```
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;
let resetcheck = false;
let resetcount = 600;
let buttonnumber = 0;
let actioncount = 50;
let actionnum = 0;
let actnum = 0;
let donecount = 7200;

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 + buttonnumber].mousePressed(hold0);
```

```
buttons[1 + buttonnumber].mousePressed(hold1);
buttons[2 + buttonnumber].mousePressed(hold2);
buttons[3 + buttonnumber].mousePressed(hold3);
buttons[4 + buttonnumber].mousePressed(hold4);
buttons[5 + buttonnumber].mousePressed(aces);
buttons[6 + buttonnumber].mousePressed(twos);
buttons[7 + buttonnumber].mousePressed(threes);
buttons[8 + buttonnumber].mousePressed(fours);
buttons[9 + buttonnumber].mousePressed(fives);
buttons[10 + buttonnumber].mousePressed(sixes);
buttons[14 + buttonnumber].mousePressed(toak);
buttons[15 + buttonnumber].mousePressed(foak);
buttons[16 + buttonnumber].mousePressed(fullhouse);
buttons[17 + buttonnumber].mousePressed(smstraight);
buttons[18 + buttonnumber].mousePressed(lgstraight);
buttons[19 + buttonnumber].mousePressed(yahtzee);
buttons[20 + buttonnumber].mousePressed(chance);
buttons[21 + buttonnumber].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() {
  if (!resetcheck) {
    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(),
    ];
    actioncount--;
    if (actioncount == 0 || rolls == 0){
      actionnum = floor(random(3)+1);
      actioncount = 50;
      if(actionnum == 3 || rolls == 3){
        actnum  = floor(random(14)+1);
        if(actnum == 1)
          aces();
        if(actnum == 2)
          twos();
        if(actnum == 3)
          threes();
        if(actnum == 4)
          fours();
        if(actnum == 5)
          fives();
        if(actnum == 6)
          sixes();
        if(actnum == 7)
           toak();
        if(actnum == 8)
          foak();
        if(actnum == 9)
          fullhouse();
    if(actnum == 10)
      smstraight();
    if(actnum == 11)
      lgstraight();
    if(actnum == 12)
      yahtzee();
    if(actnum == 13)
      chance();
    if(actnum == 14)
      yahtzeeb();
  }
  else
    roll();
}
donecount--;
if(donecount == 0){
  done()
}
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);
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 + buttonnumber].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();
        push();
        fill(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);
        pop();
      }
    else {
        resetcount--;
        if (resetcount == 0) {
```

```
      reset();
    }
  }
}

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 + buttonnumber].style("background-color", GRAY);
  else buttons[0 + buttonnumber].style("background-color", "white");
}
function hold1() {
  dices[1].changehold();
  if (dices[1].gethold() == true)
    buttons[1 + buttonnumber].style("background-color", GRAY);
  else buttons[1 + buttonnumber].style("background-color", "white");
}
function hold2() {
  dices[2].changehold();
  if (dices[2].gethold() == true)
    buttons[2 + buttonnumber].style("background-color", GRAY);
  else buttons[2 + buttonnumber].style("background-color", "white");
}
function hold3() {
  dices[3].changehold();
  if (dices[3].gethold() == true)
    buttons[3 + buttonnumber].style("background-color", GRAY);
  else buttons[3 + buttonnumber].style("background-color", "white");
}
function hold4() {
  dices[4].changehold();
  if (dices[4].gethold() == true)
    buttons[4 + buttonnumber].style("background-color", GRAY);
  else buttons[4 + buttonnumber].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++;
    }
    push();
    fill(0);
    text(str(acesscore), (45 * scale) / 33, (6 * scale) / 33);
    pop();
    rolls = 0;
    acesscoredone = true;
  }
}
function twos() {
  if (twosscoredone == false && rolls > 0) {
    for (let i = 0; i < 5; i++) {
      if (values[i] == 2) twosscore += 2;
    }
    push();
    fill(0);
    text(str(twosscore), (45 * scale) / 33, (13 * scale) / 33);
    pop();
    rolls = 0;
    twosscoredone = true;
  }
}
function threes() {
  if (threesscoredone == false && rolls > 0) {
    for (let i = 0; i < 5; i++) {
      if (values[i] == 3) threesscore += 3;
    }
    push();
    fill(0);
    text(str(threesscore), (45 * scale) / 33, (20 * scale) / 33);
    pop();
    rolls = 0;
    threesscoredone = true;
  }
}
function fours() {
  if (foursscoredone == false && rolls > 0) {
    for (let i = 0; i < 5; i++) {
      if (values[i] == 4) foursscore += 4;
    }
    push();
    fill(0);
    text(str(foursscore), (45 * scale) / 33, (27 * scale) / 33);
    pop();
    rolls = 0;
    foursscoredone = true;
  }
```

```
}
function fives() {
  if (fivesscoredone == false && rolls > 0) {
    for (let i = 0; i < 5; i++) {
      if (values[i] == 5) fivesscore += 5;
    }
    push();
    fill(0);
    text(str(fivesscore), (45 * scale) / 33, (34 * scale) / 33);
    pop();
    rolls = 0;
    fivesscoredone = true;
  }
}
function sixes() {
  if (sixesscoredone == false && rolls > 0) {
    for (let i = 0; i < 5; i++) {
      if (values[i] == 6) sixesscore += 6;
    }
    push();
    fill(0);
    text(str(sixesscore), (45 * scale) / 33, (41 * scale) / 33);
    pop();
    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];
      }
      push();
      fill(0);
      text(str(toakscore), (111 * scale) / 33, (6 * scale) / 33);
      pop();
      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];
      }
      push();
      fill(0);
      text(str(foakscore), (111 * scale) / 33, (13 * scale) / 33);
      pop();
      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;
      push();
      fill(0);
      text(str(fullhousescore), (111 * scale) / 33, (20 * scale) / 33);
      pop();
      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;
      push();
      fill(0);
      text("30", (111 * scale) / 33, (27 * scale) / 33);
      pop();
      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;
      push();
      fill(0);
      text("40", (111 * scale) / 33, (34 * scale) / 33);
      pop();
      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;
      push();
      fill(0);
      text("50", (111 * scale) / 33, (41 * scale) / 33);
      pop();
      rolls = 0;
      yahtzeescoredone = true;
    }
  }
}
function chance() {
  if (chancedone == false && rolls > 0) {
    for (let i = 0; i < 5; i++) {
      chancescore += values[i];
    }
    push();
    fill(0);
    text(str(chancescore), (111 * scale) / 33, (48 * scale) / 33);
    pop();
    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;
      push();
      fill(0);
      text("x" + str(yahtzeebonus), (111 * scale) / 33, (55 * scale) / 33);
      pop();
      rolls = 0;
    }
  }
}
function done() {
  for (let i = 0; i < 25; i++) {
    buttons[i + buttonnumber].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();
    resetcheck = true;
  }
}
function reset() {
  rolls = 0;
  acesscore = 0;
  twosscore = 0;
  threesscore = 0;
  foursscore = 0;
  fivesscore = 0;
  sixesscore = 0;
  lefttotalscore = 0;
  bonusscore = 0;
  grandlefttotalscore = 0;
  toakscore = 0;
  foakscore = 0;
  fullhousescore = 0;
  smstraightscoret = false;
  smstraightscore = 0;
  lgstraightscoret = false;
  lgstraightscore = 0;
  yahtzeescore = 0;
  yahtzeebonus = 0;
  chancescore = 0;
  righttotalscore = 0;
  grandtotalscore = 0;
  acesscoredone = false;
  twosscoredone = false;
  threesscoredone = false;
  foursscoredone = false;
  fivesscoredone = false;
  sixesscoredone = false;
  lefttotalscoredone = false;
  toakscoredone = false;
  foakscoredone = false;
  fullhousescoredone = false;
  smstraightscoredone = false;
  lgstraightscoredone = false;
  yahtzeescoredone = false;
  yahtzeebonusdone = false;
  chancedone = false;
  righttotalscoredone = false;
  grandtotalscoredone = false;
  yahtzeebcount = 4;
  resetcheck = false;
  resetcount = 600;
  buttonnumber += 25;
  setup();
}
```