

PRÀCTICA 2

PONG

Laboratori Multimèdia

Daniel Ruano, David Pujol, Alex Guijarro

1/06/2023

Fundació Tecnocampus, ESUPT

Integrants del Grup

Setmana 29/5

Valoració

David Pujol 2

Daniel Ruano 1

Alex Guijarro 1

Setmana 5/6

David Pujol 2

Daniel Ruano 1

Alex Guijarro 1

Tasques Realitzades

Setmana 29/5

Tasca 1	- Finalitzades les tasques de la classe pala rectangle
Tasca 2	- Finalitzades les tasques de la classe segment
Tasca 3	- Acabada la primera tasca de la funció update de la classe bola

Setmana 5/6

Pràcticament totes les tasques realitzades la setmana passada han sigut modificades perquè donen problemes durant l'execució del joc.

Tasca 1	- Actualitzar tots els errors de la setmana anterior
PalaRectangle	- La pala del jugador funciona perfectament - La funció updateAuto no està ben feta la pala es mou de forma aleatòria però no es un moviment fluid
Segment	- Cap complicació, simplement replicar l'exemple amb les altres vores

Bola	<p>Funció Update:</p> <ul style="list-style-type: none"> - Primera part cap problema, guardem el punt actual, el punt següent i creem un segment - A la segona part on revisem el xoc amb els marges en aquesta part tampoc hem tingut cap problema, només hem d'identificar si hi ha hagut un xoc amb qualsevol dels objectes - A la tercera part el que hi ha implementat actualment no hi ha hagut cap problema només s'ha de canviar la velocitat segons la vora amb la que xoca. Falta implementar que identifiqui amb quina pala xoca per què reconeix la palaJugador però no la palaOrdinador. - Per revisar el xoc amb els marges hem hagut de fer prova i error per alguns paràmetres mal implementats però sense problema, simplement seguir el model del exemple. <p>S'ha implementat la funció ColocarAlCentre(), falta desenvolupar-la i donar-li més ús però encara estem solucionant errors de la funció del joc principal</p> <p>Funció revisaXocPales:</p> <ul style="list-style-type: none"> - Moltes complicacions, molts proves i errors però s'ha trobat una solució i finalment funciona.
Joc	<p>Una classe en principi molt senzilla en les tasques demanades, no ha donat cap complicació.</p> <p>Primer creem els elements, implementem la funció draw, ella sola dibuixa els objectes. També l'utilitzem a la funció update, on es crida la funció update per modificar la posició dels objectes.</p>
App	<p>No hem implementat gran cosa, simplement per començar a testear les funcions que donen error.</p>

Codi

Classe PalaRectangle

```

update(key,alcada) :void {
    if(key && key.DOWN && key.DOWN.pressed){
        /*****
        * Tasca. Definir el moviment de la pala
        * en funció de la telca premuda
        *****/
        this.puntPosicio.y = Math.min( values: alcada-this.alcada,
            this.puntPosicio.y + this.velocitatY)
    }
    if(key && key.UP && key.UP.pressed){
        /*****
        * Tasca. Definir el moviment de la pala
        * en funció de la telca premuda
        *****/
        this.puntPosicio.y = Math.max( values: 0,this.puntPosicio.y - this.velocitatY)
    }
}

```

```

updateAuto(alcada) :void {
    /*****
    * Tasca. Definir el moviment de la pala
    * automàtica en moviment constant
    * o amb variacions aleatòries
    *****/
    const direction :number = Math.floor( x: Math.random() * 2); // randomly choose 0 or 1
    const speed :number = 5; // set a fixed speed for the paddle to move
    if (direction === 0) { // move up
        this.puntPosicio.y = Math.max( values: 0, this.puntPosicio.y - speed);
    } else { // move down
        this.puntPosicio.y = Math.min( values: alcada - this.alcada, this.puntPosicio.y + speed);
    }
}

```

Classe Segment

```
* Taseq. Replicar la identificació del punt
* d'intersecció per a cada vara
*****/

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

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

//vara dreta
puntI = this.puntInterseccio(segmentVoraDreta);
if (puntI){
    //distància entre dos punts, el punt inicial del segment i el punt d'intersecció
    distanciaI = Punt.distanciaDosPunts(this.puntA,puntI);
    if (distanciaI < distanciaIMin){
        distanciaIMin = distanciaI;
        puntIMin = puntI;
        voraI = "dreta";
    }
}
```

Classe Bola

```

update(ampleCanva, altCanva, palaJugador, palaOrdinador) :void {
  /*****
   * Tasca. Actualitzar la posició de la bola tenint en compte
   * Si xoca o no amb els marges del canvas
   * Si xoca o no amb les pales dels jugadors
   *****/
  /*****
   * Identifica el punt actual
   * Defineix el punt següent. On ha d'anar la bola
   * Defineix un SEGMENT que vagi del PuntActual al PuntSegüent
   * Revisar si xoca amb les vores del canvas
   * Si xoca amb una vora superior o inferior, canviar el sentit i sortir
   * Si xoca amb una vora lateral, identificar punt aconseguit i reiniciar
   * Revisar si xoca amb una Pala
   * Si xoca, canviar el sentit en funció de si ha xocar
   * a dreta, esquerra, a dalt o a baix de la pala
   * canviar el sentit en funció d'on ha xocat i sortir
   *****/

  let puntActual=this.puntPosicio;

  let puntSeguent :Punt = new Punt( x: this.puntPosicio.x + this.velocitatx,
    | y: this.puntPosicio.y+this.velocitaty);

  let segmentTrajectoria :Segment =new Segment(puntActual, puntSeguent);

  /*****
   * Tasca. Revisar si xoca amb tots els marges del canvas
   *****/

  let xoc :boolean|{...} = this.revisaXocBot(segmentTrajectoria)||this.revisaXocTop(segmentTrajectoria)||
    | this.revisaXocRight(segmentTrajectoria)||this.revisaXocLeft(segmentTrajectoria)||
    | this.revisaXocPales(segmentTrajectoria,palaJugador, palaOrdinador);

```

```

/*****
* Tasca. Revisar si xoca amb tots els marges del canya
*****/

let xoc : boolean | {...} = this.revisaXocBot(segmentTrajectoria) || this.revisaXocTop(segmentTrajectoria) ||
    this.revisaXocRight(segmentTrajectoria) || this.revisaXocLeft(segmentTrajectoria) ||
    this.revisaXocPales(segmentTrajectoria, palaJugador, palaOrdinador);

if(xoc){
    /*****
    * Tasca. Revisar si xoca amb alguna pala i
    * en quina vora de la pala xoca
    *****/
    let xocPala : {...} = this.revisaXocPales(segmentTrajectoria, palaJugador, palaOrdinador);
    if(xocPala){
        xoc = true;

        /*****
        * Tasca. Si xoca amb alguna pala
        * canviar el sentit en funció de si ha xocar
        * a dreta, esquerra, a dalt o a baix de la pala
        * Poder heu de tenir en compte en quina pala s'ha produït el xoc
        *****/
        this.puntPosicio.x=xocPala.pI.x;
        this.puntPosicio.y=xocPala.pI.y;
        switch(xocPala.vora) {
            case "superior":
                this.velocitaty *= -1;
                break;
            case "inferior":
                this.velocitaty *= -1;
                break;
            case "esquerra":
                this.velocitatx *= -1;
                break;
            case "dreta":
                this.velocitatx *= -1;
                break;
        }
    }
}
}

```

```
1 usage  ± DavidPujol
revisaXocBot(segmentTrajectoria) : boolean | undefined {
    if(segmentTrajectoria.puntB.y + this.alçada > joc.alçada){
        let exces : number = ((segmentTrajectoria.puntB.y + this.alçada - joc.alçada)/this.velocitaty);
        this.puntPosicio.x = segmentTrajectoria.puntB.x - exces*this.velocitatx;
        this.puntPosicio.y = joc.alçada - this.alçada;
        this.velocitaty = - this.velocitaty;
        return true;
    }
}

1 usage  ± DavidPujol
revisaXocLeft(segmentTrajectoria) : boolean | undefined {
    if(segmentTrajectoria.puntB.x ≤ 0){
        this.velocitatx=0;
        this.velocitaty=0;
        this.ColocarAlCentre();
        return true;
    }
}

1 usage  ± DavidPujol
revisaXocRight(segmentTrajectoria) : boolean | undefined {
    if(segmentTrajectoria.puntB.x + this.amplada ≥ joc.amplada){
        let exces : number = ((segmentTrajectoria.puntB.x + this.amplada - joc.amplada)/this.velocitatx);
        this.velocitatx=0;
        this.velocitaty=0;
        this.ColocarAlCentre();
        return true;
    }
}
```

```
2 usages  ± DavidPujol
ColocarAlCentre() : void {
    this.puntPosicio.x=150-this.amplada/2;
    this.puntPosicio.y=75-this.alçada/2;
}
```



```
2 usages  ▲ DavidPujol *
revisaXocPales(segmentTrajectoria, palaJugador, palaOrdinador) : {...} | null {
    let interseccio : {pl: any, vora: any} = segmentTrajectoria.interseccioSegmentRectangle(palaJugador);
    let costatPala : string = "none";
    let palaId : string = "none";

    if (interseccio) {
        costatPala = interseccio.vora;
        palaId = "jugador";
    } else {
        interseccio = segmentTrajectoria.interseccioSegmentRectangle(palaOrdinador);
        if (interseccio) {
            costatPala = interseccio.vora;
            palaId = "ordinador";
        }
    }

    if (interseccio) {
        let puntVora : {costat: string, pl: any, pala: string} = {
            pI: interseccio.pI,
            costat: costatPala,
            pala: palaId
        };
        return puntVora;
    }

    return null;
}
```

Classe Joc

```
//Elements del joc
/*****
 * Tasca. Crear els elements del joc
 * Pales, bola, etc
 *****/
this.palaJugador=new PalaRectangle(new Punt( x: 15, y: this.alçada/3), amplada: 7, alçada: 30);
this.palaOrdinador = new PalaRectangle(new Punt( x: this.amplada-15, y: this.alçada/3), amplada: 7, alçada: 30);
this.bola = new Bola(new Punt( x: this.amplada/2, y: this.alçada/2), amplada: 10, alçada: 10);
```

```
inicialitza() :void {
    $(document).on( types: "keydown", selector: {joc:this}, data: function(e) :void {
        if(e.keyCode == e.data.joc.key.UP.code){
            e.data.joc.key.UP.pressed = true;
        }
        else if(e.keyCode == e.data.joc.key.DOWN.code){
            e.data.joc.key.DOWN.pressed = true;
        }
    });
    $(document).on( types: "keyup", selector: {joc:this}, data: function(e) :void {
        if(e.keyCode == e.data.joc.key.UP.code){
            e.data.joc.key.UP.pressed = false;
        }
        else if(e.keyCode == e.data.joc.key.DOWN.code){
            e.data.joc.key.DOWN.pressed = false;
        }
    });

    /*****
    * Tasca. Dibuixar inicialment els elements del joc
    * al canva: Pales, bola, etc
    *****/

    this.draw();
    //Mètode de crida recursiva per generar l'animació dels objectes
    requestAnimationFrame(animacio);
}
```

```
update() :void {  
    /*****  
    * Tasca. Actualitzar les posicions  
    * dels elements del joc  
    * al canva: Pales, bola, etc  
    *****/  
  
    this.draw();  
    this.palaOrdinador.updateAuto(this.alcada,this.velocitatJoc);  
    this.palaJugador.update(this.key,this.alcada);  
    this.bola.update(this.amplada,this.alcada,this.palaJugador,this.palaOrdinador);  
}  
  
1 usage  ± DavidPujol +1  
draw() :void {  
    this.clearCanvas();  
    /*****  
    * Tasca. Dibuixar els elements del joc  
    * al canva, un cop actualitzades  
    * les seves posicions: Pales, bola, etc  
    *****/  
  
    this.palaOrdinador.draw(this.myCtx);  
    this.palaJugador.draw(this.myCtx);  
    this.bola.draw(this.myCtx)  
}
```

Classe App

```
± DavidPujol +1
$(function() :void {

    let myCanvas = $("#joc")[0];
    let myCtx = myCanvas.getContext( contextId: "2d");

    /*****
     * Tasca. Inicialitza la classe JOC les posicions
     * dels elements del joc
     * al canva: Pales, bola, etc
     *****/

    joc = new Joc(myCanvas,myCtx)
    joc.inicialitza();
    animacio();
})

2 usages ± DavidPujol
function animacio() :void {
    joc.update();
    //Oportunitat per actualitzar les puntuacions
    //revisar si seguim jugant o no
    //Si pudem de nivell, etc

    //Crida recursiva per generar animació
    requestAnimationFrame(animacio);
}
```

Resum

Setmana 29/5

Contacte inicial amb el codi, no ens hem implicat molt només comprensió del codi i saber com funciona tot el projecte.

Setmana 5/6

S'ha modificat tot el que havien creat anteriorment. L'objectiu principal d'aquesta setmana era tenir les funcionalitats bàsiques del programa finalitzades i només tenir que pulir i finalitzar el tema del display durant la següent setmana però no ha sigut possible. Tenim gran part del programa enllestit, només queden un parell d'errors que solucionar.