```
let dices = [];
let scale = 100;
let rollbutton;
let buttons = [];
let values;
let rolls = 0;
let acesscore = 0;
let twosscore = 0;
let threesscore = 0;
let foursscore = 0;
let fivesscore = 0;
let sixesscore = 0;
let lefttotalscore = 0;
let bonusscore = 0;
let grandlefttotalscore = 0;
let toakscore = 0;
let foakscore = 0;
let fullhousescore = 0;
let smstraightscoret = false;
let smstraightscore = 0;
let lgstraightscoret = false;
let lgstraightscore = 0;
let yahtzeescore = 0;
let yahtzeebonus = 0;
let chancescore = 0;
let righttotalscore = 0;
let grandtotalscore = 0;
let acesscoredone = false;
let twosscoredone = false;
let threesscoredone = false;
let foursscoredone = false;
let fivesscoredone = false;
let sixesscoredone = false;
let lefttotalscoredone = false;
let toakscoredone = false;
let foakscoredone = false;
let fullhousescoredone = false;
let smstraightscoredone = false;
let lgstraightscoredone = false;
let yahtzeescoredone = false;
let yahtzeebonusdone = false;
let chancedone = false;
let righttotalscoredone = false;
let grandtotalscoredone = false;
```

```
let yahtzeebcount = 4;

function setup() {
  createCanvas(400, 400);
  background(200, 0, 0);
  for(let i = 0; i < 5; i++){
    die = new dice(scale/2 + (5*scale/8)*i, 3*scale)
    append(dices, die);
  }
  for(let i = 0; i < 5; i++){
    button = createButton('hold')
    button.position(11*scale/20 + (5*scale/8)*i, 3.5*scale)
    append(buttons, button);
  }
  for(let i = 0; i < 2; i++){
    for(let j = 0; j < 9; j++){
      button = createButton('>')
      button.position(0 + 2*scale*i, 0 + 7*scale*j/33)
      append(buttons, button)
    }
  }
  push();
  fill(200, 0, 0)
  rect(0, 0, 4*scale, 7*scale/3)
  line(2*scale, 0, 2*scale, 63*scale/33)
  for(let i = 0; i<9; i++){
    line(0, 7*scale/33 + 7*i*scale/33, scale*4, 7*scale/33 + 7*i*scale/33)
  }
  line(4*scale/3, 0, 4*scale/3, 63*scale/33)
  line(10*scale/3, 0, 10*scale/3, 63*scale/33)
  pop();
  buttons[0].mousePressed(hold0);
  buttons[1].mousePressed(hold1);
  buttons[2].mousePressed(hold2);
  buttons[3].mousePressed(hold3);
  buttons[4].mousePressed(hold4);
  buttons[5].mousePressed(aces);
  buttons[6].mousePressed(twos);
  buttons[7].mousePressed(threes);
  buttons[8].mousePressed(fours);
  buttons[9].mousePressed(fives);
  buttons[10].mousePressed(sixes);
  buttons[14].mousePressed(toak);
  buttons[15].mousePressed(foak);
```

```
  buttons[16].mousePressed(fullhouse);
  buttons[17].mousePressed(smstraight);
  buttons[18].mousePressed(lgstraight);
  buttons[19].mousePressed(yahtzee);
  buttons[20].mousePressed(chance);
  buttons[21].mousePressed(yahtzeeb);
  push();
  fill(0)
  text('Aces', scale/3, 6*scale/33)
  text('Twos', scale/3, 13*scale/33)
  text('Threes', scale/3, 20*scale/33)
  text('Fours', scale/3, 27*scale/33)
  text('Fives', scale/3, 34*scale/33)
  text('Sixes', scale/3, 41*scale/33)
  text('Total (B if > 62)', scale/3, 48*scale/33)
  text('B (+35)', scale/3, 55*scale/33)
  text('Total + B', scale/3, 62*scale/33)
  text('3 of a kind', 7*scale/3, 6*scale/33)
  text('4 of a kind', 7*scale/3, 13*scale/33)
  text('Full House', 7*scale/3, 20*scale/33)
  text('SM Straight', 7*scale/3, 27*scale/33)
  text('LG Straight', 7*scale/3, 34*scale/33)
  text('Yahtzee', 7*scale/3, 41*scale/33)
  text('Chance', 7*scale/3, 48*scale/33)
  text('Yahtzee B (+100)', 7*scale/3, 55*scale/33)
  text('Total', 7*scale/3, 62*scale/33)
  textSize(30)
  text('Grand Total:', scale/3, 75*scale/33)
  pop()
  donebutton = createButton('done')
  donebutton.position(3*scale, 2.5*scale)
  donebutton.mousePressed(done)
  append(buttons, donebutton);
  rollbutton = createButton('roll')
  rollbutton.position(37*scale/20, 2.5*scale)
  rollbutton.mousePressed(roll);
  append(buttons, rollbutton)
}

function draw() {
  for(let i = 0; i < 5; i++){
    dices[i].show();
    dices[i].dots();
  }
```

```
  values = [dices[0].getvalue(), dices[1].getvalue(), dices[2].getvalue(), dices[3].getvalue(),
dices[4].getvalue(),]
  square(scale/3, 2.5*scale, scale/5)
  square(20*scale/33, 2.5*scale, scale/5)
  square(29*scale/33, 2.5*scale, scale/5)
  push()
  fill(0)
  textSize(20);
  if(rolls == 0){
    push()
    fill(255)
    square(scale/3, 2.5*scale, scale/5)
    square(20*scale/33, 2.5*scale, scale/5)
    square(29*scale/33, 2.5*scale, scale/5)
    pop();
    for(let i = 0; i < 5; i++){
      if(dices[i].gethold() == true)
        dices[i].changehold();
        buttons[i].style('background-color', 'white')
    }
  }
  if(rolls > 0){
    text('X', 12*scale/33, 88*scale/33);
  }
  if(rolls > 1){
    text('X', 21*scale/33, 88*scale/33);
  }
  if(rolls > 2){
    text('X', 30*scale/33, 88*scale/33);
  }
  pop()
  push()
  fill(200, 0, 0)
  stroke(200, 0, 0)
  text(str(lefttotalscore), 45*scale/33, 48*scale/33)
  text(str(bonusscore), 45*scale/33, 55*scale/33)
  text(str(grandlefttotalscore), 45*scale/33, 62*scale/33)
  text(str(righttotalscore), 111*scale/33, 62*scale/33)
  textSize(30)
  text(str(grandtotalscore), 2*scale, 75*scale/33)
  lefttotalscore = acesscore + twosscore + threesscore + foursscore + fivesscore + sixesscore;
  if(lefttotalscore > 62){
    bonusscore = 35;
  }
```

```
  grandlefttotalscore = lefttotalscore + bonusscore;
  righttotalscore = toakscore + foakscore + fullhousescore + smstraightscore + lgstraightscore +
yahtzeescore + chancescore + (yahtzeebonus * 100);
  grandtotalscore = grandlefttotalscore + righttotalscore;
  pop()
  text(str(lefttotalscore), 45*scale/33, 48*scale/33)
  text(str(bonusscore), 45*scale/33, 55*scale/33)
  text(str(grandlefttotalscore), 45*scale/33, 62*scale/33)
  text(str(righttotalscore), 111*scale/33, 62*scale/33)
  push()
  textSize(30)
  text(str(grandtotalscore), 2*scale, 75*scale/33)
  pop()
}

function roll(){
  if(rolls < 3){
  for(let i = 0; i < 5; i++){
    if(dices[i].gethold() == false)
    dices[i].changevalue();
  }
  rolls++;
}
}

function hold0(){
  dices[0].changehold();
  if(dices[0].gethold() == true)
  buttons[0].style('background-color', GRAY);
  else
    buttons[0].style('background-color', 'white');
}
function hold1(){
  dices[1].changehold();
    if(dices[1].gethold() == true)
  buttons[1].style('background-color', GRAY);
  else
    buttons[1].style('background-color', 'white');
}
function hold2(){
  dices[2].changehold();
    if(dices[2].gethold() == true)
  buttons[2].style('background-color', GRAY);
  else
```

```
    buttons[2].style('background-color', 'white');
}
function hold3(){
  dices[3].changehold();
    if(dices[3].gethold() == true)
  buttons[3].style('background-color', GRAY);
  else
    buttons[3].style('background-color', 'white');
}
function hold4(){
  dices[4].changehold();
    if(dices[4].gethold() == true)
  buttons[4].style('background-color', GRAY);
  else
    buttons[4].style('background-color', 'white');
}

class dice{
  constructor(x, y){
    this.x = x;
    this.y = y;
    this.value = floor(random(6)+1);
    this.hold = false;
  }
  show(){
    square(this.x, this.y, scale/2, scale/10);
    push();
    stroke(255);
    pop();
  }
  gethold(){
    return this.hold;
  }
  changehold(){
    this.hold = !this.hold
  }
  changevalue(){
    this.value = floor(random(6) + 1);
  }
  getvalue(){
    return this.value;
  }
  dots(){
    if(this.value == 1){
```

```
  push();
  fill(0);
  circle(this.x + scale/4, this.y + scale/4, scale/20)
  pop();
}
if(this.value == 2){
  push();
  fill(0);
  circle(this.x + scale/10, this.y + scale/10, scale/20)
  circle(this.x + 8*scale/20, this.y + 8*scale/20, scale/20)
  pop();
}
if(this.value == 3){
  push();
  fill(0);
  circle(this.x + scale/4, this.y + scale/4, scale/20)
  circle(this.x + scale/10, this.y + scale/10, scale/20)
  circle(this.x + 8*scale/20, this.y + 8*scale/20, scale/20)
  pop();
}
if(this.value == 4){
  push();
  fill(0);
  circle(this.x + scale/10, this.y + scale/10, scale/20)
  circle(this.x + 8*scale/20, this.y + 8*scale/20, scale/20)
  circle(this.x + scale/10, this.y + 4*scale/10, scale/20)
  circle(this.x + 8*scale/20, this.y + 2*scale/20, scale/20)
  pop();
}
if(this.value == 5){
  push();
  fill(0);
  circle(this.x + scale/10, this.y + scale/10, scale/20)
  circle(this.x + 8*scale/20, this.y + 8*scale/20, scale/20)
  circle(this.x + scale/10, this.y + 4*scale/10, scale/20)
  circle(this.x + 8*scale/20, this.y + 2*scale/20, scale/20)
  circle(this.x + scale/4, this.y + scale/4, scale/20)
  pop();
}
if(this.value == 6){
  push();
  fill(0);
  circle(this.x + scale/10, this.y + scale/10, scale/20)
  circle(this.x + 8*scale/20, this.y + 8*scale/20, scale/20)
```

```
      circle(this.x + scale/10, this.y + 4*scale/10, scale/20)
      circle(this.x + 8*scale/20, this.y + 2*scale/20, scale/20)
      circle(this.x + scale/10, this.y + scale/4, scale/20)
      circle(this.x + 8*scale/20, this.y + scale/4, scale/20)
      pop();
    }
  }
}

function aces(){
  if(acesscoredone == false && rolls > 0){
  for(let i = 0; i<5; i++){
    if(values[i] == 1)
      acesscore++;
  }
    text(str(acesscore), 45*scale/33, 6*scale/33);
  rolls = 0;
  acesscoredone = true;

  }
}
function twos(){
  if(twosscoredone == false && rolls > 0){
  for(let i = 0; i<5; i++){
    if(values[i] == 2)
      twosscore += 2;
  }
    text(str(twosscore), 45*scale/33, 13*scale/33);
  rolls = 0;
  twosscoredone = true;

  }
}
function threes(){
  if(threesscoredone == false && rolls > 0){
  for(let i = 0; i<5; i++){
    if(values[i] == 3)
      threesscore += 3;
  }
    text(str(threesscore), 45*scale/33, 20*scale/33);
  rolls = 0;
  threesscoredone = true;

  }
```

```
}
function fours(){
  if(foursscoredone == false && rolls > 0){
  for(let i = 0; i<5; i++){
    if(values[i] == 4)
      foursscore += 4;
  }
    text(str(foursscore), 45*scale/33, 27*scale/33);
  rolls = 0;
  foursscoredone = true;

  }
}
function fives(){
  if(fivesscoredone == false && rolls > 0){
  for(let i = 0; i<5; i++){
    if(values[i] == 5)
      fivesscore += 5;
  }
    text(str(fivesscore), 45*scale/33, 34*scale/33);
  rolls = 0;
  fivesscoredone = true;

  }
}
function sixes(){
  if(sixesscoredone == false && rolls > 0){
  for(let i = 0; i<5; i++){
    if(values[i] == 6)
      sixesscore += 6;
  }
    text(str(sixesscore), 45*scale/33, 41*scale/33);
  rolls = 0;
  sixesscoredone = true;

  }
}
function toak(){
  if(toakscoredone == false && rolls > 0){
    sort(values);
    if((values[0] == values[1] && values[1] == values[2]) || (values[2] == values[1] && values[3] ==
values[2]) || (values[2] == values[3] && values[3] == values[4])){
      for(let i = 0; i<5; i++){
        toakscore += values[i]
```

```
      }
      text(str(toakscore), 111*scale/33, 6*scale/33);
      rolls = 0;
      toakscoredone = true;
    }

  }
}
function foak(){
  if(foakscoredone == false && rolls > 0){
    sort(values);
    if((values[0] == values[1] && values[1] == values[2] && values[2] == values[3]) || (values[2] ==
values[1] && values[3] == values[2] && values[4] == values[3])){
      for(let i = 0; i<5; i++){
        foakscore += values[i]
      }
      text(str(foakscore), 111*scale/33, 13*scale/33);
      rolls = 0;
      foakscoredone = true;
    }

  }
}
function fullhouse(){
  if(fullhousescoredone == false && rolls > 0){
    sort(values);
    if((values[0] == values[1] && values[2] == values[3] && values[3] == values[4]) || (values[0] ==
values[1] && values[1] == values[2] && values[3] == values[4])){
       fullhousescore += 25;
      text(str(fullhousescore), 111*scale/33, 20*scale/33);
      rolls = 0;
      fullhousescoredone = true;
     }

  }
}
function smstraight(){
  if(smstraightscoredone == false && rolls > 0){
    sort(values);
      for(let j = 0; j<4; j++){
        if(values[0] + j == values[j])
          smstraightscoret = true;
        else{
            smstraightscoret = false;
```

```
        break;
      }
    }
  if(smstraightscoret == false){
  for(let i = 0; i<4; i++){
    if(values[0] + i == values[i])
        smstraightscoret = true;
      else{
         smstraightscoret = false;
        break;
      }
  }
  }
  if(smstraightscoret){
    smstraightscore += 30;
    text('30', 111*scale/33, 27*scale/33);
    rolls = 0;
    smstraightscoredone = true;
  }

 }
}
function lgstraight(){
 if(lgstraightscoredone == false && rolls > 0){
   sort(values);
    for(let j = 0; j<5; j++){
      if(values[0] + j == values[j])
        lgstraightscoret = true;
      else{
         lgstraightscoret = false;
        break;
      }
    }
  if(lgstraightscoret){
    lgstraightscore += 40;
    text('40', 111*scale/33, 34*scale/33);
    rolls = 0;
    lgstraightscoredone = true;
  }

 }
}
function yahtzee(){
 if(yahtzeescoredone == false && rolls > 0){
```

```
    if(values[0] == values[1] && values[0] == values[2] && values[0] == values[3] && values[0] ==
values[4]){
      yahtzeescore += 50;
      text('50', 111*scale/33, 41*scale/33);
      rolls = 0;
      yahtzeescoredone = true;
    }

  }
}
function chance(){
  if(chancedone == false && rolls > 0){
    for(let i = 0; i<5; i++){
      chancescore += values[i];
    }
    text(str(chancescore), 111*scale/33, 48*scale/33);
    rolls = 0;
    chancedone = true;

  }
}
function yahtzeeb(){
  if(yahtzeescoredone == true && yahtzeebcount > 0 && rolls > 0){
    if(values[0] == values[1] && values[0] == values[2] && values[0] == values[3] && values[0] ==
values[4]){
      push()
      fill(200, 0, 0)
      text('x' + str(yahtzeebonus), 111*scale/33, 55*scale/33)
      pop()
      yahtzeebonus += 1;
      text('x' + str(yahtzeebonus), 111*scale/33, 55*scale/33);
      rolls = 0;

  }
  }
}
function done(){
 for(let i = 0; i < 25; i++){
   noLoop()
   buttons[i].style('background-color', 'black');
   fill(0)
   square(0, 0, scale*4)
   push()
   fill(255)
```

```
    textSize(20)
    text('Congrats! Your grand total was ' + str(grandtotalscore), scale/2, 2.9*scale)
    pop()
  }
}
```