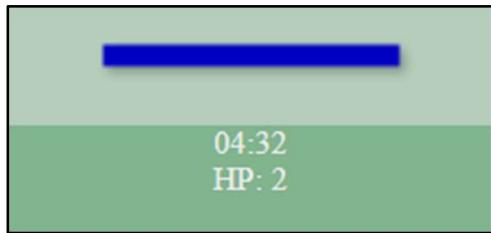
```
function countdown(callback) {
    let counter = 3;
    let countdownInterval = setInterval(() => {
        ctx.clearRect(0, 0, myCanvas.width, myCanvas.height);
        ctx.font = "32px Arial";
        ctx.fillStyle = "#000000";
        ctx.textAlign = "center";

        ctx.shadowColor = "rgba(0, 0, 0, 0.3)";
        ctx.shadowOffsetX = 1;
        ctx.shadowOffsetY = 1;
        ctx.shadowBlur = 2;

        ctx.fillText(counter > 0 ? counter : "GO!", myCanvas.width / 2, myCanvas.height / 2);

        if (counter === 0) {
            clearInterval(countdownInterval);
            setTimeout(() => {
                if (callback) callback();
            }, 1000); // Esperar un segundo antes de iniciar el juego
        }
        counter--;
    }, 1000); // Actualiza cada segundo
}
```

**Tasca 6:** Implementació del sistema de vides (en procés)



```
joc.vides--;
$('#vides').text('HP: ' + joc.vides); // Actualizar la visualización de vidas

if (joc.vides > 0) {
    window.restartCountdown(); // Reiniciar el contador
} else {
    joc.stopTimer();
    let playerName = prompt("Game Over! Enter your name:");
    if (playerName) {
        updateHighscores(playerName, joc.elapsedTime);
    }
    displayHighscores();
}
return;
```

## SETMANA 5

### IMPLICACIÓ DE CADA INTEGRANT

Noms dels integrants del grup	Valoració
Liquihao Qiu	1
Laura Martínez	1
Pau Serra	1

### TASQUES REALITZADES

Número de tasca	Descripció de la tasca
Tasca 1	Implementació de persistència (en procés)
Tasca 2	Millora visual final de partida
Tasca 3	Millora de la UI
Tasca 4	Implementació de comptador en perdre una vida

## EVIDÈNCIES

### Tasca 1: Implementació de persistència (en procés)



```
// Highscore functions
function updateHighscores(name, score) {
    let highscores = JSON.parse(localStorage.getItem('highscores')) || [];
    highscores.push({ name: name, score: score });
    highscores.sort((a, b) => a.score - b.score); // Ordenar por score
    if (highscores.length > 10) highscores.pop(); // Mantener solo los 10 mejores
    localStorage.setItem('highscores', JSON.stringify(highscores));
}

function displayHighscores() {
    let highscores = JSON.parse(localStorage.getItem('highscores')) || [];
    let highscoreTable = '<h2>Highscores</h2><ul style="list-style-type: none; padding: 0; text-align: left;">';
    highscores.forEach(highscore => {
        highscoreTable += `<li>${highscore.name}: ${highscore.score} seconds</li>`;
    });
    highscoreTable += '</ul>';
    $('#highscore').html(highscoreTable);
}

// Mostrar highscores al cargar la página
displayHighscores();
```

## Tasca 2: Millora visual final de partida



```
function popupPerdre() {
    if (gameOverShown) return; // Si el popup ya se mostró, no hacer nada
    gameOverShown = true; // Marcar como mostrado

    $('.carta').off("click");
    var overlay = $('

</div>');
    var popup = $('

</div>');
    popup.html(`
        <div class="popup-content">
            <h2>¡Game Over!</h2>
            <p>Introduce tu nombre:</p>
            <input type="text" id="playerName" />
            <p id="errorMessage" class="error-message">Debes introducir tu nombre</p>
            <button id="closePopup">Guardar y tornar</button>
        </div>
    `);
    $('body').append(overlay).append(popup);

    $('#closePopup').on('click', function() {
        let playerName = $('#playerName').val();
        if (playerName) {
            updateHighscores(playerName, joc.elapsedTime);
            displayHighscores();
            overlay.remove();
            popup.remove();
            setTimeout(() => {
                location.reload();
            }, 100); // Asegurar que se elimine el overlay y el popup antes de recargar
        } else {
            $('#errorMessage').show();
        }
    });
}


```

### Tasca 3: Millora de la UI



```
<div id="principal" style="display: none;">
    <canvas id="joc"></canvas>
    <div class="marcador">
        <div id="time">05:00</div>
        <div id="vides">HP: 3</div>
    </div>
    <div id="highscore"></div>
</div>
```

```
.marcador {
    position: fixed;
    top: 20px;
    right: 30px;
    background-color: □ #000;
    padding: 10px;
    color: ■ #fff;
    font-family: Arial, sans-serif;
    font-size: 18px;
    border-radius: 5px;
    box-shadow: 0px 0px 10px □rgba(0, 0, 0, 0.5);
}

#highscore {
    position: fixed;
    top: 20px;
    left: 30px;
    background-color: □ #000;
    padding: 10px;
    color: ■ #fff;
    font-family: Arial, sans-serif;
    font-size: 18px;
    border-radius: 5px;
    box-shadow: 0px 0px 10px □rgba(0, 0, 0, 0.5);
}
```

### Tasca 4: Implementació de comptador en perdre una vida



```

var isPaused = false;

function countdown(callback) {
    isPaused = true; // Pause the game during countdown
    let counter = 3;
    let countdownInterval = setInterval(() => {
        ctx.clearRect(0, 0, myCanvas.width, myCanvas.height); // Clear only the countdown area
        drawGame();
        ctx.font = "32px Arial";
        ctx.fillStyle = "#000000";
        ctx.textAlign = "center";

        ctx.shadowColor = "rgba(0, 0, 0, 0.3)";
        ctx.shadowOffsetX = 1;
        ctx.shadowOffsetY = 1;
        ctx.shadowBlur = 2;

        ctx.fillText(counter > 0 ? counter : "GO!", myCanvas.width / 2, myCanvas.height / 2);

        if (counter === 0) {
            clearInterval(countdownInterval);
            setTimeout(() => {
                isPaused = false; // Resume the game
                if (callback) callback();
            }, 1000); // Wait one second before resuming the game
        }
        counter--;
    }, 1000); // Update every second
}

function restartCountdown() {
    countdown(() => {
        animacio(); // Resume animation after countdown
    });
}

```