Living Code & Pseudo Code:

```
/**
 * final project - plane shooting game
 * game name: plane shooter
 * instruction:
 * after loading successfully, clicking the start button to start
 * using the mouse to control the shooter
 * mouseclicking to shoot the bullet
 * getting 10 points and 1 more bullet for reward when hitting a plane
 * the more points players get, the faster the planes fly
 * pressing the key 'i' and 'd' to increase or decrease the speed the planes
 * game's over when running out of bullets
 * pressing the key 'r' to restart the game
 */

// global variables

// importing the background music file
import processing.sound.*;
SoundFile file;
// the array of stars
ArrayList<PVector> star = new ArrayList();
// the remaining bullets
// there are 10 bullets in total
int remain=10;
// the score players get
// the number of hits
// initializing the score and hit
int score=0;
int hit=0;
// each page of the game
int page=0;
// the mark to determine whether the planes are hit, bullets are fired
int mark;
// the duration of the loading bar
int duration=5;
// the percent of loading bar
int percent=0;
// text fonts
PFont f1,f2;
// background image
PImage bg;
```

```
// plane icon
PShape plane;
// shooter icon
PShape shooter;

// the array of shooter
// moving area
shooter s=new shooter((int)random(20,500));
// the array of bullets
// the number of bullets
bullet[] b=new bullet[10];
// the array of planes
// the moving area
plane p=new plane(0,(int)random(20,200));

// setup functions
void setup()
{ // the size the window
  size(700,500);
  background(255);
  smooth();
  // for each star, they will display at the given position
  for (int i=0;i<1000;i++) {
  PVector P = new PVector(random(2*width), random(2*height));
  // the target vector is P
  star.add(P);
  }

  // adding text fonts
  f1 = createFont("AvenirNext-Bold", 30);
  f2 = createFont("Avenir-Heavy", 15);

  // inserting images
  bg = loadImage("bg.jpg");
  bg.resize(width,height);

  // loading icons
  plane = loadShape("plane.svg");
  shooter = loadShape("shooter.svg");

  // importing sound files
  file = new SoundFile(this,"creamydream.mp3");
  // playing the music
  file.play();
  // changing the volume of the sound
  file.amp(0.3);

  // without running out of bullets
  // displaying the properties of the array of bullets
  for(int i=0;i<10;i++)
  {  b[i]=new bullet();
  }
}

// draw functions
void draw()
  // 3 pages in total - prestart, game and gameover page
  { switch(page) {

  // prestart page
  case 0:
  background(26, 51, 101);
  image(bg, 0, 0);

  // moving star
  noStroke();
  fill(-1);
  for (int i=0;i<star.size();i++) {
  // gets the star with the drawn shape
  PVector P = star.get(i);
  // stars will move with the mouse
  PVector M = new PVector(pmouseX-mouseX, pmouseY-mouseY);
  // the target vector is M
  P.add(M);
  // floating area
  // sets the distance of the mouse between two positions
  float d = dist(P.x, P.y, width/2, height/2);
  // remaps the number from one range to another
  d = map(d, 0, width/2, 0, 3);
  // the drawn shape
  ellipse(P.x, P.y, d, d);
  }

  // the name of the game
  fill(255);
  textFont(f1);
```

```
textSize(45);
text("Plane Shooter", 200, 150);

// loading
stroke(255);
textSize(13);
textFont(f2);
fill(255);
text("LOADING", 250, 220);
// loading bar
fill(0,0,0,0);
strokeWeight(2);
stroke(230);
rect(250, 230, 200, 20);

fill(255);
textSize(15);
// percent of loading
text(percent+"%", 330, 220);
// progressing bar
fill(255);
noStroke();
rect(251, 231, percent*2,19);

// the duration of loading bar when percent is less than 100%
if(frameCount % duration/2==0 && percent<100){
  percent++; // percents increase gradually
  }
 // when percent is 100%
 // it loads successfully
 if(percent==100){
  // there will be a start button shown under the loading bar
  fill(0,0,0,0);
  stroke(255);
  strokeWeight(5);
  rect(250, 310, 205, 80, 15);
  fill(255);
  textSize(45);
  textFont(f1);
  text("Start !", 310, 360);

 }
break;
```

```
// game page
case 1:

background(255);
image(bg, 0, 0);

// display of shooter and planes
s.display();
p.fly();
p.display();

// without running out of bullets
for(int i=0;i<10;i++)
{ // displaying the setup of planes crashed
  p.crash(b[i]);
  // marks that bullets hit planes
if(b[i].mark==0)
 { // when bullets hit planes, it will show the display and fly setups of bullets
   // views properties in the class of bullet
   b[i].fly();
   b[i].display();
  }
}

// when hitting one plane
if(hit==1)
{ // resets the number of hits
  hit=0;
  // without running of bullets
  for(int i=0;i<10;i++)
  { // bullets hit planes
    if(b[i].mark==0)
    { // displaying the properties of bullets
      // remaining bullets will be incremented
      b[i]=new bullet();
      remain++;
      break;
    }
  }
}

// players can view their remaining bullets and score on upper right corner
```

```
  fill(255);
  textSize(20);
  textFont(f2);
  text("remain:"+remain,610,40);
  text("score:"+score,610,65);

  // when runing out of bullets
  // game's over
  // gameover page
  if(remain==0)
  { background(153, 0, 0);
    image(bg, 0, 0);
    fill(255);
    textSize(35);
    textFont(f1);
    text("Game Over :(",250,150);
    text("You run out of bullets!",190,220);
    text("Your final score is: " +score,190,290);
    text("Press R to restart",220,360);
  }
 }
}

// mouseclicked functions
void mouseClicked(){
  // it will switch the pages from prestart page to gameover page
  // on the 1st(prestart) page
  if (page ==0) {
    // sets the click area to the position of start button
    // when clicking the start button, the page will switch to the next one
    if (mouseX > 250 && mouseX < 455 && mouseY > 310 && mouseY < 390 ) {
      page++;
    }
  }
}

// keypressed functions
void keyPressed()
{ // uses the key 'r' to restart the game
  if(key=='r')
  {  // initializes the number of remaining bullets and score
     remain=10;
     score=0;
```

```
   // without running of bullets
   for(int i=0;i<10;i++)
    // resets the properties and speed of bullets
    b[i]=new bullet();
    p.speed=3;
 }
 // uses the key "i" to increase the speed of planes
 if(key=='i')
    p.speed+=3;
 // uses the key "d" to decrease the speed of planes
 if(key=='d')
 { if(p.speed<=3) return;
  else p.speed-=3;
 }
}

// mousepressed functions
void mousePressed()
{ // initializes the number of clicks
 int i=0;
  // when mouse is pressed less than 10 times, bullets can be fired
  for(i=0;i<10;i++)
  {if(b[i].mark==1)
   break;
  }
  // if pressed 10 times
  if(i==10) return;
  // there's a removable shooter at the bottom
  // the number of remaining bullets will decrease
  s.fire(b[i]);
  remain--;
}

// properties of plane
class plane
{ // the position of x and y
 int x;
 int y;
 // determines whether the planes are hit
 int mark;
 // floating speed
 int speed;
```

```
// constructor
plane(int x,int y)
{ // draws planes at such position
  this.x=x;
  this.y=y;
  speed=3; // sets the speed
  mark=0;  // planes can be hit (have not been hit)
}

// the setup of planes flying in the sky
void fly()
{   // when planes are not hit
    // they fly from left to right at the given speed
    if(mark==0)
    { x+=speed;
      // when planes fly out of the window
      // there will be another plane flying from left to right again
      if(x>=700)
      { // floating area
        x-=700;
        y=(int)random(100,300);
      }
    }
   // when planes are hit
   if(mark==1)
   {  // resets the planes to make them can still be hit
      mark=0;
      x=0;
      speed++;
   }
}

// the setup when planes are hit
void crash(bullet b)
{ // when the distance between the coordinates of planes and bullets are less than 50
  // it's determined that planes are hit
  // score will be incremented 10 each time
  if(dist(x,y,b.x,b.y)<50)
  { score+=10;
    mark=1;
    hit=1;
  }
}
```

```
  // the display of planes
  void display()
  { // in the window
    if(x<700)
    { // plane icon
      shape(plane, x+15,y+20,60,80);
    }
  }
}

// properties of bullets
class bullet
{ // constructor
  int x;
  int y;
  int mark;

  bullet()
  { x=700;
    y=500;
    mark=1; // can be fired
  }

  // the setup when bullets are fired
  void fire(int x,int y)
  { // constructor
    this.x=x;
    this.y=y;
    mark=0; // fired
  }

  // the setup when bullets fly in the sky
  void fly()
  { // when fired, the bullet will shoot from bottom at the given speed
    if(mark==0) y-=5;
  }

  // the display of bullets
  void display()
  { // at the given area
    if(y<500)
    { // the shape of bullet
```

```
      fill(240);
      noStroke();
      ellipse(x,y,20,20);
    }
  }
}

// properties of shooter
class shooter
{ // constructor
  int x;
  shooter(int x)
  { this.x=x;  }

  // display of shooter
  void display()
  { fill(0);
    noStroke();
    // shooter icon
    // it will move with the mouse
    shape(shooter, mouseX,440,80,90);
  }

  // when firing bullets, bullets will be out at the given position
  void fire(bullet b)
  { b.fire(mouseX+35,450);
  }
}
```