

Bacione Javascript Editor is an Integrated Development Environment (IDE) for programming small javascript games on a phone.

License: GPL

Github: github.com/bacionejs

Several games are included in the editor: ski, marslander, chess, monkeymaze, snake, pairs, smash.

Below are several screenshots of ski and marslander with relevant code highlighted to help understand what the code does.

ski

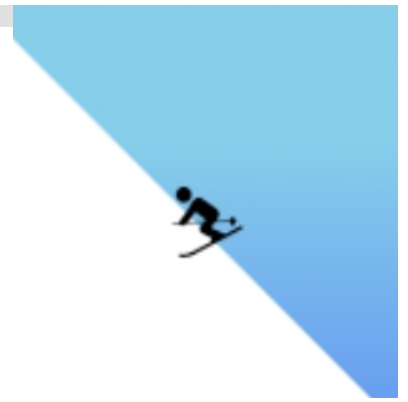
```
//tree
let speed=W/FPS/3, os=[], c=L.canvas();
c.white();
c.fillStyle=L.gradient(c,W,0,[.2,"blue"],[.8,"skyblue"]);
c.fill(L.shape([[0,0,W,W,W,0]]));
for(let i=0;i<2;i++)
os.push(
(
function(){
let s=floor(W/10), c=L.canvas(s), x=0;
return function(){
c.clear();
x=x>0?x-speed:L.rnd(W*2,W);
c.icon(127794,x,x);
}
}
)()
);
(
function animation(){
os.forEach(o=>o?().());
requestAnimationFrame(animation);
}
)()
```



```

//skier
let speed=W/FPS/3, os=[], c=L.canvas();
c.white();
c.fillStyle=L.gradient(c,W,0,[.2,"blue"],[.8,"skyblue"]);
c.fill(L.Shape([[0,0,W,W,W,0]]));
for(let i=0;i<2;i++)
os.push(
(
function(){
let s=floor(W/10), c=L.canvas(s), x=0;
return function(){
c.clear();
x=x>0?x-speed:L.rnd(W*2,W);
c.icon(127794,x,x);
}
}
)()
);
os.push(
(
function(){
let s=floor(W/20), c=L.canvas(s), x=W/4;
c.translate(x,x);
c.icon(9975);
}
)()
);
(
function animation(){
os.forEach(o=>o?o.());
requestAnimationFrame(animation);
}
)()

```



```

//color and flip
let speed=W/FPS/3, os=[], c=L.canvas();
c.white();
c.fillStyle=L.gradient(c,W,0,[.2,"blue"],[.8,"skyblue"]);
c.fill(L.shape([[0,0,W,W,W,0]]));
for(let i=0;i<2;i++)
os.push(
(
function(){
let s=floor(W/10), c=L.canvas(s), x=0;
return function(){
c.clear();
x=x>0?x-speed:L.rnd(W*2,W);
c.icon(127794,x,x);
};
})();
);
os.push(
(
function(){
let s=floor(W/20), c=L.canvas(s), x=W/4;
c.fillStyle="red";
c.scale(-1,1);
c.translate(-x,x);
c.icon(9975);
};
})();
);
(
function animation(){
os.forEach(o=>o?o.():);
requestAnimationFrame(animation);
}
)()

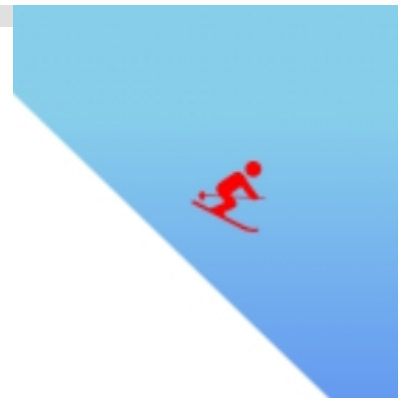
```



```

//jump
let speed=W/FPS/3, os=[], c=L.canvas();
c.white();
c.fillStyle=L.gradient(c,W,0,[.2,"blue"],[.8,"skyblue"]);
c.fill(L.shape([[0,0,W,W,W,0]]));
for(let i=0;i<2;i++)
os.push(
(
function(){
let s=floor(W/10), c=L.canvas(s), x=0;
return function(){
c.clear();
x=x>0?x-speed:L.rnd(W*2,W);
c.icon(127794,x,x);
};
})();
);
os.push(
(
function(){
let s=floor(W/20), c=L.canvas(s), x=W/4;
c.fillStyle="red";
c.scale(-1,1);
c.translate(-x,x);
c.icon(9975);
let tgl=1;
addEventListener("click",()=>i|=1,false);
function jump(){
c.clearRect(-s,-s,s*2,s*2);
c.translate((tgl*=-1)*(s),0);
c.icon(9975);
}
let i=0;
let time=floor(s/speed);
let reset=3*time;
return function(){
if(i==1 || i==reset){
jump();
}
if(i==reset){
i=0;
}
i && i++;
};
})();
);
(
function animation(){
os.forEach(o=>o?().());
requestAnimationFrame(animation);
}
)()

```



```

//hit
let hit;
let speed=W/FPS/3, os=[], c=L.canvas();
c.white();
c.fillStyle=L.gradient(c,W,0,[.2,"blue"],[.8,"skyblue"]);
c.fill(L.shape([[0,0,W,W,W,0]]));
for(let i=0;i<2;i++)
os.push(
(
function(){
let s=floor(W/10), c=L.canvas(s), x=0;
return function(){
c.clear();
x=x>0?x-speed:L.rnd(W*2,W);
c.icon(hit?(x)?128165:127794,x,x);
}
}
)()
);
os.push(
(
function(){
hit=function(t){ return((x-s)<t && t<(x+s) && !(0<i && i<reset)); }
let s=floor(W/20), c=L.canvas(s), x=W/4;
c.fillStyle="red";
c.scale(-1,1);
c.translate(-x,x);
c.icon(9975);
let tgl=1;
addEventListener("click",()=>i||=1,false);
let i=0;
let time=floor(s/speed);
let reset=3*time;
function jump(){
c.clearRect(-s,-s,s*2,s*2);
c.translate((tgl*=-1)*(s),0);
c.icon(9975);
}
return function(){
if(i==1 || i==reset){
jump();
}
if(i==reset){
i=0;
}
i && i++;
}
};
}
)()
);
(
function animation(){
os.forEach(o=>o?().());
requestAnimationFrame(animation);
}
)()

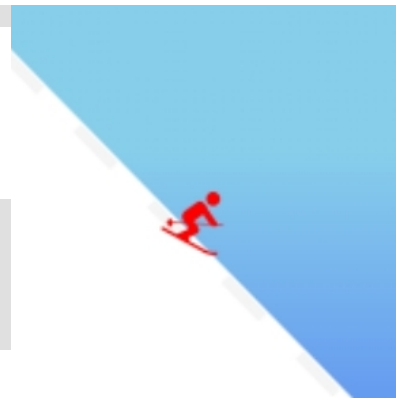
```



```

//motion effect
let hit;
let speed=W/FPS/3, os=[], c=L.canvas();
c.white();
c.fillStyle=L.gradient(c,W,0,[.2,"blue"],[.8,"skyblue"]]);
c.fill(L.Shape([[0,0,W,W,0]]));
os.push(
  (
    function(){
      let s=floor(W/30), c=L.canvas(), lw=s/5;
      c.strokeStyle="whitesmoke";
      c.lineWidth=lw;
      c.setLineDash([s,s]);
      c.moveTo(0,lw);
      c.lineTo(W,W+lw);
      return function(){ c.clear(); c.lineDashOffset+=speed; c.stroke(); };
    }
  )()
);
for(let i=0;i<2;i++)
os.push(
  (
    function(){
      let s=floor(W/10), c=L.canvas(s), x=0;
      return function(){
        c.clear();
        x=x>0?x-speed:L.rnd(W*2,W);
        c.icon(hit?(x)?128165:127794,x,x);
      };
    }
  )()
);
os.push(
  (
    function(){
      hit=function(t){ return((x-s)<t && t<(x+s) && !(0<i && i<reset)); }
      let s=floor(W/20), c=L.canvas(s), x=W/4;
      c.fillStyle="red";
      c.scale(-1,1);
      c.translate(-x,x);
      c.icon(9975);
      let tgl=1;
      addEventListener("click",()=>i|=1,false);
      let i=0;
      let time=floor(s/speed);
      let reset=3*time;
      function jump(){
        c.clearRect(-s,-s,s*2,s*2);
        c.translate((tgl*=-1)*(s),0);
        c.icon(9975);
      }
      return function(){
        if(i==1 || i==reset){
          jump();
        }
        if(i==reset){
          i=0;
        }
        i && i++;
      };
    }
  )()
);
(
  function animation(){
    os.forEach(o=>o?().());
    requestAnimationFrame(animation);
  }
)()

```



marslander

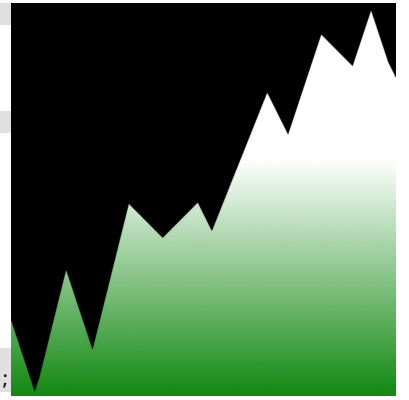
```
//mountain
let cb=L.canvas();
let mountain=[floor(W*0.7)];
(
  function(){
    let tgl=-1, rise, mx=0, mx2=0, y;
    while(mx<W){
      if(mx==mx2){
        rise=(tgl*=-1)*L.rnd(5,1);
        mx2=mx2+L.rnd(floor(W/10), floor(W/20));
      }
      y=mountain[mx]+rise;
      if(y<0 || y>W){mx2=mx; continue;}
      mountain[1+(mx++)]=y;
    }
    cb.stroke(L.shape([L.vertices(mountain)]));
  }
)()
```




```

//mountain gradient
let cb=L.canvas();
let mountain=[floor(W*0.7)];
(
  function(){
    cb.black();
    let tgl=-1, rise, mx=0, mx2=0, y;
    while(mx<W){
      if(mx==mx2){
        rise=(tgl*=-1)*L.rnd(5,1);
        mx2=mx2+L.rnd(floor(W/10), floor(W/20));
      }
      y=mountain[mx]+rise;
      if(y<0 || y>W){mx2=mx;continue;}
      mountain[1+(mx++)]=y;
    }
    cb.fillStyle=L.gradient(cb,W,-W,[[0,"green"],[0.3,"white"]]);
    cb.fill(L.shape([[W,mountain.at(-1),W,W,0,W,0,mountain[0],...L.vertices(mountain)]]));
  }
)()

```



```

//landing site
let cb=L.canvas();
let mountain=[floor(W*0.7)];
(
  function(){
    cb.black();
    let lx=L.rnd(W);
    let tgl=-1, rise, mx=0, mx2=0, y;
    while(mx<W){
      if(mx==lx){
        rise=0;
        mx2=lx+40;
      }else if(mx==mx2){
        rise=(tgl*=-1)*L.rnd(5,1);
        mx2=mx2+L.rnd(floor(W/10), floor(W/20));
      }
      y=mountain[mx]+rise;
      if(y<0 || y>W){mx2=mx; continue;}
      mountain[1+(mx++)]=y;
    }
    cb.fillStyle=L.gradient(cb,W,-W,[[0,"green"],[0.3,"white"]]);
    cb.fill(L.shape([[W,mountain.at(-1),W,W,0,W,0,mountain[0],...L.vertices(mountain)]]));
  }
)()

```



```

//ship
let cb=L.canvas();
let mountain=[floor(W*0.7)];
(
function(){
  cb.black();
  let lx=L.rnd(W);
  let tgl=-1, rise, mx=0, mx2=0, y;
  while(mx<W){
    if(mx==lx){
      rise=0;
      mx2=lx+40;
    }else if(mx==mx2){
      rise=(tgl*=-1)*L.rnd(5,1);
      mx2=mx2+L.rnd(floor(W/10), floor(W/20));
    }
    y=mountain[mx]+rise;
    if(y<0 || y>W){mx2=mx; continue;}
    mountain[1+(mx++)]=y;
  }
  cb.fillStyle=L.gradient(cb,W,-W,[["green"],[0.3,"white"]]);
  cb.fill(L.shape([[W,mountain.at(-1),W,W,0,W,0,mountain[0],...L.vertices(mountain)]]));
})()
let p=floor(W/20);
let ship=L.shape([[0,0,-1,0,-2,-1,0,-4,2,-1,1,0]],p/4);
let cf=L.canvas();
cf.fillStyle=L.gradient(cf,0,-p,[["gold"],[.8,"red"]]);
(
function animation(){
  cf.resetTransform();
  cf.clear();
  cf.translate(200,200);
  cf.fill(ship);
  requestAnimationFrame(animation);
})()
)()

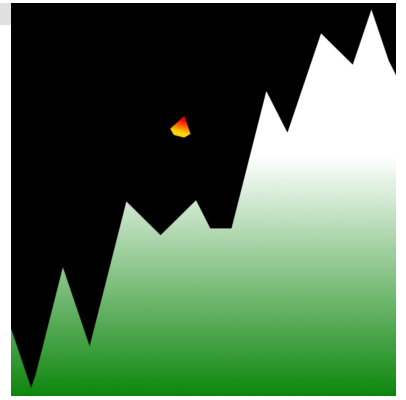
```



```

//move
let cb=L.canvas();
let mountain=[floor(W*0.7)];
(
function(){
  cb.black();
  let lx=L.rnd(W);
  let tgl=-1, rise, mx=0, mx2=0, y;
  while(mx<W){
    if(mx==lx){
      rise=0;
      mx2=lx+40;
    }else if(mx==mx2){
      rise=(tgl*=-1)*L.rnd(5,1);
      mx2=mx2+L.rnd(floor(W/10), floor(W/20));
    }
    y=mountain[mx]+rise;
    if(y<0 || y>W){mx2=mx; continue;}
    mountain[1+(mx++)]=y;
  }
  cb.fillStyle=L.gradient(cb,W,-W,[[0,"green"],[0.3,"white"]]);
  cb.fill(L.shape([[W,mountain.at(-1),W,W,0,W,0,mountain[0],...L.vertices(mountain)]]));
})()
let p=floor(W/20);
let ship=L.shape([[0,0,-1,0,-2,-1,0,-4,2,-1,1,0]],p/4);
let cf=L.canvas();
cf.fillStyle=L.gradient(cf,0,-p,[[.2,"gold"],[.8,"red"]]);
addEventListener("click",move,false);
let x=L.rnd(W),y=0;
let xx=0.0,yy=0.5;
function move({pageX}){
  yy-=0.2;
  if(pageX<(W/2)){xx+=0.1;}
  else {xx-=0.1;}
}
(
function animation(){
  cf.resetTransform();
  cf.clear();
  x+=xx;
  y+=(yy+=0.001);
  cf.translate(x,y);
  cf.rotate(xx/4);
  cf.fill(ship);
  requestAnimationFrame(animation);
}
)()

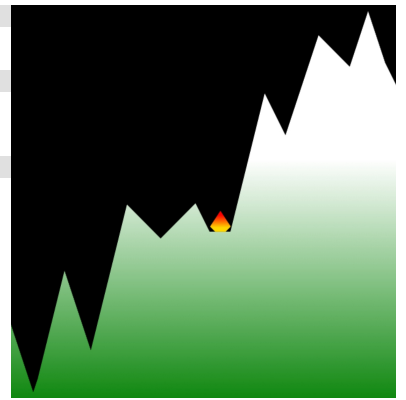
```



```

//land
let cb=L.canvas();
let mountain=[floor(W*0.7)];
let lx;
(
  function(){
    cb.black();
    lx=L.rnd(W);
    let tgl=-1, rise, mx=0, mx2=0, y;
    while(mx<W){
      if(mx==lx){
        rise=0;
        mx2=lx+40;
      }else if(mx==mx2){
        rise=(tgl*=-1)*L.rnd(5,1);
        mx2=mx2+L.rnd(floor(W/10), floor(W/20));
      }
      y=mountain[mx]+rise;
      if(y<0 || y>W){mx2=mx;continue;}
      mountain[1+(mx++)]=y;
    }
    cb.fillStyle=L.gradient(cb,W,-W,[[0,"green"],[0.3,"white"]]);
    cb.fill(L.shape([[W,mountain.at(-1),W,W,0,W,0,mountain[0],...L.vertices(mountain)]]));
  })
let p=floor(W/20);
let ship=L.shape([[0,0,-1,0,-2,-1,0,-4,2,-1,1,0]],p/4);
let cf=L.canvas();
cf.fillStyle=L.gradient(cf,0,-p,[[.2,"gold"],[.8,"red"]]);
addEventListener("click",move,false);
let x=L.rnd(W),y=0;
let xx=0.0,yy=0.5;
function move({pageX}){
  yy-=0.2;
  if(pageX<(W/2)){xx+=0.1;}
  else {xx-=0.1;}
}
(
  function animation(){
    x+=xx;
    y+=(yy+=0.001);
    cf.resetTransform();
    cf.clear();
    cf.translate(x,y);
    cf.rotate(xx/4);
    let landed=(y>mountain[floor(x)]);
    cf.fill(ship);
    if(landed){return;}
    requestAnimationFrame(animation);
  }
)()

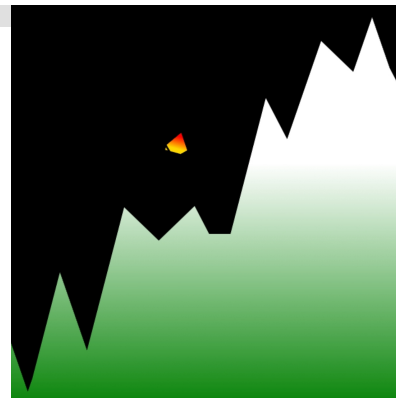
```



```

//rockets
let cb=L.canvas();
let mountain=[floor(W*0.7)];
let lx;
(
function(){
  cb.black();
  lx=L.rnd(W);
  let tgl=-1, rise, mx=0, mx2=0, y;
  while(mx<W){
    if(mx==lx){
      rise=0;
      mx2=lx+40;
    }else if(mx==mx2){
      rise=(tgl*=-1)*L.rnd(5,1);
      mx2=mx2+L.rnd(floor(W/10), floor(W/20));
    }
    y=mountain[mx]+rise;
    if(y<0 || y>W){mx2=mx;continue;}
    mountain[1+(mx++)]=y;
  }
  cb.fillStyle=L.gradient(cb,W,-W,[[0,"green"],[0.3,"white"]]);
  cb.fill(L.shape([[W,mountain.at(-1),W,W,0,W,0,mountain[0],...L.vertices(mountain)]]));
})();
let p=floor(W/20);
let rocketleft=L.shape([[0,0,-1,0,-2,-1,0,-4,2,-1,1,0],[-1.5,0,-2,0,-2,-0.5]],p/4);
let rocketright=L.shape([[0,0,-1,0,-2,-1,0,-4,2,-1,1,0],[1.5,0,2,0,2,-0.5]],p/4);
let rocketnone=L.shape([[0,0,-1,0,-2,-1,0,-4,2,-1,1,0]],p/4);
let ship=rocketnone;
let cf=L.canvas();
cf.fillStyle=L.gradient(cf,0,-p,[[.2,"gold"],[.8,"red"]]);
addEventListener("click",move,false);
let x=L.rnd(W),y=0;
let xx=0.0,yy=0.5;
function move({pageX}){
  yy-=0.2;
  if(pageX<(W/2)){xx+=0.1;ship=rocketleft;}
  else {xx-=0.1;ship=rocketright;}
  setTimeout(()=> ship=rocketnone,200);
}
(
function animation(){
  x+=xx;
  y+=(yy+=0.001);
  cf.resetTransform();
  cf.clear();
  cf.translate(x,y);
  cf.rotate(xx/4);
  let landed=(y>mountain[floor(x)]);
  cf.fill(ship);
  if(landed){return;}
  requestAnimationFrame(animation);
}
)()

```



```

//stars and crash
let cb=L.canvas();
let mountain=[floor(W*0.7)];
let lx;
(
function(){
  cb.black();
  cb.fillStyle="white";
  for(let stars=W*W/2000;stars>0;stars--){cb.fillRect(L.rnd(W),L.rnd(W),1,1);}
  lx=L.rnd(W);
  let tgl=-1, rise, mx=0, mx2=0, y;
  while(mx<W){
    if(mx==lx){
      rise=0;
      mx2=lx+40;
    }else if(mx==mx2){
      rise=(tgl*=-1)*L.rnd(5,1);
      mx2=mx2+L.rnd(floor(W/10), floor(W/20));
    }
    y=mountain[mx]+rise;
    if(y<0 || y>W){mx2=mx;continue;}
    mountain[1+(mx++)]=y;
  }
  cb.fillStyle=L.gradient(cb,W,-W,[0,"green"],[0.3,"white"]);
  cb.fill(L.shape([[W,mountain.at(-1),W,W,0,W,0,mountain[0],...L.vertices(mountain)]]));
})()
let p=floor(W/20);
let rocketleft=L.shape([[0,0,-1,0,-2,-1,0,-4,2,-1,1,0],[-1.5,0,-2,0,-2,-0.5]],p/4);
let rocketright=L.shape([[0,0,-1,0,-2,-1,0,-4,2,-1,1,0],[1.5,0,2,0,2,-0.5]],p/4);
let rocketnone=L.shape([[0,0,-1,0,-2,-1,0,-4,2,-1,1,0]],p/4);
let ship=rocketnone;
let cf=L.canvas();
cf.fillStyle=L.gradient(cf,0,-p,[.2,"gold"],[.8,"red"]);
addEventListener("click",move,false);
let x=L.rnd(W),y=0;
let xx=0.0,yy=0.5;
function move({pageX}){
  yy-=0.2;
  if(pageX<(W/2)){xx+=0.1;ship=rocketleft;}
  else {xx-=0.1;ship=rocketright;}
  setTimeout(()=> ship=rocketnone,200);
}
(
function animation(){
  x+=xx;
  y+=(yy+=0.001);
  cf.resetTransform();
  cf.clear();
  cf.translate(x,y);
  cf.rotate(xx/4);
  let landed=(y>mountain[floor(x)]);
  let crash=(yy>0.5 || x<lx || x>(lx+40));
  if(landed && crash){ ship=L.shape([Array.from({length:40},()=>L.rnd(p*2)-p)]); }
  cf.fill(ship);
  if(landed){return;}
  requestAnimationFrame(animation);
}
)()

```

