

Cerchiamo adesso di implementare qualche livello successivo al primo. La prima cosa da fare è ovviamente costruirsi il nuovo livello generandolo con l'applicazione TileEdit presentata precedentemente. Creiamo quindi un nuovo array contenente tutte le informazioni sul nuovo livello. In ogni nuovo livello oltre a cambiare ovviamente lo scenario, potremmo anche aggiungere dei nuovi nemici delle trappole dei crediti e così via. Il secondo livello che io ho creato utilizzando sempre la mia applicazione LevelEdit è rappresentato dalla struttura seguente:

```
levels2=[[64,2,64,20],[69,25,10],[16,29,12],[16,30,12],[16,30,13],[16,30,14],[16,29,13],[17,29,11],[17,30,11],[7,28,13],[6,27,13],[6,26,13],[16,26,14],[16,26,15],[16,30,15],[16,30,16],[16,29,17],[16,27,18],[16,26,17],[16,26,16],[16,30,17],[16,30,18],[16,29,19],[16,28,19],[16,27,19],[16,26,18],[16,26,19],[16,30,19],[33,22,16],[33,21,16],[33,20,16],[33,19,16],[33,18,16],[33,15,16],[33,14,16],[33,11,16],[33,11,17],[33,15,18],[33,16,18],[33,21,18],[33,23,16],[33,24,16],[17,31,11],[16,27,16],[16,28,16],[16,29,16],[15,27,17],[15,28,17],[15,28,18],[15,29,18],[15,29,15],[15,28,15],[15,31,12],[15,31,13],[15,31,14],[15,31,15],[15,31,16],[15,31,17],[15,31,18],[15,31,19],[15,27,15],[15,27,14],[15,28,14],[15,29,14],[23,21,7],[33,12,17],[33,12,16],[33,14,18],[33,13,16],[33,17,18],[33,18,18],[33,19,18],[33,20,18],[33,17,16],[33,16,16],[28,14,13],[28,23,13],[28,22,12],[19,17,6],[34,17,5],[58,19,5],[34,51,20],[34,52,20],[45,58,8],[46,64,12],[16,63,12],[16,62,12],[16,61,12],[16,61,11],[16,61,10],[16,60,10],[16,60,11],[16,61,13],[16,62,13],[16,63,13],[34,53,0],[18,34,20],[8,46,15],[15,60,13],[15,60,12],[1,44,20],[16,48,11],[16,48,15],[16,48,16],[16,48,17],[16,48,18],[16,48,19],[16,47,11],[16,47,17],[16,47,18],[16,47,19],[8,47,10],[8,48,9],[16,48,10],[8,49,9],[8,50,9],[8,51,9],[8,52,9],[1,53,20],[16,49,10],[16,52,10],[8,48,14],[90,0,13],[90,0,14],[90,0,15],[90,0,16],[90,0,17],[90,1,17],[90,1,16],[90,1,15],[90,1,14],[92,0,18],[92,1,18],[92,2,18],[92,3,18],[9,1,0,19],[9,1,1,19],[9,1,2,19],[9,1,3,19],[9,1,4,19],[9,1,6,19],[9,1,5,19],[9,1,7,19],[9,1,8,19],[9,1,9,19],[9,1,2,14],[9,1,2,15],[9,1,3,15],[9,1,3,16],[9,1,2,16],[9,1,2,17],[9,1,3,17],[9,1,4,16],[128,4,15],[44,5,16],[44,6,16],[97,5,15],[98,6,15],[128,7,16],[98,8,17],[53,5,17],[53,4,17],[53,6,17],[53,7,17],[53,4,18],[53,5,18],[53,6,18],[53,7,18],[53,8,18],[53,9,18],[136,1,13],[137,2,13],[70,3,14],[84,10,12],[28,10,14],[28,11,13],[28,11,14],[28,12,12],[28,12,13],[28,12,14],[28,13,12],[28,13,13],[28,13,14],[28,14,12],[28,15,12],[28,15,13],[28,16,12],[28,16,13],[28,17,12],[28,17,13],[28,17,14],[28,18,12],[28,18,13],[28,18,14],[28,19,12],[28,19,13],[28,20,12],[28,21,14],[28,22,13],[28,22,14],[28,23,12],[28,23,14],[28,24,12],[28,24,13],[28,24,14],[28,25,12],[28,25,13],[28,25,14],[113,19,15],[113,20,15],[113,11,15],[28,10,15],[28,10,16],[28,10,17],[28,25,15],[28,25,16],[28,25,17],[71,14,9],[71,15,9],[71,16,9],[71,17,9],[71,18,9],[71,19,9],[71,20,9],[71,21,9],[71,22,9],[44,14,10],[44,15,10],[44,16,10],[44,17,10],[44,18,10],[44,19,10],[44,20,10],[44,21,10],[44,22,10],[44,23,10],[120,15,8],[120,16,8],[120,17,8],[120,18,8],[120,19,8],[120,20,8],[120,21,8],[17,14,8],[17,22,8],[23,20,7],[23,19,7],[23,18,7],[23,17,7],[23,16,7],[34,17,4],[19,18,6],[18,16,6],[20,20,6],[19,19,6],[54,18,5],[134,23,8],[128,22,7],[128,21,6],[128,19,4],[128,18,4],[128,17,3],[128,14,7],[128,13,8],[128,24,9],[128,12,10],[128,11,11],[47,35,16],[47,35,17],[47,35,18],[47,35,19],[47,34,19],[34,29,10],[27,36,11],[27,36,12],[27,36,13],[27,36,14],[27,36,15],[27,36,16],[27,36,17],[27,36,18],[27,36,19],[29,37,11],[29,37,12],[29,37,13],[29,37,14],[29,37,15],[29,37,16],[29,37,17],[29,37,18],[29,37,19],[18,36,10],[20,37,10],[50,33,19],[50,32,19],[50,32,18],[70,35,15],[70,32,17],[70,34,18],[70,33,18],[120,39,8],[3,42,7],[4,43,7],[5,44,7],[12,42,8],[12,43,8],[12,44,8],[12,42,9],[12,43,9],[12,44,9],[12,42,10],[12,43,10],[12,44,10],[12,42,11],[12,43,11],[12,44,11],[12,42,12],[12,42,17],[12,42,18],[12,42,19],[50,38,15],[50,38,16],[50,38,17],[50,38,18],[50,38,19],[50,39,14],[50,39,15],[50,39,16],[50,39,17],[50,39,18],[50,39,19],[50,40,19],[50,40,19],[50,41,19],[50,41,18],[50,41,17],[50,41,16],[50,41,15],[50,41,14],[50,41,13],[50,41,12],[50,41,11],[50,40,14],[50,40,13],[50,40,12],[50,41,10],[50,40,15],[50,40,16],[50,40,17],[50,40,18],[3,47,13],[4,48,13],[4,49,13],[4,50,13],[4,51,13],[4,52,13],[4,53,13],[4,54,13],[5,55,13],[17,47,14],[8,46,14],[12,42,13],[12,42,14],[12,42,15],[12,42,16],[14,44,17],[14,44,18],[14,44,19],[14,43,13],[14,43,14],[14,43,15],[13,43,16],[13,43,17],[13,43,18],[13,43,19],[70,38,14],[94,39,13],[93,39,12],[94,40,11],[93,40,10],[84,41,9],[8,46,11],[16,46,12],[16,46,13],[16,45,15],[16,44,15],[16,44,14],[16,44,13],[16,44,12],[17,43,12],[17,42,12],[17,44,16],[16,47,15],[17,47,16],[138,45,17],[138,46,17],[138,46,18],[138,45,18],[138,45,19],[138,46,19],[121,45,16],[121,46,16],[65,49,14],[65,50,14],[65,51,14],[65,52,14],[65,53,14],[65,54,14],[65,55,14],[65,49,15],[65,50,15],[65,51,15],[65,52,15],[65,53,15],[65,54,15],[65,55,15],[65,49,16],[65,50,16],[65,50,17],[65,49,17],[65,49,18],[65,49,19],[65,50,19],[65,50,18],[65,51,16],[65,51,17],[65,51,18],[65,51,19],[65,52,19],[65,52,18],[65,52,17],[65,52,16],[65,53,16],[65,54,16],[65,54,17],[65,53,17],[65,55,17],[65,55,16],[65,56,17],[65,56,18],[65,55,18],[65,54,18],[65,53,18],[65,53,19],[65,54,19],[65,55,19],[65,56,19],[58,56,14],[58,57,16],[58,58,18],[67,58,19],[67,56,15],[66,57,18],[67,57,17],[65,57,19],[66,56,16],[140,60,19],[140,59,19],[140,61,19],[140,62,19],[140,63,19],[140,63,18],[140,63,17],[140,63,16],[140,63,15],[140,63,14],[142,59,18],[70,52,12],[60,48,12],[60,54,12],[8,53,9],[8,54,9],[16,53,10],[16,54,10],[16,49,8],[16,50,8],[16,51,8],[16,52,8],[16,53,8],[16,54,8],[17,49,7],[17,50,7],[17,51,7],[17,52,7],[17,53,7],[17,54,7],[37,51,5],[38,52,5],[39,53,5],[47,52,6],[49,53,6],[45,51,6],[84,52,4],[49,63,8],[38,62,8],[38,61,8],[38,60,8],[38,59,8],[36,58,7],[40,63,7],[8,60,9],[8,61,9],[34,60,14],[34,60,15],[34,60,16],[34,60,17],[34,60,18],[62,61,14],[62,62,14],[62,62,15],[62,62,16],[62,62,17],[62,61,17],[62,61,16],[62,61,15],[71,62,9],[71,63,9],[71,63,10],[71,63,11],[71,62,10],[71,62,11],[62,61,18],[62,62,18],[50,11,18],[50,12,18],[50,13,17],[50,13,18],[50,14,17],[50,15,17],[50,16,17],[50,17,17],[50,18,17],[50,19,17],[50,20,17],[50,21,17],[50,22,17],[50,24,18],[28,25,18],[28,25,19],[28,24,19],[28,10,18],[28,10,19],[28,11,19],[33,23,17],[33,24,17],[47,23,18],[47,22,18],[113,12,19],[113,13,19],[113,14,19],[113,15,19],[113,16,19],[113,17,19],[113,18,19],[113,19,19],[113,21,19],[113,22,19],[113,23,19],[113,12,15],[113,13,15],[113,14,15],[113,15,15],[113,16,15],[113,17,15],[113,18,15],[113,21,15],[113,22,15],[113,23,15],[113,24,15],[113,14,14],[113,15,14],[113,16,14],[113,19,14],[113,20,14],[113,20,13],[113,21,13],[113,21,12],[8,13,11],[8,14,11],[8,15,11],[8,16,11],[8,17,11],[8,18,11],[8,19,11],[8,20,11],[8,21,11],[8,22,11],[8,23,11],[8,24,11],[8,12,11],[8,11,12],[8,10,13],[8,25,11],[8,24,10],[44,13,10],[44,13,9],[120,23,9],[24,10],[23,10],[22,10],[21,10],[20,10],[19,10],[18,10],[17,10],[16,10],[15,10],[14,10],[13,10],[12,11],[11,12],[10,13],[10,14],[10,15],[10,16],[10,17],[9,18],[8,17],[7,17],[6,15],[6,16],[25,11],[25,12],[26,13],[27,13],[28,13],[29,12],[29,11],[30,11],[31,11],[21,7],[20,6],[19,6],[18,5],[17
```

```
,6],[16,6],[16,7],[15,8],[16,8],[17,7],[18,7],[19,7],[58,7],[58,8],[59,8],[60,8],[61,8],[62,8],[63,7],[54,7],[54,8],[54,9],[54,10],[55,13],[56,14],[56,15],[57,16],[57,17],[58,18],[58,19],[53,7],[52,7],[51,7],[50,7],[49,7],[49,8],[48,9],[47,10],[47,13],[47,16],[47,17],[47,18],[47,19],[44,16],[44,17],[44,18],[44,19],[43,15],[43,14],[43,13],[43,12],[42,12],[42,13],[42,14],[42,15],[42,16],[42,17],[42,18],[42,19],[37,10],[37,11],[37,12],[37,13],[37,14],[37,15],[37,16],[37,17],[37,18],[37,19],[36,19],[36,18],[36,17],[36,16],[36,15],[36,14],[36,13],[36,12],[36,11],[36,10],[31,12],[31,13],[31,14],[31,15],[31,16],[31,17],[31,18],[31,19],[4,17],[5,15],[5,16],[42,7],[43,7],[44,7],[46,14],[49,13],[50,13],[51,13],[52,13],[49,10],[50,9],[51,9],[52,10],[53,13],[54,13],[53,10],[48,11],[47,11],[48,13],[1,18],[2,18],[3,18],[6,0,11],[60,13],[60,10],[60,9],[60,12],[39,8],[0,18],[13,9],[14,8],[29,10],[0,7],[0,8],[[22,9,"coin"],[21,9,"coin"],[20,9,"coin"],[19,9,"coin"],[19,8,"coin"],[18,9,"coin"],[18,8,"coin"],[17,8,"coin"],[17,9,"coin"],[16,9,"coin"],[15,9,"coin"],[14,9,"coin"],[15,7,"coin"],[16,5,"coin"],[17,5,"coin"],[17,4,"coin"],[18,4,"coin"],[18,3,"coin"],[17,3,"coin"],[19,3,"coin"],[19,2,"coin"],[20,2,"coin"],[20,3,"coin"],[21,3,"coin"],[21,4,"coin"],[22,4,"coin"],[22,5,"coin"],[23,5,"coin"],[23,6,"coin"],[24,6,"coin"],[24,7,"coin"],[25,8,"coin"],[25,7,"coin"],[30,10,"coin"],[28,12,"coin"],[27,12,"coin"],[26,12,"coin"],[8,16,"coin"],[8,15,"coin"],[7,15,"coin"],[7,16,"coin"],[8,14,"coin"],[9,17,"coin"],[58,6,"turret"],[45,10,"umbrella"],[27,10,"umbrella"],[10,8,"umbrella"],[43,6,"coin"],[42,6,"coin"],[44,6,"coin"],[43,5,"coin"],[36,7,"coin"],[37,7,"coin"],[42,10,"coin"],[43,10,"coin"],[47,9,"coin"],[48,8,"coin"],[50,6,"coin"],[50,5,"coin"],[51,4,"coin"],[51,5,"coin"],[52,3,"coin"],[52,4,"coin"],[53,4,"coin"],[54,5,"coin"],[55,4,"coin"],[56,3,"coin"],[57,3,"coin"],[58,3,"coin"],[59,4,"coin"],[60,5,"coin"],[58,12,"umbrella"],[63,6,"portale"],[50,11,"coin"],[51,11,"coin"],[52,11,"coin"],[53,11,"coin"],[54,11,"coin"],[56,13,"coin"],[57,14,"coin"],[57,15,"coin"],[58,16,"coin"],[58,17,"coin"],[1,13,"inizio"],[29,9,"turret"]]]];
```

Come si vede ho inserito un nuovo nemico rappresentato da una torretta che si attiva a tempi prestabiliti sparando dei proiettili



come prima cosa carichiamo gli sprites per questi nuovi oggetti nella funzione Init del file main.js.

```
this.sprTurret = this.gr.LoadSprite("immagini/turret.png", 2);
this.sprTurretBullet = this.gr.LoadSprite("immagini/turretbullet.png", 1);
this.sprExplosion = this.gr.LoadSprite("immagini/explosion.png", 4);
```

creiamo quindi un nuovo file Torretta.js nel quale inseriamo il codice seguente:

```
function Turret(x,y){
    GameObj.call(this, x, y);
    this.SetSprite(game.sprTurret, true);
    this.x += this.sprite.w/2;
    this.nome="torretta";
    this.timer = 50 * Math.random();// tempo casuale con i quale si attiva per sparare
    this.animSpeed = 0;
    this.shotTimer = 0;
    this.life = 5;//numero di volte che deve essere colpito per essere distrutto
    this.bbox = new BoundingBox(this.x-this.sprite.w/2+5, this.y, this.sprite.w-10,
this.sprite.height);

    this.Update = function(){
        this.timer--;

        if(this.timer<=0){
            this.curFrame = !this.curFrame;//si attiva o disattiva in base al timer
            this.timer = 200;
            this.shotTimer = 0;
            if(game.player.x < this.x) this.flip = 1;//direzione di sparo cambia in
base alla posizione del player
            else this.flip = -1;
        }

        if(this.curFrame == 1){//se è nello stato di sparo
            this.shotTimer --;
            if(this.shotTimer<=0){//se è trascorso il tempo per un nuovo sparo
                //effettua lo sparo creando un nuovo proiettile
                new TurretBullet(x + 32 - this.flip*32,y+32,-10*this.flip);

                AudioPlay(game.sndShotEnemy);
                this.shotTimer = 80;
            }
        }
        if(this.life < 0){
            this.Destroy();
        }
    }
}
```

```

    this.OnDestroy = function(){
        game.nemici.splice(game.nemici.indexOf(this), 1);
        new Explosion(this.x, this.y);
    }

    game.nemici.push(this);
}
Eredita(Turret);

```

come si vede nel codice la torretta spara dei proiettili ad intervalli di tempo calcolati. Abbiamo aggiunto il nome dell'oggetto per riconoscerlo quando lo colpiamo con un proiettile e fare in modo che subisca danni solo quando è aperta. Creiamo quindi in questo stesso file anche la funzione per la creazione dell'oggetto TurretBullet con il codice seguente:

```

function TurretBullet(x,y,hspeed){
    GameObj.call(this, x, y);
    this.SetSprite(game.sprTurretBullet);
    this.OffsetCenter();
    this.hSpeed = hspeed;
    this.bbox = new BoundingBox(x-this.xOffset, y-this.yOffset, this.sprite.width,
this.sprite.height);

    this.Update = function(){//se incontra un muro si distrugge
        if(this.GetCollision(game.blocks, this.hSpeed/2, 0)){
            this.Destroy();
        }
        //se il player non è nello stato colpito e collide con esso
        if(!game.player.hit)
            if(this.bbox.CollidesAt(game.player.bbox, this.hSpeed/2, 0)){
                game.player.Hit();
                this.Destroy();
            }
        this.x += this.hSpeed;
        this.bbox.Move(this.x-game.sprTurretBullet.width/2, this.y-
game.sprTurretBullet.height/2);
        //se il proiettile esce dallo schermo si distrugge
        if(this.x > game.AreaW * game.cellSize || this.x < 0){
            this.Destroy();
        }
    }

    this.OnDestroy = function(){
        game.bullets.splice(game.bullets.indexOf(this), 1);
        new BulletExplosion(this.x, this.y);
    }

    game.bullets.push(this);
}
Eredita(TurretBullet);

```

modifichiamo il file GameObjects.js per fare in modo che la torretta venga disegnata nella direzione per sparare al player. Il codice che dobbiamo aggiungere è:

*/\*questo metodo realizza il draw per GameObject ribaltabili ad esempio la torretta si gira in base alla posizione\**

*\*del player  
\*/*

```

this.__DrawAnimatedMirror = function(){
    game.ctx.save();
    game.ctx.translate(this.x-game.viewX, this.y-game.viewY);
    game.ctx.scale(this.flip, 1);
    var ox = Math.floor(this.curFrame) * this.sprite.w;
    game.ctx.drawImage(this.sprite, ox, 0, this.sprite.w, this.sprite.height, -
this.sprite.w/2, 0, this.sprite.w, this.sprite.height);
    game.ctx.restore();
}

```

E modifichiamo anche la funzione SetSprite di GameObjct come segue:

```

/*questo metodo imposta lo sprite e seleziona l'evento draw corretto*/
this.SetSprite = function(sprite, mirror){
    if(sprite == null || sprite == undefined){
        this.Draw = function(){};
        return;
    }

    this.sprite = sprite;

    if(mirror != undefined){
        this.flip = 1;
        this.Draw = this.__DrawAnimatedMirror;
    }
    else{
        if(sprite.frames > 0){
            this.Draw = this.__DrawAnimated;
        }else{
            this.Draw = this.__DrawSimple;
            this.UpdateAnimation = function(){};
        }
    }
}

```

Modifichiamo quindi l'oggetto Bullet per fare in modo che danneggi la torretta solo quando è aperta cambiando il codice:

```

if(inst = this.GetCollision(game.nemici, this.hSpeed/2, 0)){
    inst.life--;
    this.Destroy();
}

```

Con il codice:

```

if (inst = this.GetCollision(game.nemici, this.hSpeed / 2, 0)) {
    if (inst.name == "torretta") {
        if (inst.curFrame == 1) { inst.life--; }
    }
    else
        inst.life--;
    this.Destroy();
}

```

facciamo in modo che questo nuovo oggetto sia caricato nel gioco inserendo nello switch di main.js il case per questo oggetto:

```

.....
switch (objects[i][2]) {
    case "umbrella":
        new Umbrella(x, y);
        break;
    case "turret":
        new Turret(x, y);
        break;
}
.....

```

Importiamo in index.html il file Torretta.js

Carichiamo in init del main il suono per lo sparo del nemico:

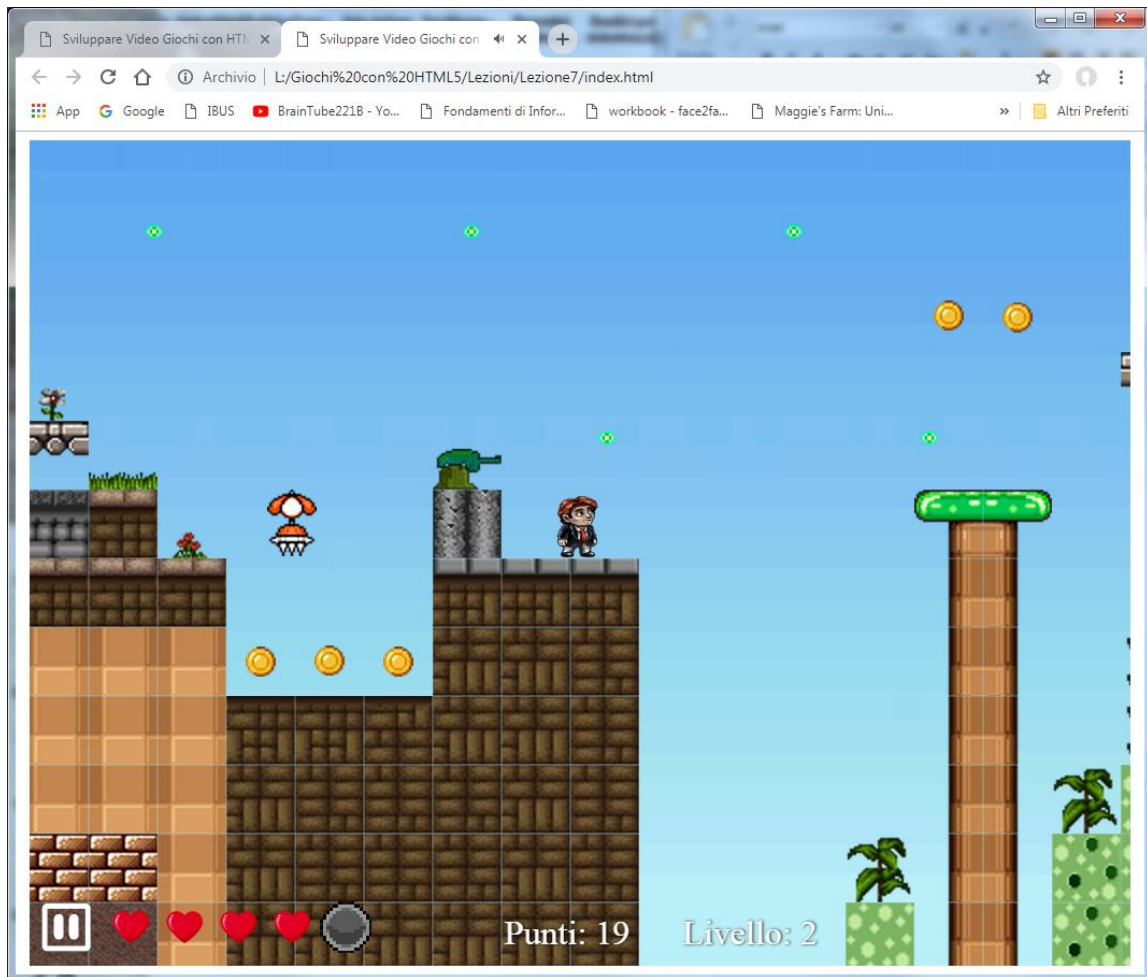
```

//caricamento suoni
this.sndShotEnemy=this.gr.LoadSound("audio/shotenemy.mp3");

```

proviamo a mandare in esecuzione dovremmo avere le torrette posizionate che sparano e ruotano in base alla posizione del player.

Di seguito uno screenshot del secondo livello:



Adesso facciamo in modo che quando passiamo ad un livello successivo il player mantenga il punteggio realizzato nel livello precedente. Aggiungiamo in fondo a `init` di `main.js`:

```
this.score=0;
```

ed eliminiamo la stessa riga dalla funzione `ResetLevel`.

A questo punto dobbiamo gestire la fine dei livelli. Per gestire questa fase potremmo semplicemente pensare che venga ripresentato il menù iniziale oppure visualizzare uno splash screen che mostra i crediti classifiche ecc. Io mi limito a ripresentare il menù iniziale aggiungendo alla funzione `LoadLevel` Di `Main.js` il codice seguente:

```
/*metodo per caricare un livello di gioco*/
this.LoadLevel = function (lev) {
  this.level = lev;
  this.ResetLevel();
  /*Non ci sono nuovi livelli da caricare*/
  if (levels[lev] == undefined) {
    this.level = 0;
    this.score = 0;
    lev = 0;
  }
  /*Il resto rimane uguale*/
  if (lev == 0) {
    this.mainMenu = new MainMenu();
    this.score = 0;
  }
  else {
    // carica le info sul livello
    .....
  }
}
```

Come ultima cosa possiamo aggiungere un abbellimento al display che visualizza le vite visualizzando dei cuori pulsanti. Per ottenere funzionalità nella funzione `Init` di `main.js` sostituiamo l'istruzione:

```
this.sprLife = this.gr.LoadSprite("immagini/life.png", 1);
```

con l'istruzione:

```
this.sprLife = this.gr.LoadSprite("immagini/life5.png", 5);
```

in modo tale da caricare lo sprite animato per il cuore poi alla fine del file display.js creiamo un nuovo oggetto: Life come segue:

```
function Life(x, y) {
    // costruttore
    GameObj.call(this, 90 + x * 50, y);
    // sprite e metodo draw
    this.SetSprite(game.sprLife);
    this.OffsetCenter();
    this.animSpeed = 0.15;
    this.Update = function ()
    {
        this.x=game.viewX+(90 + x * 50);
        this.y=game.viewY + game.canvas.height-30;
    }
    this.OnDestroy = function(){
        game.heart.splice(game.heart.indexOf(this), 1);
    }
}
Eredita(Life); //eredita tutte le funzionalità dei GameObject
Sempre nel file dispaly.js nella funzione Draw dell'oggetto Display eliminiamo le seguenti righe:
//disegna il numero di vite disponibili
    for (var i = 0; i < 5; i++) {
        if (game.player.lives > i)
            game.ctx.drawImage(game.sprLife, 70 + i * 50, game.canvas.height - 60);
        else
            game.ctx.drawImage(game.sprLifeLost, 70 + i * 50, game.canvas.height - 60);
    }
```

E aggiungiamo invece all'inizio le righe seguenti:

```
function Display() {
    for (var i = 0; i < 5; i++){
        var life = new Life(i, game.canvas.height - 30);
        //imposta il punto di ingresso
        game.heart.push(life);
    }
    this.Draw = function () {
        //variabili per i bottoni
    }
}
.....
```

Aggiungiamo nella funzione ResetLevel del file Main.js il vettore heart per contenere gli oggetti Life come di seguito. Riporto il metodo definitivo:

```
/*metodo per resettare il gioco*/
this.ResetLevel = function() {
    this.mainMenu = null;
    this.display = null;
    this.paused = false;
    this.levelCompleted = null;
    this.blocks = [];
    this.tiles= [];
    this.gameobjects = [];
    this.nemici =[];
    this.bullets = [];
    this.heart = [];
}
/*fine reset*/
```

E all'interno dell'oggetto Player facciamo in modo che quando il player perde una vita venga eliminato un oggetto life dai GameObjects. Per ottenere Questo aggiungiamo il codice seguente nella funzione Hit:

```
.....
// riduce le vite di 1
    this.lives--;
    game.heart[this.lives].Destroy();
    game.heart.splice(this.lives,1);
    if (this.lives <= 0) {
        this.Die();
    }
}
.....
```

E allo stesso modo quando il player supera l'area canvas. Quindi aggiungiamo:

```
.....  
//se il player cade sotto il livello, muore  
    if (this.y > game.areaH) {  
        this.lives--;  
        game.heart[this.lives].Destroy();  
        game.heart.splice(this.lives,1);  
        this.x = this.xStart;  
        this.y = this.yStart;  
        if (this.lives <= 0) {  
            this.Die();  
        }  
    }  
}  
.....
```

E modifichiamo la funzione Die come segue:

```
this.Die = function () {  
    game.LoadLevel(0);  
}
```

Eliminiamo la riga che carica l'immagine di vita persa nella funzione init che ormai non serve più:  
`this.sprLifeLost = this.gr.LoadSprite("immagini/lifeLost.png", 1);`

Lascio a voi sviluppare tutte le funzionalità che vorrete aggiungere

# Fine