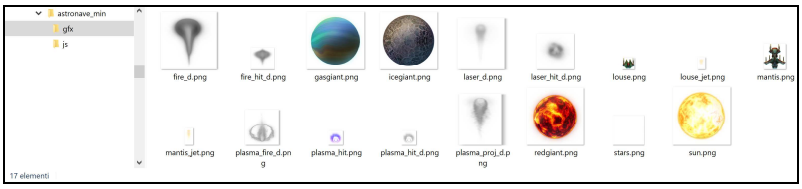
PixiJS: ASTRONAVE_min

Creare la cartella ***astronave_min*** sotto 1Bi e poi le 2 cartelle ***gfx*** e ***js*** sotto *astronave_min*

Ricordandosi di selezionare *"Tutti i file (*.*)"* per l'opzione *Salva Come*:

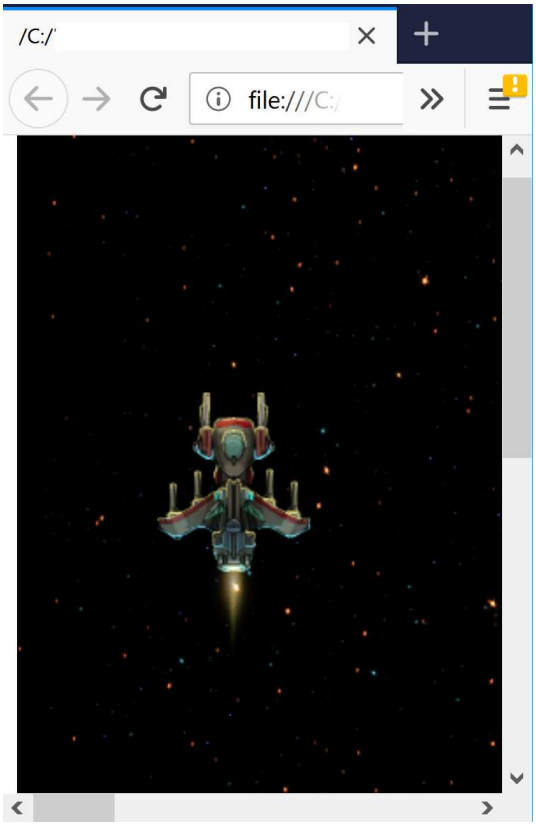
1. creare il file ***index.html*** salvandolo nella cartella ***astronave_min***
2. creare i file ***main.js*** e ***keyboard.js*** nella cartella ***js*** (i codici HTML e JS sono riportati alla pagina successiva)
3. copiare nella cartella ***gfx*** le immagini di seguito indicate inserite nel cartella:
   http://laprofpgr.altervista.org/sitoSc/appINF/files/VG/astronave_min/gfx/

mantis.png	fire_d.png			
stars.png	fire_hit_d.png			
sun.png	laser_d.png			
mantis_jet.png	laser_hit_d.png			
icegiant.png	gasgiant.png	louse.png	plasma_fire_d.png	plasma_hit_d.png
	redgiant.png	louse_jet.png	plasma_hit.png	plasma_proj_d.png

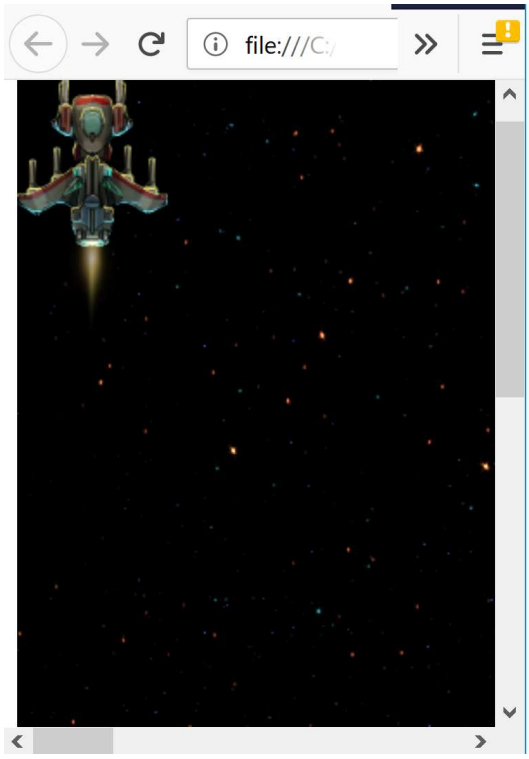


Aprendo con Google Chrome o Firefox la pagina ***index.html*** si vede l'astronave su uno sfondo stellato.

1) situazione iniziale



2) l'astronave si sposta utilizzando le frecce a sinistra e in alto



index.html

```
<html>
  <head>
    <script src="https://pixijs.download/release/pixi.min.js"></script>
  </head>

  <body>
   <div id="pixiview" name="pixiview"> </div>
  </body>

  <script src="js/keyboard.js"></script>
  <script src="js/main.js"></script>
</html>
```

keyboard.js

```
function keyboard(keyCode) {
  let key  = {};
  key.code = keyCode;
  key.isDown  = false;
  key.isUp    = true;
  key.press   = undefined;
  key.release = undefined;
  //The `downHandler`
  key.downHandler = event => {
    if (event.keyCode === key.code) {
      if (key.isUp && key.press) key.press();
      key.isDown = true;
      key.isUp = false;
    }
    event.preventDefault();
  };

  //The `upHandler`
  key.upHandler = event => {
    if (event.keyCode === key.code) {
      if (key.isDown && key.release) key.release();
      key.isDown = false;
      key.isUp = true;
    }
    event.preventDefault();
  };

  //Attach event listeners
  window.addEventListener(
    "keydown", key.downHandler.bind(key), false
  );
  window.addEventListener(
    "keyup",   key.upHandler.bind(key),   false
  );
  return key;
}
```

main.js

```
var app = new PIXI.Application(
      {
            width: 1280,
            height: 720,
            orwidth: 1280,
            orheight: 720
      }
);
gamel = document.getElementById("pixiview");
gamel.appendChild(app.view);

game = {
      back: {},
      players: {},
      input: {},
      objects: []
}
game.switchFullscreen = function () {
      hratio = app.screen.height / app.screen.width;

      if (document.fullscreenElement || document.webkitFullscreenElement ||
          document.mozFullScreenElement) {
             if(document.exitFullscreen)
                  document.exitFullscreen();
             else if (document.webkitExitFullscreen)
                  document.webkitExitFullscreen();
             else if (document.msExitFullscreen)
                  document.msExitFullscreen();

            //app.stage.scale.x = app.stage.scale.y = 1.0;
            //app.renderer.resize(app.screen.orwidth, app.screen.orheight);

            this.UIfullscreen = false;

      } else {

            if (app.renderer.view.requestFullscreen) {
              app.renderer.view.requestFullscreen();
            } else if (app.renderer.view.msRequestFullscreen) {
              app.renderer.view.msRequestFullscreen();
            } else if (app.renderer.view.mozRequestFullScreen) {
              app.renderer.view.mozRequestFullScreen();
            } else if (app.renderer.view.webkitRequestFullscreen) {
              app.renderer.view.webkitRequestFullscreen();
            }
```

```
            //nheight = Math.ceil(window.outerWidth*hratio);

            //app.stage.scale.x = window.outerWidth/app.screen.height;
            //app.stage.scale.y = nheight/app.screen.height;

            //app.renderer.resize(window.outerWidth, nheight);

            app.UIfullscreen = true;
      }
      return app.UIfullscreen;
}

game.input.enter = keyboard(13);
game.input.enter.press = function () {
      game.switchFullscreen();
}

///// LOADER
app.loader.add("stars","gfx/stars.png");
app.loader.add("mantis","gfx/mantis.png");
app.loader.add("mantis_jet","gfx/mantis_jet.png");

////////////////////////////

game.back.space = new PIXI.Container();
game.back.space.addChild(new PIXI.extras.TilingSprite.fromImage("gfx/stars.png",2998,1529,false));
game.back.space.addChild(new PIXI.extras.TilingSprite.fromImage("gfx/stars.png",2998,1529,false));
game.back.space.addChild(new PIXI.extras.TilingSprite.fromImage("gfx/stars.png",2998,1529,false));

{
      function shipCommon(ship) {

            var shipgfx = ship.gfx;

            shipgfx.anchor.x = 0.5;
            shipgfx.anchor.y = 0.5;
            shipgfx.hwidth = shipgfx.width*0.5;
            shipgfx.hheight = shipgfx.height*0.5;
            shipgfx.gamebounds = {
                  right: app.screen.width - shipgfx.hwidth,
                  left: shipgfx.hwidth,
                  top: shipgfx.hheight,
                  bottom: app.screen.height - shipgfx.hheight
            }
      }
```

```
game.addShip = function (name,type) {
	var ship = {};

	ship.life = 100;
	ship.shield = 100;
	ship.speed = 8;
	ship.curspeed = 0;
	ship.name = name;
	ship.type = type;
	ship.active = true;
	ship.vx = 0;
	ship.vy = 0;

	ship.stimer = {dt: 0};

	return ship;
}

game.addPlayer = function (name) {
	var ship = this.addShip(name,"player");

	ship.gfx = new PIXI.Sprite.fromImage("gfx/mantis.png");
	ship.gfx.addChild(new PIXI.Sprite.fromImage("gfx/mantis_jet.png"));

	shipCommon(ship);

	ship.sanchor = [{x: -15,y: 50},{x: 15,y: 50}];
	////////////////////

	ship.gfx.x = app.screen.width*0.5;
	ship.gfx.y = app.screen.height*0.5;
	// SHIP Engine
	engine = ship.gfx.children[0];
	engine.anchor.x = 0.5;
	engine.anchor.y = 0.5;

	engine.x = 0;
	engine.y = ship.gfx.height*0.7;
	engine.blendMode = PIXI.BLEND_MODES.ADD;
	game.input.left = keyboard(37);
	game.input.up = keyboard(38);
	game.input.right = keyboard(39);
	game.input.down = keyboard(40);

	game.input.space = keyboard(32); // dopo...
```

```
        ship.update = function () {

            if(this.active){
                ship.vx = (game.input.right.isDown) ? this.speed : 0;
                ship.vx = ship.vx + ((game.input.left.isDown) ? -this.speed : 0);
                ship.vy = (game.input.up.isDown) ? -this.speed : 0;
                ship.vy = ship.vy + ((game.input.down.isDown) ? this.speed : 0);

                ship.gfx.x = ship.gfx.x + ship.vx*app.ticker.deltaTime;
                ship.gfx.y = ship.gfx.y + ship.vy*app.ticker.deltaTime;

                ship.gfx.x = (ship.gfx.x > ship.gfx.gamebounds.right) ?
                             ship.gfx.gamebounds.right : ship.gfx.x;
                ship.gfx.x = (ship.gfx.x < ship.gfx.gamebounds.left) ?
                             ship.gfx.gamebounds.left : ship.gfx.x;
                ship.gfx.y = (ship.gfx.y > ship.gfx.gamebounds.bottom) ?
                             ship.gfx.gamebounds.bottom : ship.gfx.y;
                ship.gfx.y = (ship.gfx.y < ship.gfx.gamebounds.top) ?
                             ship.gfx.gamebounds.top : ship.gfx.y;

                ship.stimer.dt = ship.stimer.dt + app.ticker.deltaTime;
            }
        }

        game.objects[game.objects.length] = ship;
        app.stage.addChild(ship.gfx);
        return ship;
    }
}

{
    var space = game.back.space;

    ////////////////////////////////////////

    var stars1 = space.children[0];
    var stars2 = space.children[1];
    var stars3 = space.children[2];

    stars1.scale.x = 0.4;
    stars1.scale.y = 0.4;
    stars2.scale.x = 0.75;
    stars2.scale.y = 0.75;

    stars1.tint = 0x5588FF;
    stars2.tint = 0x55FFFF;
    stars3.tint = 0xFF8844;
```

```
	stars1.alpha = 3.0;
	stars2.alpha = 3.0;
	stars3.alpha = 8.0;

	stars1.blendMode = PIXI.BLEND_MODES.ADD;
	stars2.blendMode = PIXI.BLEND_MODES.ADD;
	stars3.blendMode = PIXI.BLEND_MODES.ADD;
}

app.stage.addChild(game.back.space);


game.update = function() {


	var ticker = app.ticker;

	var space = game.back.space;
	var stars1 = space.children[0];
	var stars2 = space.children[1];
	var stars3 = space.children[2];


	stars1.tilePosition.y = stars1.tilePosition.y+0.25*ticker.deltaTime;
	stars2.tilePosition.y = stars2.tilePosition.y+1.5*ticker.deltaTime;
	stars3.tilePosition.y = stars3.tilePosition.y+3*ticker.deltaTime;

	for(i=0;i<game.objects.length;i++){
		game.objects[i].update();
	}
}

app.loader.onComplete.add(function () {

	var ship = game.addPlayer("test");

	app.ticker.add(game.update);
});
app.loader.load();
```