MinisterSpillet_F_RDIG_2

```
int state;

PImage christiansborg;
PImage MetteFr;
PImage DanJ;




void setup() {
  //size(800, 600);
  size(1000, 800);
  //fullScreen();

  christiansborg = loadImage("Christiansborg1.jpg");
  MetteFr = loadImage("MetteF.jpg");
  DanJ = loadImage("DanJ.jpg");




  //ad 5 journalister
  for (int counter = 0; counter < 5; counter += 1) {
    JournalistList.add(new Journalist());
  }

  //ad Mette
  MetteF = new Mette();

  //ad Koteletter
  for (int counter = 0; counter < 3; counter += 1) {
    KoteletList.add(new Kotelet());
  }

  //ad frikadeller


  for (int counter = 0; counter < 3; counter += 1) {
    FrikadelleList.add(new Frikadelle());
  }

  //ad Vindmøller
  for (int counter = 0; counter < 3; counter += 1) {
    VindmolleList.add(new Vindmolle());
  }
  //ad ad Klima Dan
  KlimaD = new KlimaDan();



  state = 10;
}

void draw() {
  if (state == 10) {
    Forside();
  }

  if (state == 20) {
    MetteRegler();
  }

  if (state == 30) {
    MetteSpillet();
  }

  if (state == 40) {
    MetteVinder();
  }

  if (state == 50) {
    MetteTaber();
  }

  if (state == 60) {
    KlimaDanRegler();
  }

  if (state == 70) {
    KlimaDanSpillet();
  }

  if (state == 80) {
    KlimaDanVinder();
  }
  if (state == 90) {
    KlimaDanTaber();
  }
```

```
}
```

Forside

```
void Forside() {
  //background(86, 214, 183);

  rectMode(CENTER);
  imageMode(CENTER);

  image(christiansborg, width/2, height/2, width, height);
  textAlign(CENTER);

  textSize(48);
  fill(255, 0, 0);
  text("Minister-spillet", width/2, 80);
  textAlign(LEFT);
  textSize(30);
  String spilForklaring = "Vil du gerne prøve at være minister for et lille øjeblik? Så giv det et forsøg som Mette Fredriksen eller KlimaDan";
  String spilVaelg1 = "Tryk på Mette for Mette-spillet";
  String spilVaelg2 = "Tryk Dan for KlimaDan-spillet";
  text(spilForklaring, width/4, 530, 400, 800);
  textSize(24);
  fill(70);
  text(spilVaelg1, width/4*3.2, 570, 400, 800);
  text(spilVaelg2, width/4*3.2, 630, 400, 800);

  stroke(255, 0, 0);
  noFill();

  //Buttons to choose game
  image(MetteFr, width/4, height/4*3);
  image(DanJ, width/4*3, height/4*3);

  if(dist(width/4, height/4*3, mouseX, mouseY)<MetteFr.width){
     rect(width/4, height/4*3, 201, 201);
     if(mousePressed){
      state = 20;
     }
  }

  if(dist(width/4*3, height/4*3, mouseX, mouseY)<MetteFr.width){
     rect(width/4*3, height/4*3, 201, 201);
```

```
    if(mousePressed){
     state = 60;
    }
  }




}
```

Keys

```
void keyPressed() {
  if (key == &apos;w&apos; || key == &apos;W&apos;) {
    if (state == 30) {
      MetteF.moveUp();
    }
    if (state == 70) {
      KlimaD.moveUp();
    }
  }

  if (key == &apos;s&apos; || key == &apos;S&apos;) {
    if (state == 30) {
      MetteF.moveDown();
    }
    if (state == 70) {
      KlimaD.moveDown();
    }
  }

  if (key == &apos;a&apos; || key == &apos;A&apos;) {
    if (state == 30) {
      MetteF.moveLeft();
    }
    if (state == 70) {
      KlimaD.moveLeft();
    }
  }

  if (key == &apos;d&apos; || key == &apos;D&apos;) {
    if (state == 30) {
      MetteF.moveRight();
    }
    if (state == 70) {
      KlimaD.moveRight();
    }
  }

  if (key == &apos; &apos;) {

    if (state == 30) {
      Bullet newBullet = new Bullet();
      newBullet.x = MetteF.x;
      newBullet.y = MetteF.y;
      bulletList.add(newBullet);
    }
    if (state == 70) {
      KlimaBullet newKlimaBullet = new KlimaBullet();
      newKlimaBullet.x = KlimaD.x;
      newKlimaBullet.y = KlimaD.y;
      KlimaBulletList.add(newKlimaBullet);
    }
  }
}

void keyReleased() {

  if (key == &apos;w&apos; || key == &apos;s&apos; || key == &apos;W&apos; || key ==
&apos;S&apos;) {
    if (state == 30) {
      MetteF.stopYmovement();
    }
    if (state == 70) {
      KlimaD.stopYmovement();
    }
  }



  if (key == &apos;a&apos; || key == &apos;A&apos;|| key == &apos;d&apos; || key ==
&apos;D&apos;) {
    if (state == 30) {
      MetteF.stopXmovement();
    }
    if (state == 70) {
      KlimaD.stopXmovement();
    }
  }
}
```

KlimaDanSpil

```
ArrayList<Kotelet> KoteletList = new ArrayList<Kotelet>();
ArrayList<Frikadelle> FrikadelleList = new ArrayList<Frikadelle>();
ArrayList<Vindmolle> VindmolleList = new ArrayList<Vindmolle>();
KlimaDan KlimaD;
ArrayList<KlimaBullet> KlimaBulletList = new ArrayList<KlimaBullet>();
int KlimaPoint = 0;
int KlimaLiv = 3;



void KlimaDanRegler() {
  background (0, 200, 0);
  textAlign(CENTER);
  rectMode(CENTER);
  textSize(48);
  fill(255);
  text("KlimaDan-spillet", width/2, 80);
  textSize(24);
  String KlimaDanForklaring = "Du er den grønneste klimaminister Danmark har haft. Vi skal have massere af svinekød og olie op af Nordsøen, for vi må ikke gøre landbruget sure";
  String KlimaDanRegler = "Du styrer således: a = venstre, d = højre, w = op, s = ned, mellemrum = du skyder op ad. Du har 3 liv, og mister et hver gang du støder ind i vindmøller. Du får point når du nedskyder vindmøller og når du spiser svinekød. Du vinder når du har fået 10 point. Du forlader spillet og vender tilbage til forsiden ved at trykke på P";
  text(KlimaDanForklaring, width/2, 330, 800, 400);
  text(KlimaDanRegler, width/2, 530, 800, 400);
  fill(70);
  text("Tryk mellemrum for at starte spillet", width/2, 600);

  if (keyPressed && key ==&apos; &apos;) {
     state = 70;
  }
}

void KlimaDanVinder() {
  background (0, 200, 0);
  textAlign(CENTER);
  rectMode(CENTER);
  textSize(48);
  fill(255);
  text("KlimaDan-spillet", width/2, 80);
  String KlimaDVinder = "Du har indsamlet nok svinekød og har fået 10 point. Tillykke du er en rigtig god klimaminsiter!";
  textSize(30);
  text(KlimaDVinder, width/2, 430, 800, 400);
  fill(0);
  textSize(24);
  fill(70);
  text("Tryk P for at vende tilbage til start og vælg et spil", width/2, 500);

  //Tilbage til start
  if (keyPressed && key == &apos;p&apos; || key == &apos;P&apos;) {
    state = 10;
    KlimaDanReset();
  }
}

void KlimaDanTaber() {
  background (150);
  textAlign(CENTER);
  rectMode(CENTER);
  textSize(48);
  fill(255);
  text("KlimaDan-spillet", width/2, 80);
  textSize(30);
  String RossenSkuffet = "Landbruget er nu blevet rigtig sure";
  String RossenSkuffet1 = "Skynd dig at købe en masse svinekød, så landbruget ikke stemmer blåt næste gang!";
  text(RossenSkuffet, width/2, 430, 800, 400);
  text(RossenSkuffet1, width/2, 530, 800, 400);
  textSize(24);
  fill(70);
  text("Tryk P for at vende tilbage til start og vælg et spil", width/2, 580);

  //tilbage til start
  if (keyPressed && key == &apos;p&apos; || key == &apos;P&apos;) {
    state = 10;
    KlimaDanReset();
  }
}

void KlimaDanReset() {
  KlimaLiv = 3;
  KlimaPoint = 0;
  KlimaBulletList.remove(KlimaBulletList);
}
```

```
void KlimaDanSpillet() {
  background (0, 200, 0);
  textAlign(CENTER);
  textSize(30);
  fill(255);
  text("Liv : "+ KlimaLiv, width/2-100, 25);
  text("Point : " + KlimaPoint, width/2+100, 25);


  //Show all Koteletter from arraylist
  for (int index = 0; index<KoteletList.size(); index ++) {
    Kotelet tempKotelet = KoteletList.get(index);

    float tempDist = dist(tempKotelet.x, tempKotelet.y, KlimaD.x, KlimaD.y);

    if (tempDist <KlimaD.shipHeight/2+tempKotelet.size/2) {
      println(tempDist);
      KoteletList.remove(index);

      KlimaPoint +=1;
      KoteletList.add(new Kotelet ());
    }
    tempKotelet.move();

    tempKotelet.show();
  }

  // Show frikadeller from arraylist
  for (int index = 0; index<FrikadelleList.size(); index ++) {
    Frikadelle tempFrikadelle = FrikadelleList.get(index);

    float tempDist = dist(tempFrikadelle.x, tempFrikadelle.y, KlimaD.x, KlimaD.y);

    if (tempDist<KlimaD.shipHeight/2 + tempFrikadelle.size/2) {
      FrikadelleList.remove(index);

      KlimaPoint +=1;
      FrikadelleList.add(new Frikadelle());
    }
    tempFrikadelle.move();
    tempFrikadelle.show();
```

```
}

//Show all vindmoller from arraylist
for (int index=0; index<VindmolleList.size(); index ++) {
  Vindmolle tempVindmolle = VindmolleList.get(index);

  float tempDist = dist(tempVindmolle.x, tempVindmolle.y, KlimaD.x, KlimaD.y);

  if (tempDist<KlimaD.shipHeight/2 + tempVindmolle.size/2) {
    VindmolleList.remove(index);

    KlimaLiv -=1;
    KlimaD.x=width/2;
    KlimaD.y=height/2;
    VindmolleList.add(new Vindmolle());
  }
  tempVindmolle.move();
  tempVindmolle.show();
}



//Show KlimaDan
KlimaD.update();

//Show bullets

for (int indexBullet = 0; indexBullet < KlimaBulletList.size(); indexBullet++) {
  KlimaBullet tempBullet = KlimaBulletList.get(indexBullet);
  tempBullet.movement();



  //Koteletter
  for (int indexKotelet = 0; indexKotelet < KoteletList.size(); indexKotelet ++) {
    Kotelet tempKotelet = KoteletList.get(indexKotelet);
    float distanceBulletKotelet = dist(tempKotelet.x, tempKotelet.y, tempBullet.x, tempBullet.y);


    if (distanceBulletKotelet <tempBullet.size/2+tempKotelet.size/2)
    {

      KoteletList.remove(indexKotelet);
      KlimaBulletList.remove(indexBullet);
      KlimaPoint -=1;
      KlimaLiv -=1;
```

```
        KoteletList.add(new Kotelet ());
      }
    }

    //Frikadeler
    for (int indexFrikadelle = 0; indexFrikadelle < FrikadelleList.size(); indexFrikadelle ++) {
      Frikadelle tempFrikadelle = FrikadelleList.get(indexFrikadelle);

      float distanceBulletFrikadelle = dist(tempFrikadelle.x, tempFrikadelle.y, tempBullet.x,
tempBullet.y);



      if (distanceBulletFrikadelle <tempBullet.size/2+tempFrikadelle.size/2) {
        FrikadelleList.remove(indexFrikadelle);
        KlimaBulletList.remove(indexBullet);
        KlimaPoint -=1;
        KlimaLiv -=1;
        FrikadelleList.add(new Frikadelle ());
      }
    }

    //Vindmøller
    for (int indexVindmolle = 0; indexVindmolle < VindmolleList.size(); indexVindmolle ++) {
      Vindmolle tempVindmolle = VindmolleList.get(indexVindmolle);
      float distanceBulletVindmolle = dist(tempVindmolle.x, tempVindmolle.y, tempBullet.x,
tempBullet.y);

      if (distanceBulletVindmolle < tempBullet.size/2 + tempVindmolle.size/2) {
        VindmolleList.remove(indexVindmolle);
        KlimaBulletList.remove(indexBullet);
        KlimaPoint += 1;
        VindmolleList.add(new Vindmolle());
      }
    }

    tempBullet.show();

    if (tempBullet.y<0+tempBullet.size/2 && KlimaBulletList.size()<1) {
      KlimaBulletList.remove(indexBullet);
    }
  }

  //KlimaDan Vinder
  if (KlimaPoint ==10) {
    state = 80;
  }

  //KlimaDan taber
  if (KlimaLiv==0) {
    state = 90;
  }



  //Tilbage til start
  if (keyPressed && key == &apos;p&apos; || key == &apos;P&apos;) {
    state = 10;
    KlimaDanReset();
  }
}
```

KlimaDan_Komponents

```
class Kotelet {
  float x, y, vX, vY;
  float speed = 3;
  int size = floor(random(30, 60));
  float greyColor= random(10, 100);
  PImage kotelet;



  Kotelet() {
    x = random(width);
    y = random(height);

    vX = random(-speed, speed);
    vY = random(-speed, speed);
    kotelet = loadImage("Kotelet.png");
    kotelet.resize(size,size);
  }

  void move() {
    x += vX;
    y += vY;

    if (x>width + size/2) {
      x=0-size/2;
    }

    if (x<0-size/2) {
      x=width+size/2;
    }

    if (y>height+size/2) {
      y=0-size/2;
    }

    if (y<0-size/2) {
      y=height+size/2;
    }
  }

  void show() {
    //fill(greyColor);
    //ellipse(x, y, size, size);
    image(kotelet, x, y, size, size);
  }
}

class Frikadelle {
  float x, y, vX, vY;
  float speed = 3;
  int size = floor(random(30, 60));
  float greyColor= random(10, 100);
  PImage frikadelle;



  Frikadelle() {
    x = random(width);
    y = random(height);

    vX = random(-speed, speed);
    vY = random(-speed, speed);
    frikadelle = loadImage("frikadeller.png");
    frikadelle.resize(size,size);
  }

  void move() {
    x += vX;
    y += vY;

    if (x>width + size/2) {
      x=0-size/2;
    }
    if (x<0-size/2) {
      x=width+size/2;
    }

    if (y>height+size/2) {
      y=0-size/2;
    }

    if (y<0-size/2) {
      y=height+size/2;
    }
  }

  void show() {
    //fill(greyColor);
    //ellipse(x, y, size, size);
    image(frikadelle, x, y, size, size);
  }
}

class Vindmolle {
  float x, y, vX, vY;
  float speed = 3;
  int size = floor(random(30, 60));
  float greyColor= random(10, 100);
  PImage vindmolle;

  Vindmolle() {
    x = random(width);
    y = random(height);

    vX = random(-speed, speed);
    vY = random(-speed, speed);
    vindmolle = loadImage("Vindmolle.png");
    vindmolle.resize(size,size);
  }

  void move() {
    x += vX;
    y += vY;

    if (x>width + size/2) {
      x=0-size/2;
    }

    if (x<0-size/2) {
      x=width+size/2;
    }

    if (y>height+size/2) {
      y=0-size/2;
    }

    if (y<0-size/2) {
      y=height+size/2;
    }
  }

  void show() {
    //fill(greyColor);
    //ellipse(x, y, size, size);
    image(vindmolle, x, y, size, size);
  }
}

class KlimaBullet {

  float x, y;
  float speed =17;
  float size = 5;
  float vY;


  KlimaBullet() {

    x = 30;
    y = 600;
    vY = -speed;
  }

  void show() {
    fill(255, 222, 8);
    ellipse(x, y, size, size);
  }
  void movement() {
    y = y + vY;
  }
}

class KlimaDan {

  float x, y, vX, vY;
  float speed = 3;
  float shipWidth = 50;
  float shipHeight = 50;
  PImage KlimaDan;



  KlimaDan() {

    x = width/2;
    y = height/2;
    KlimaDan =  loadImage("KlimaDan.png");
  }

  void show() {

    imageMode(CENTER);
    image(KlimaDan, x, y);
  }

  void update() {
    movement();
    show();
  }

  //Movement of Klima Dan
  void movement() {
    x+= vX;
    y+=vY;

    if (x>width + shipWidth/2) {
      x=0-shipWidth/2;
    }

    if (x<0-shipWidth/2) {
      x=width+shipWidth/2;
    }

    if (y>height+shipHeight/2) {
      y=0-shipHeight/2;
    }

    if (y<0-shipHeight/2) {
      y=height+shipHeight/2;
    }
  }

  void stopYmovement() {
    vY=0;
  }

  void stopXmovement() {
    vX = 0;
  }


  void moveUp() {
    vY = -1*speed;
  }

  void moveDown() {
    vY = speed;
  }

  void moveLeft() {
    vX = -1 * speed;
  }

  void moveRight() {
    vX=speed;
  }
}
```

MetteSpillet

```
ArrayList<Journalist> JournalistList = new ArrayList<Journalist>();
Mette MetteF;
ArrayList<Bullet> bulletList = new ArrayList<Bullet>();
int point = 0;
int liv = 3;
```

```
void MetteRegler() {
  background (255, 0, 0);
  textAlign(CENTER);
  rectMode(CENTER);
  textSize(48);
  fill(255);
  text("Mette-spillet", width/2, 80);
  textSize(24);
  String MetteForklaring = "Du vil ikke give interview, fordi journalister er alt for kritiske over for din politik. I stedet skal du undgå dem og skyde dem ned, så de ikke kan stille kritiske spørgsmål til din eller din regerings gode politik";
  String MetteRegler = "Du styrer således: a = venstre, d = højre, w = op, s = ned, mellemrum = du skyder op ad. Du har 3 liv, og mister et hver gang du støder ind i en journalist. Du får point ved at skyde journalister. Du vinder når du har fået 5 point. Du forlader spillet og vender tilbage til forsiden ved at trykke på P";
  text(MetteForklaring, width/2, 330, 800, 400);
  text(MetteRegler, width/2, 530, 800, 400);
  fill(70);
  text("Tryk mellemrum for at starte spillet", width/2, 600);

  if (keyPressed && key ==&apos; &apos;) {
    state = 30;
  }
}

void MetteVinder() {
  background (255, 0, 0);

  textAlign(CENTER);
  rectMode(CENTER);
  textSize(48);
  fill(255);
  text("Mette-spillet", width/2, 80);
  String RossenErStolt = "Du har undgået 5 journalister. Martin Rossen er stolt af dig!";
  textSize(30);
  text(RossenErStolt, width/2, 430, 800, 400);
  fill(0);
  textSize(24);
  fill(70);
  text("Tryk P for at vende tilbage til start og vælg et spil", width/2, 500);

  //Tilbage til start
  if (keyPressed && key == &apos;p&apos; || key == &apos;P&apos;) {
    state = 10;
    MetteReset();
  }
}

void MetteTaber() {
  background (150);
  textAlign(CENTER);
  rectMode(CENTER);
  textSize(48);
  fill(255);
  text("Mette-spillet", width/2, 80);
  textSize(30);
  String RossenSkuffet = "Martin Rossen er ikke sur, Martin Rossen er skuffet!";
  String RossenSkuffet1 = "Du har givet for mange interview til kritiske journalister, skynd dig tilbage til statsministerieret og spis en rugbrød med leverpostej og lidt hjemmebag";
  text(RossenSkuffet, width/2, 430, 800, 400);
  text(RossenSkuffet1, width/2, 530, 800, 400);
  textSize(24);
  fill(70);
  text("Tryk P for at vende tilbage til start og vælg et spil", width/2, 580);

  //tilbage til start
  if (keyPressed && key == &apos;p&apos; || key == &apos;P&apos;) {
    state = 10;
    MetteReset();
  }
}

void MetteReset() {
  liv = 3;
  point = 0;
  bulletList.remove(bulletList);
}




void MetteSpillet() {
  background (255, 0, 0);
  textAlign(CENTER);
  textSize(30);
  fill(255);
  text("Liv : "+ liv, width/2-100, 25);
  text("Point : " + point, width/2+100, 25);


  //Show all journalister from arraylist
  for (int index = 0; index<JournalistList.size(); index ++) {
    Journalist tempJournalist = JournalistList.get(index);

    float tempDist = dist(tempJournalist.x, tempJournalist.y, MetteF.x, MetteF.y);

    if (tempDist <MetteF.shipHeight/2 + tempJournalist.size/2)
    {

      MetteF.x=width/2;
      MetteF.y=height/2;
      liv -=1;
    }

    //line(tempAsteroid.x, tempAsteroid.y, player1.x, player1.y);

    tempJournalist.move();
    tempJournalist.show();
  }





  //Show ship
  MetteF.update();

  //Show bullets

  for (int indexBullet = 0; indexBullet < bulletList.size(); indexBullet++) {
    Bullet tempBullet = bulletList.get(indexBullet);
    tempBullet.movement();

    for (int indexJournalist = 0; indexJournalist < JournalistList.size(); indexJournalist ++) {
      Journalist tempJournalist = JournalistList.get(indexJournalist);
      float distanceBulletJournalist = dist(tempJournalist.x, tempJournalist.y, tempBullet.x, 
tempBullet.y);


      if (distanceBulletJournalist <tempBullet.size/2+tempJournalist.size/2)
      {
        JournalistList.remove(indexJournalist);
        bulletList.remove(indexBullet);
        point +=1;
        JournalistList.add(new Journalist ());
      }
      //line(tempBullet.x, tempBullet.y, tempAsteroid.x, tempAsteroid.y);
    }

    tempBullet.show();

    if (tempBullet.y<0+tempBullet.size/2 && bulletList.size()>1) {
      println(bulletList.size());
      bulletList.remove(indexBullet);
    }
  }

  //Mette taber
  if (point ==5) {
    state =40;
  }

  if (liv==0) {
    state = 50;
  }



  //Tilbage til start
  if (keyPressed && key == &apos;p&apos; || key == &apos;P&apos;) {
    state = 10;
    MetteReset();
  }
}
```

Mette_Komponents

```
class Journalist {
  float x, y, vX, vY;
  float speed = 3;
  int size = floor(random(30, 50));
  float greyColor= random(10, 100);
  PImage jour;

  Journalist() {
    x = random(width);
    y = random(height);

    vX = random(-speed, speed);
    vY = random(-speed, speed);
    jour = loadImage("Journalist2.png");
    jour.resize(size, size);
  }

  void move() {
    x += vX;
    y += vY;

    if (x>width + size/2) {
      x=0-size/2;
    }

    if (x<0-size/2) {
      x=width+size/2;
    }

    if (y>height+size/2) {
      y=0-size/2;
    }

    if (y<0-size/2) {
      y=height+size/2;
    }
  }

  void show() {
    //fill(greyColor);
    //ellipse(x, y, size, size);
    image(jour, x, y);
  }
}

class Bullet {

  float x, y;
  float speed =17;
  float size = 5;
  float vY;
  Bullet() {

    x = 30;
    y = 600;
    vY = -speed;
  }

  void show() {
    fill(255, 222, 8);
    ellipse(x, y, size, size);
  }



  void movement() {
    y = y + vY;
  }
}

class Mette {

  float x, y, vX, vY;
  float speed = 3;
  float shipWidth = 50;
  float shipHeight = 50;
  PImage Mette;

  Mette() {

    x = width/2;
    y = height/2;

    Mette =  loadImage("Mette1.png");
  }

  void show() {

    imageMode(CENTER);
    image(Mette, x, y);
  }

  void update() {
    movement();
    show();
}

//Movement of Mette
void movement() {
    x+= vX;
    y+=vY;

    if (x>width + shipWidth/2) {
        x=0-shipWidth/2;
    }

    if (x<0-shipWidth/2) {
        x=width+shipWidth/2;
    }

    if (y>height+shipHeight/2) {
        y=0-shipHeight/2;
    }

    if (y<0-shipHeight/2) {
        y=height+shipHeight/2;
    }
}

void stopYmovement() {
    vY=0;
}

void stopXmovement() {
    vX = 0;
}



void moveUp() {
    vY = -1*speed;
}

void moveDown() {
    vY = speed;
}

void moveLeft() {
    vX = -1 * speed;
}
```

```
  void moveRight() {
      vX=speed;
  }
}
```