Final Program code Space Assault

var f = createFont("Arial Black");

// bitmoji code for Jarrod's Bitmoji

var drawBitmojiHead = function (headX, headY, headH){

noStroke();

fill(255,205,148);

ellipse(headX-(headH/100\*0),headY-(headH/100\*0),headH/100\*83,headH/100\*100); //head

fill(53,40,30);

arc(headX-(headH/100\*32), headY+(headH/100\*13), headH/100\*20, headH/100\*20, 0, 361 );//beard section

arc(headX-(headH/100\*32), (headY+headH/100\*19), headH/100\*20, headH/100\*20, 0,361);

arc(headX-(headH/100\*30), (headY+headH/100\*25), headH/100\*20, headH/100\*20, 0,361);

arc(headX-(headH/100\*26), headY+(headH/100\*32), headH/100\*20, headH/100\*20, 0, 361 );

arc(headX-(headH/100\*23), headY+(headH/100\*35), headH/100\*20, headH/100\*20, 0, 361 );

arc(headX-(headH/100\*17), headY+(headH/100\*40), headH/100\*20, headH/100\*20, 0, 361 );

arc(headX-(headH/100\*9), headY+(headH/100\*43), headH/100\*20, headH/100\*20, 0, 361);

arc(headX-(headH/100\*0), headY+(headH/100\*43), headH/100\*20, headH/100\*20, 0, 361);

arc(headX+(headH/100\*7), headY+(headH/100\*43), headH/100\*20, headH/100\*20, 0, 361);

arc(headX+(headH/100\*17), headY+(headH/100\*36), headH/100\*20, headH/100\*20, 0, 361 );

arc(headX+(headH/100\*27), headY+(headH/100\*28), headH/100\*20, headH/100\*20, 0,361);

arc(headX+(headH/100\*33), headY+(headH/100\*18), headH/100\*20, headH/100\*20, 0,361);

arc(headX+(headH/100\*35), headY+(headH/100\*13), headH/100\*20, headH/100\*20, 0,361);

rect(headX-(headH/100\*11), headY+(headH/100\*26), headH/100\*19, headH/100\*9, headH /100\*15);

rect(headX-(headH/100\*11), headY+(headH/100\*9), headH/100\*19, headH/100\*9, headH /100\*15);// moustache

quad(headX-(headH/100\*44),headY-(headH/102\*0), headX-(headH/100\*35), headY-(headH /100\*32), headX-(headH/100\*14), headY-(headH/100\*45), headX-(headH/100\*37), headY -(headH/100\*10));// left hair

quad(headX+(headH/100\*43),headY-(headH/100\*5),headX+(headH/100\*34),headY-(headH/100 \*32),headX+(headH/100\*13),headY-(headH/100\*50),headX+(headH/100\*35),headY-(headH/100 \*3));//right hair

stroke(0, 0, 0);//hat

fill(72, 211, 242);

rect(headX-(headH/100\*36), headY-(headH/100\*59), headH/100\*73, headH/100\*31, headH /100\*15);

rect(headX-(headH/100\*26), headY-(headH/100\*73), headH/100\*52, headH/100\*28, headH /100\*3);

fill(21, 122, 66);

rect(headX-(headH/100\*31), headY-(headH/100\*33), headH/100\*62, headH/100\*6, headH /100\*15);

stroke(255, 255, 255);

fill(72, 211, 242);

ellipse(headX-(headH/100\*16),headY-(headH/100\*9),headH/100\*10,headH/100\*8);//left eye

ellipse(headX+(headH/100\*11),headY-(headH/100\*9),headH/100\*10,headH/100\*8);//right eye

noStroke();

fill(0, 0, 0);

ellipse(headX-(headH/100\*16),headY-(headH/100\*9),headH/100\*4,headH/100\*4);//pupils

ellipse(headX+(headH/100\*11),headY-(headH/100\*9),headH/100\*4,headH/100\*4);

stroke(0, 0, 0);

fill(255,205,148);

bezier(headX-(headH/100\*2),headY+(headH/100\*2),headX+(headH/100\*27),headY+(headH /100\*13),headX-(headH/100\*16),headY+(headH/100\*12),headX-(headH/100\*4),headY+(headH /100\*10));// nose

fill(255, 255, 255); //white teeth

arc(headX-(headH/100\*2),headY+(headH/100\*19),headH/100\*30,headH/100\*13,1,180); //mouth

line(headX-(headH/100\*16),headY+(headH/100\*18),headX+(headH/100\*12),headY+(headH /100\*18));

};

var drawBitmojiBody = function(bodyX, bodyY,bodyH) {

noStroke();

fill(255,205,148);//neck

rect(bodyX-(bodyH/100\*17), bodyY+ (bodyH/100\*51), bodyH/100\*32, bodyH/100\*21, bodyH /100\*8);

fill(0, 0, 0);

rect(bodyX-(bodyH/100\*48), bodyY+(bodyH/100 \*70), bodyH/100\*97, bodyH/100\*70, bodyH /100\*26);//shirt

fill(255, 255, 255);

textSize(bodyH/100\*29);

text("JB",bodyX-(bodyH/100\*17), bodyY+(bodyH/100\*92), bodyH/100\*49, bodyH/100\*29);

};

var drawBitmoji = function(bitmojiX,bitmojiY,bitmojiH) {

drawBitmojiHead(bitmojiX,bitmojiY,bitmojiH);

drawBitmojiBody(bitmojiX,bitmojiY,bitmojiH);

};

// bitmoji code for Nicolas's Bitmoji

var drawBitMojiHead = function (bodyX,bodyY,bodyH) {

noStroke();

fill (0, 0, 0);

fill ( 255, 224, 189); // skin tone fill

ellipse (bodyX, bodyY, bodyH / 100\*86, bodyH / 100 \* 94); // head/face

fill (255, 255, 255);

fill (94, 20, 20); // hair color tone fill

triangle (bodyX + bodyH / 100 \* 54, bodyY + bodyH / 100\*5, bodyX + bodyH / 100 \* 38, bodyY - bodyH / 100 \* 9, bodyX + ( bodyH / 100 \* 33 ), bodyY - (bodyH / 100 \* 28 )); // right hair piece

triangle (bodyX + (bodyH / 100 \* 35), bodyY + (bodyH / 100 \* 5), bodyX + (bodyH / 100 \* 38), bodyY - (bodyH / 100 \* 9), bodyX + (bodyH / 100 \* 33), bodyY - (bodyH / 100 \* 28));

// right piece continued

triangle (bodyX - bodyH / 100 \* 26, bodyY - (bodyH / 100 \* 45), bodyX + (bodyH / 100 \* 38), bodyY-(bodyH/100\*9), bodyX + (bodyH / 100 \* 33), bodyY-(bodyH/100\*28));

// right hair piece continued

triangle (bodyX - (bodyH / 100 \* 37), bodyY - bodyH / 100 \* 23, bodyX - bodyH / 100 \* 54, bodyY-bodyH/100\*7, bodyX + bodyH / 100 \* 33, bodyY - bodyH / 100 \* 28); // left hair piece

fill (79, 76, 79); // beanie color fill

arc (bodyX - bodyH / 100 \* 2, bodyY - bodyH / 100 \* 19, bodyH / 100 \* 84, bodyH / 100 \* 75, 178, 363); // beanie

stroke (0, 0, 0); // fill for the outline for glasses

fill (255, 224, 189); //fill to skin tone for glasses

rect (bodyX - bodyH / 100 \* 27, bodyY - bodyH / 100 \* 11,bodyH / 100 \* 17,bodyH / 100 \* 12); // left glasses

rect (bodyX + bodyH / 100\*6, bodyY - bodyH / 100\*11, bodyH / 100 \* 17, bodyH / 100 \* 12); // right glasses

line (bodyX - bodyH / 100 \* 10,bodyY-bodyH / 100 \* 3, bodyX + bodyH / 100 \* 6, bodyY - bodyH / 100 \* 3);

// middle bridge of glasses

line (bodyX - bodyH/100\*44, bodyY - bodyH / 100 \* 5,bodyX - bodyH / 100 \* 26,bodyY - bodyH/100\*9); // glasses side left

line (bodyX + (bodyH / 100 \* 24), bodyY-(bodyH / 100 \* 10), bodyX + (bodyH / 100 \* 44), bodyY - (bodyH / 100 \* 7)); // glasses side right

fill (115, 23, 23); // fill for brown eyes

ellipse (bodyX - (bodyH / 100 \* 17),bodyY - (bodyH / 100 \* 4),bodyH / 100 \* 6,bodyH / 100\* 6);// left eye

ellipse (bodyX + (bodyH / 100 \* 17),bodyY - (bodyH / 100 \* 4),bodyH / 100 \* 6,bodyH / 100\* 6);// right eye

fill (255, 224, 189); // skin tone fill

bezier (bodyX , (bodyY - bodyH / 100 \* 1), bodyX + (bodyH / 100 \* 21),bodyY + (bodyH / 100 \* 17), bodyX - (bodyH / 100 \* 8), bodyY + (bodyH / 100 \* 17),bodyX + (bodyH / 100 \* 8),bodyY + (bodyH / 100 \* 19)); // nose

fill (255, 255, 255); // fil the whites of the teeth

arc (bodyX + (bodyH / 100 \* 3),bodyY + (bodyH / 100 \* 25),bodyH / 100 \* 30,bodyH / 100 \* 13, 1, 180);

line (bodyX - (bodyH / 100 \* 14),bodyY + (bodyH / 100 \* 24),bodyX + (bodyH / 100 \* 19), bodyY + (bodyH / 100 \* 24)); // top of the mouth

};

var drawBitMojiBody = function (bodyX, bodyY, bodyH) {

fill ( 255, 224, 189); // skin tone fill

noStroke();

rect (bodyX - bodyH / 100 \* 18,bodyY + bodyH / 100 \* 39, bodyH / 100 \* 42, bodyH / 100 \* 24); // neck

fill (0, 0, 0);

quad (bodyX - bodyH / 100 \* 52, bodyY + (bodyH / 100 \* 88), bodyX + (bodyH / 100 \* 63), bodyY + (bodyH / 100 \* 88), bodyX + (bodyH / 100 \* 24), bodyY + (bodyH / 100 \* 52 ), bodyX - (bodyH / 100 \* 17), bodyY + (bodyH/100\*52));// t -shirt/body

rect (bodyX - bodyH / 100 \* 52, bodyY + bodyH / 100 \* 87,bodyH / 100 \* 115, bodyH / 100\* 51); // body continued

fill (255, 255, 255);

textSize ( bodyH/ 100 \* 30);

text ("NL", bodyX - bodyH / 100 \* 15, bodyY + bodyH / 100 \* 110);

};

var drawBitMoji = function (bodyX, bodyY, bodyH) {

drawBitMojiHead ( bodyX, bodyY, bodyH);

drawBitMojiBody ( bodyX, bodyY, bodyH);

};

var drawSpaceShip = function(shipX,shipY, shipH){

noStroke();

//top middle section of spaceship and wings

fill(252, 252, 252);

rect(shipX+(shipH/100\*141),shipY+(shipH/100\*170),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*141),shipY+(shipH/100\*155),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*141),shipY+(shipH/100\*141),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*141),shipY+(shipH/100\*126),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*141),shipY+(shipH/100\*111),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*126),shipY+(shipH/100\*111),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*126),shipY+(shipH/100\*125),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*111),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*96),shipY+(shipH/100\*111),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*81),shipY+(shipH/100\*111),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*81),shipY+(shipH/100\*125),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*96),shipY+(shipH/100\*125),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*81),shipY+(shipH/100\*140),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*81),shipY+(shipH/100\*155),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*81),shipY+(shipH/100\*170),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*96),shipY+(shipH/100\*96),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*96),shipY+(shipH/100\*81),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*96),shipY+(shipH/100\*66),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*96),shipY+(shipH/100\*51),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*96),shipY+(shipH/100\*170),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*170),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*126),shipY+(shipH/100\*170),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*153),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*161),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*185),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*200),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*96),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*81),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*66),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*51),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*96),shipY+(shipH/100\*200),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*96),shipY+(shipH/100\*185),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*126),shipY+(shipH/100\*96),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*126),shipY+(shipH/100\*81),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*126),shipY+(shipH/100\*66),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*126),shipY+(shipH/100\*51),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*154),shipY+(shipH/100\*142),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*169),shipY+(shipH/100\*142),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*183),shipY+(shipH/100\*142),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*183),shipY+(shipH/100\*128),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*183),shipY+(shipH/100\*113),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*183),shipY+(shipH/100\*100),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*183),shipY+(shipH/100\*156),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*66),shipY+(shipH/100\*142),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*51),shipY+(shipH/100\*142),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*36),shipY+(shipH/100\*142),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*36),shipY+(shipH/100\*157),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*36),shipY+(shipH/100\*127),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*36),shipY+(shipH/100\*112),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*36),shipY+(shipH/100\*100),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*36),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*21),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*16),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*125),shipY+(shipH/100\*185),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*125),shipY+(shipH/100\*199),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*215),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*230),shipH/100\*15,shipH/100\*15);

// red center of spaceship

fill(255, 0, 0);

rect(shipX+(shipH/100\*126),shipY+(shipH/100\*140),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*140),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*126),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*96),shipY+(shipH/100\*139),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*96),shipY+(shipH/100\*154),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*126),shipY+(shipH/100\*154),shipH/100\*15,shipH/100\*15);

//bottom red middle

rect(shipX+(shipH/100\*81),shipY+(shipH/100\*185),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*81),shipY+(shipH/100\*200),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*81),shipY+(shipH/100\*214),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*66),shipY+(shipH/100\*200),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*66),shipY+(shipH/100\*214),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*141),shipY+(shipH/100\*185),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*141),shipY+(shipH/100\*200),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*141),shipY+(shipH/100\*214),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*156),shipY+(shipH/100\*200),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*156),shipY+(shipH/100\*214),shipH/100\*15,shipH/100\*15);

// tips of the spaceship

fill(20, 35, 201);

rect(shipX+(shipH/100\*36),shipY+(shipH/100\*86),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*183),shipY+(shipH/100\*86),shipH/100\*15,shipH/100\*15);

rect(shipX+(shipH/100\*111),shipY+(shipH/100\*2),shipH/100\*15,shipH/100\*15);

}; // code for the image of the spaceship

var drawAlien = function(alienX,alienY,alienH) {

noStroke();

// left to right starting at 200 center point.

fill(30, 214, 67);

rect(alienX+(alienH/100\*100),alienY+(alienH/100\*100),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*115),alienY+(alienH/100\*100),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*130),alienY+(alienH/100\*100),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*145),alienY+(alienH/100\*100),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*160),alienY+(alienH/100\*100),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*175),alienY+(alienH/100\*100),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*190),alienY+(alienH/100\*100),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*205),alienY+(alienH/100\*100),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*115),alienY+(alienH/100\*55),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*130),alienY+(alienH/100\*55),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*145),alienY+(alienH/100\*55),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*160),alienY+(alienH/100\*55),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*175),alienY+(alienH/100\*55),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*190),alienY+(alienH/100\*55),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*130),alienY+(alienH/100\*40),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*145),alienY+(alienH/100\*40),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*160),alienY+(alienH/100\*40),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*175),alienY+(alienH/100\*40),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*100),alienY+(alienH/100\*85),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*145),alienY+(alienH/100\*85),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*160),alienY+(alienH/100\*85),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*175),alienY+(alienH/100\*100),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*190),alienY+(alienH/100\*100),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*205),alienY+(alienH/100\*85),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*100),alienY+(alienH/100\*70),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*115),alienY+(alienH/100\*70),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*130),alienY+(alienH/100\*70),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*145),alienY+(alienH/100\*70),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*160),alienY+(alienH/100\*70),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*175),alienY+(alienH/100\*70),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*190),alienY+(alienH/100\*70),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*205),alienY+(alienH/100\*70),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*130),alienY+(alienH/100\*115),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*145),alienY+(alienH/100\*115),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*160),alienY+(alienH/100\*115),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*175),alienY+(alienH/100\*115),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*100),alienY+(alienH/100\*130),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*115),alienY+(alienH/100\*130),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*145),alienY+(alienH/100\*130),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*160),alienY+(alienH/100\*130),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*190),alienY+(alienH/100\*130),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*205),alienY+(alienH/100\*130),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*160),alienY+(alienH/100\*145),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*145),alienY+(alienH/100\*145),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*145),alienY+(alienH/100\*160),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*160),alienY+(alienH/100\*160),alienH/100\*15,alienH/100\*15);

fill(255, 0, 0);

rect(alienX+(alienH/100\*115),alienY+(alienH/100\*85),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*130),alienY+(alienH/100\*85),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*175),alienY+(alienH/100\*85),alienH/100\*15,alienH/100\*15);

rect(alienX+(alienH/100\*190),alienY+(alienH/100\*85),alienH/100\*15,alienH/100\*15);

}; // code for the first level alien

var numAlien = 5;

var score = 0;

var currentScene = 0;

var laserSpeed = 1000;

var shipLife = 3; // how many lives the ship has at the start

var Button = function(config) {

this.color = config.color;

this.x = config.x || 0;

this.y = config.y || 0;

this.width = config.width || 141;

this.height = config.height || 50;

this.label = config.label || "Click";

this.onClick = config.onClick || function() {};

};

Button.prototype.draw = function() {

fill(this.color);

stroke(0, 0, 0);

rect(this.x, this.y, this.width, this.height, 5);

fill(0, 0, 0);

textSize(20);

textAlign(LEFT, TOP);

text(this.label, this.x+10, this.y+this.height/4);

};

Button.prototype.isMouseInside = function() {

return mouseX > this.x &&

mouseX < (this.x + this.width) &&

mouseY > this.y &&

mouseY < (this.y + this.height);

};

Button.prototype.handleMouseClick = function() {

if (this.isMouseInside()) {

this.onClick();

}

};

var btn1 = new Button({ //start button

color: color(255, 255, 255),

x: 126,

y: 233,

label: "Start Game",

onClick: function() {

currentScene = 1;

}

});

var btn2 = new Button({ //controls

color: color(255, 255, 255),

x: 128,

y: 295,

label: " Controls",

onClick: function() {

currentScene = 2;

}

});

var btn3 = new Button({ //reset from instruction button

color: color(255, 255, 255),

x: 126,

y: 295,

label: "Main Menu",

onClick: function() {

currentScene = 0;

}

});

var btn4 = new Button({ //reset from game over and reset from winning

color: color(255, 255, 255),

x: 290,

y: 340,

label: "Restart",

onClick: function() {

Program.restart(); // helps to reset the program when the button is clicked

}

});

var splashScreen = function() { // game start screen

fill(0, 102, 255);

rectMode(CORNER);

rect(0,0,400,400);

fill(0, 0, 0);

textFont(f);

textSize(40);

text("Space Assault", 46,118);

textSize(13);

text("by Nicolas Leffray & Jarrod Blanchette",65,178);

btn1.draw();

btn2.draw();

drawBitmoji(73,260,50);

drawBitMoji(321,260,50);

};

splashScreen();

var instructions = function() { // instructions/controls scene

fill(238, 255, 0);

rectMode(CORNER);

rect(0,0,400,400);

fill(0, 0, 0);

textFont(f);

textSize(40);

text("Instructions",64,82);

textSize(15);

text("arrow keys = movement",99,142);

text("spacebar = fire",123,169);

btn3.draw();

};

instructions();

var stars = [];

var Ship = function(x, y, height) { //player 1 ship constructor function

this.x = x;

this.y = y;

this.height = height;

this.shipLife = shipLife;

};

Ship.prototype.draw = function() { //method for drawing the ship

rectMode(CENTER);

this.x = constrain(this.x, 0, height-75);

this.y = constrain(this.y, 0, height-60);

drawSpaceShip(this.x, this.y, this.height);

};

//below are methods for making the ship move

Ship.prototype.left = function() {

this.x -= 5;

};

Ship.prototype.right = function() {

this.x += 5;

};

Ship.prototype.up = function() {

this.y -= 5;

};

Ship.prototype.down = function() {

this.y += 5;

};

// checks for the collision between alien 1 (green) and the ship itself

Ship.prototype.checkForCollision = function(alien1) {

if ((alien1.x >= (this.x-30) && alien1.x <= (this.x + 30)) &&

(alien1.y >= (this.y-40) && alien1.y <= (this.y + 40))) {

alien1.y = 500;

shipLife--;

}

};

// checks for the collision between alien 2 (bitmoji) and the ship itself

Ship.prototype.checkForAlien2 = function(alien2) {

if ((alien2.x >= (this.x-25) && alien2.x <= (this.x + 25)) &&

(alien2.y >= (this.y-40) && alien2.y <= (this.y + 40))) {

alien2.y = 500;

shipLife--;

}

};

// checks for the collision between alien 3 (bitmoji) and the ship itself

Ship.prototype.checkForAlien3 = function(alien3) {

if ((alien3.x >= (this.x-25) && alien3.x <= (this.x + 25)) &&

(alien3.y >= (this.y-40) && alien3.y <= (this.y + 40))) {

alien3.y = 500;

shipLife--;

}

};

var Alien = function(x, y, height) {// alien constructor

this.x = x;

this.y = y;

this.height = height;

this.speed = 1;

};

var Alien2 = function(x,y,height) {//alien 2 constructor

this.x = x;

this.y = y;

this.height = height;

this.speed = 1;

};

var Alien3 = function(x,y,height) {//alien 3 constructor

this.x = x;

this.y = y;

this.height = height;

this.speed = 2;

};

Alien.prototype.draw = function() {// alien draw method

drawAlien(this.x,this.y,this.height);

};

Alien2.prototype.draw = function() {// alien 2 draw method

rectMode(CORNER);

drawBitmoji(this.x, this.y,this.height);

};

Alien3.prototype.draw = function() { //alien 3 draw method

rectMode(CORNER);

drawBitMoji(this.x, this.y,this.height);

};

var alien1 = []; // empty array

for (var i=0; i < numAlien; i++) { // for loop that pushes in a new version of alien 1

alien1.push(new Alien(i\*75, i/10, 25));

}

for (var i=0; i < numAlien; i++) {

alien1.push(new Alien(i\*85, 45, 25));

}

var alien2 = [];

for (var i=0; i < numAlien; i++) {

alien2.push(new Alien2(i\*75, i/10 + 40, 25));

}

for (var i=0; i < numAlien; i++) {

alien2.push(new Alien2(i\*75, i/10 - 150, 25));

}

var alien3 = [];

for (var i=0; i < numAlien; i++) {

alien3.push(new Alien3(i\*75+40, i/10 + 40, 25));

}

for (var i=0; i < numAlien; i++) {

alien3.push(new Alien3(i\*75+50, i/10 - 200, 25));

}

var ship = new Ship(177,338,25); // new instance of Ship

var Laser = function(x,y) {//constructor function laser

this.x = x;

this.y = y;

this.speed = 35;

};

Laser.prototype.draw = function() {//laser draw method

rectMode(CENTER);

fill(255, 0, 0);

rect(this.x,this.y,7,14,5);

};

Laser.prototype.checkForCollision = function(alien1) { //check for collision between laser and alien1

if ((alien1.x >= (this.x - 40) && alien1.x <= (this.x - 20)) &&

(alien1.y >= (this.y - 10) && alien1.y <= (this.y + 60))) {

alien1.y = 500;

score++;

}

};

Laser.prototype.checkForAlien2 = function(alien2) {//check for collision between laser and alien2

if ((alien2.x >= (this.x - 30) && alien2.x <= (this.x + 1)) &&

(alien2.y >= (this.y - 10) && alien2.y <= (this.y + 40))) {

alien2.y = 500;

score++;

}

};

Laser.prototype.checkForAlien3 = function(alien3) {//check for collision between laser and alien3

if ((alien3.x >= (this.x - 30) && alien3.x <= (this.x + 1)) &&

(alien3.y >= (this.y - 10) && alien3.y <= (this.y + 40))) {

alien3.y = 500;

score++;

}

};

var lasers =[]; // empty lasers array

var gameScene = function() { // the first level or the first playable scene

fill(0, 0, 0);

background(0, 0, 0);

fill(255, 255, 255);

// a loop that runs through a empty array of ellipses(stars)

for ( var i = 0; i < stars.length; i++){

ellipse(stars[i].x,stars[i].y,2,2);

}

// this if statement helps with the retro look of stars blinking in an out

if (frameCount % 80 === 0) {

for ( var i = 0; i < 100; i++)

{

stars[i] = ({x: random(0,400), y: random(0,400)});

}

}

textSize(20);

fill(255, 0, 0);

text("score: " + score ,20,20); // sets the score to print at the top of the screen

text("Lives: " + shipLife ,13,364); // sets the lives to print at the bottom of the screen

for (var i = 0; i < alien1.length; i++) {// a for loop that prints out the aliens

alien1[i].draw();

ship.checkForCollision(alien1[i]);

alien1[i].x += alien1[i].speed;

if (alien1[i].x >=350) { // the if statement causes the aliens to go faster when reaching one side of the game dimensions

alien1[i].speed = -2;

alien1[i].y += 25;

}

if (alien1[i].x <= 0) {

alien1[i].speed = 2;

alien1[i].y += 25;

}

}

// nested for loop allows detection of lasers in array interacting with alien1 in array

for ( var i = 0; i < lasers.length; i++) {

for (var j = 0; j < alien1.length;j++){

lasers[i].checkForCollision(alien1[j]);}

lasers[i].draw();

lasers[i].y -= lasers[i].speed;

// if laser goes off the screen it is then spliced out of the array

if (lasers[i].y < 0 ) {

lasers.splice(i,1);

}

}

// sets a frameCount shooting speed and pushed in a new laser once a key is pressed

if ( frameCount % 10 === 0){

if (keyIsPressed && keyCode === 0) {

lasers.push(new Laser(ship.x + 27, ship.y - 6));

}

}

if (keyIsPressed && keyCode === 37){ // movement for spaceship in the left direction

ship.left();

}

if (keyIsPressed && keyCode === 39){ // movement for spaceship in the right direction

ship.right();

}

if (keyIsPressed && keyCode === 38){ // movement for spaceship in the forward direction

ship.up();

}

if (keyIsPressed && keyCode === 40){ // movement for spaceship in the reverse direction

ship.down();

}

ship.draw();

// creates the game over if ship life reached 0

if (shipLife <= 0) {

rectMode(CORNER);

fill(255, 0, 0);

textSize(40);

text("Game Over", 79,200);

btn4.draw();

}

};

var gameScene3 = function() {

currentScene = 3; //setting the currentscene to 3

fill(0, 0, 0);

background(0, 0, 0);

fill(255, 255, 255);

// creates the moving stars background

for ( var i = 0; i < stars.length; i++){

ellipse(stars[i].x,stars[i].y,2,2);

}

if (frameCount % 80 === 0) {

for ( var i = 0; i < 100; i++)

{

stars[i] = ({x: random(0,400), y: random(0,400)});

}

}

textSize(20);

fill(255, 0, 0);

text("score: " + score ,20,20); // sets the score at the top

text("Lives: " + shipLife ,13,364);

for (var i = 0; i < alien2.length; i++) { // a for loop that prints out the aliens

alien2[i].draw();

ship.checkForAlien2(alien2[i]);

alien2[i].x += alien2[i].speed;

if (alien2[i].x >=350) { // the if statement causes the aliens to go faster when reaching one side of the game dimensions

alien2[i].speed = -2;

alien2[i].y += 25;

}

if (alien2[i].x <= 0) {

alien2[i].speed = 2;

alien2[i].y += 25;

}

}

for ( var i = 0; i < lasers.length; i++) { // a loop that goes through a laser

for (var j = 0; j < alien2.length;j++){

lasers[i].checkForAlien2(alien2[j]);}

lasers[i].draw();

lasers[i].y -= lasers[i].speed;

if (lasers[i].y < 0 ) {

lasers.splice(i,1);

}

}

if ( frameCount % 10 === 0){

if (keyIsPressed && keyCode === 0) {

lasers.push(new Laser(ship.x + 27, ship.y - 6));

}

}

for (var i = 0; i < alien3.length; i++) {// a for loop that prints out the aliens

alien3[i].draw();

ship.checkForAlien3(alien3[i]);

alien3[i].x += alien3[i].speed;

if (alien3[i].x >=350) { // the if statement causes the aliens to go faster when reaching one side of the game dimensions

alien3[i].speed = -3;

alien3[i].y += 25;

}

if (alien3[i].x <= 0) {

alien3[i].speed = 3;

alien3[i].y += 35;

}

}

for ( var i = 0; i < lasers.length; i++) {

for (var j = 0; j < alien3.length;j++){

lasers[i].checkForAlien3(alien3[j]);}

lasers[i].draw();

lasers[i].y -= lasers[i].speed;

if (lasers[i].y < 0 ) {

lasers.splice(i,1);

}

}

if ( frameCount % 10 === 0){

if (keyIsPressed && keyCode === 0) { // states the key code assigned for the spacebutton so that lasers are fired when space button is pressed.

lasers.push(new Laser(ship.x + 27, ship.y - 6));

}

}

if (keyIsPressed && keyCode === 37){ // movement for spaceship in the left direction

ship.left();

}

if (keyIsPressed && keyCode === 39){ // movement for spaceship in the right direction

ship.right();

}

if (keyIsPressed && keyCode === 38){ // movement for spaceship in the forward direction

ship.up();

}

if (keyIsPressed && keyCode === 40){ // movement for spaceship in the reverse direction

ship.down();

}

ship.draw();

// boolean creates condition in which the gameWin text pops up and the game is completed

var moveToWinScene = true;

for ( var i = 0; i < alien2.length; i++){ // a for loop that runs through alien2 array stating if alien2 is false if alien2's y position is less than 400

if(alien2[i].y < 400){

moveToWinScene = false;

}

}

for ( var i = 0; i < alien3.length; i++){ // a for loop that runs through alien2 array stating if alien2 is false if alien3's y position is less than 400

if(alien3[i].y < 400){

moveToWinScene = false;

}

}

if (moveToWinScene && shipLife > 0 && currentScene !== 0){

rectMode(CORNER);

btn4.draw();

fill(255, 0, 0);

text("You Win! Thanks for playing!",32,82);

textSize(15);

text("Thank you to all faculty that helped us learn to program. Much appreciated!", 77, 154, 236, 100);

}

else if (shipLife <= 0) {

rectMode(CORNER);

fill(255, 0, 0);

textSize(40);

text("Game Over", 79,200);

btn4.draw();

}

};

mouseClicked = function()

{

if (currentScene === 0)

{

// button for start game button

btn1.handleMouseClick();

// Go to instructions scene

btn2.handleMouseClick();

}

else if (currentScene === 2)

{

btn3.handleMouseClick();

}

else if (currentScene === 3 || currentScene === 1)

{// allows the game to restart if gameOver or gameWin

if (btn4.handleMouseClick())

{

Program.restart();

}

}

};

draw = function() {

if (currentScene === 0)

{

splashScreen();

}

if (currentScene === 1)

{

gameScene();

}

if (currentScene === 2)

{

background(255, 0, 0);

instructions();

}

// Set a variable equal to true,

// look thoruh all aliens, if any are y < 400, set the varialbe each to false

// if the variable is true && lives > 0 , move on to scene 3 (e.g., gameScene2(); )

var moveOnToScene3 = true;

for ( var i = 0; i < alien1.length; i++){

if(alien1[i].y < 400){

moveOnToScene3 = false;

}

}

if(moveOnToScene3 && shipLife > 0 && currentScene !== 0){

gameScene3(); // calling scene3

if ( shipLife <=0)

{

fill(0,0,0);

rect(6, 329, 119, 57);

}

}

};