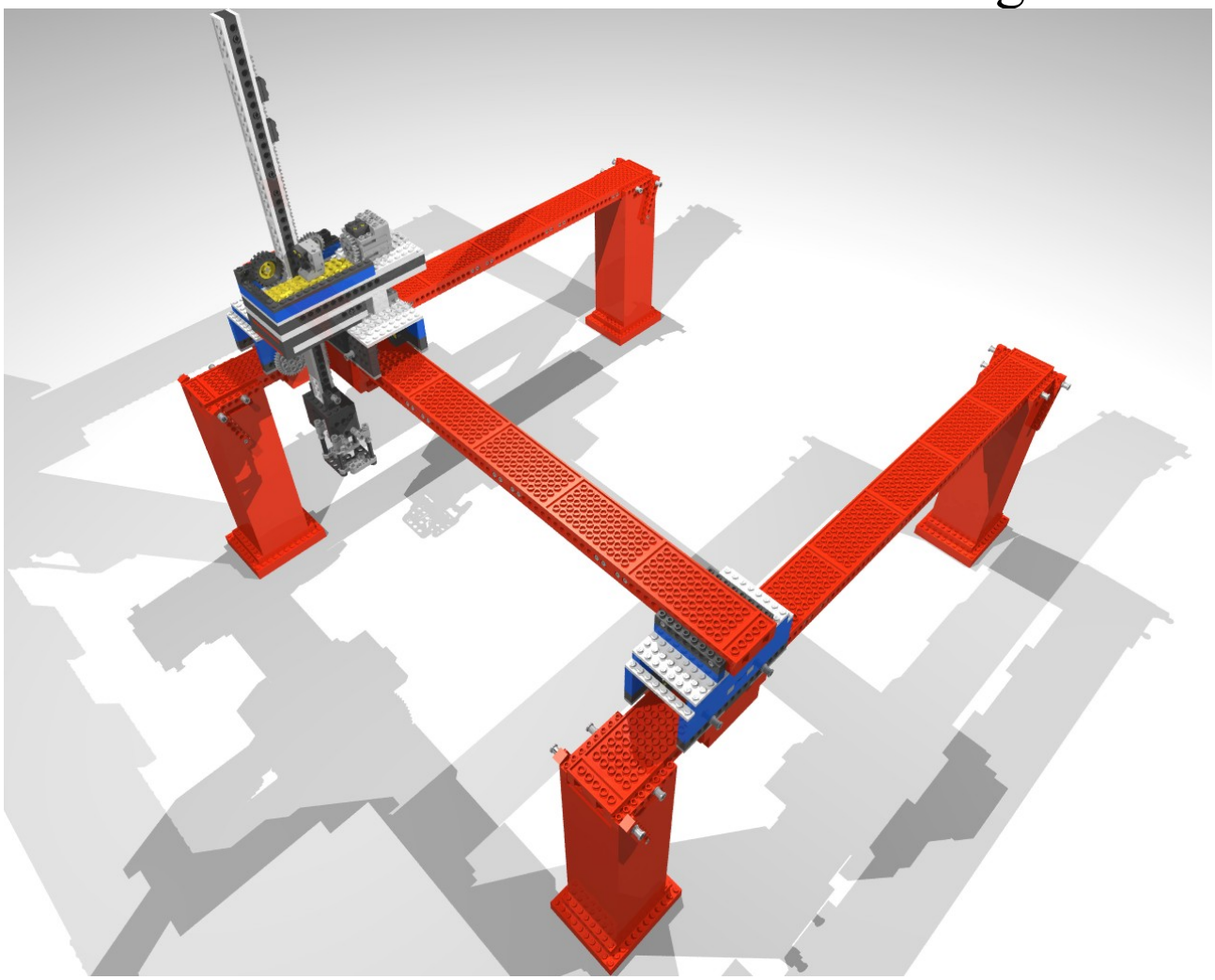
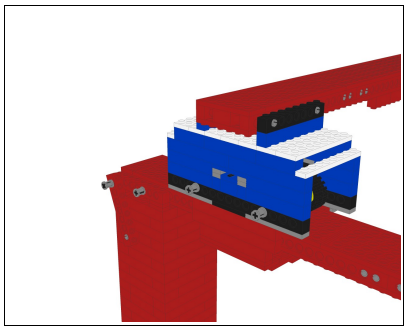
SCHACKROBOT

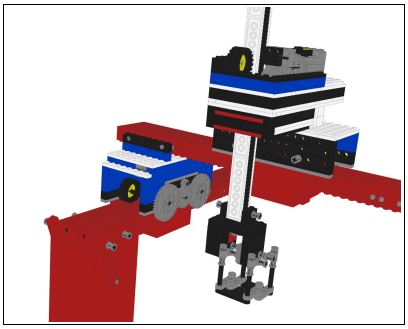
Ett specialarbete av

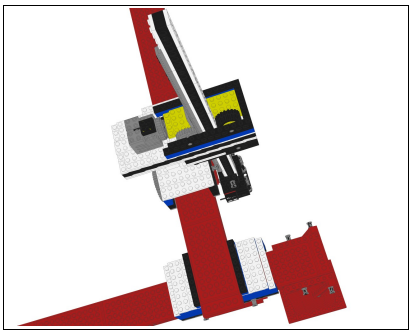
Jonas Forsslund och Johan Friberg



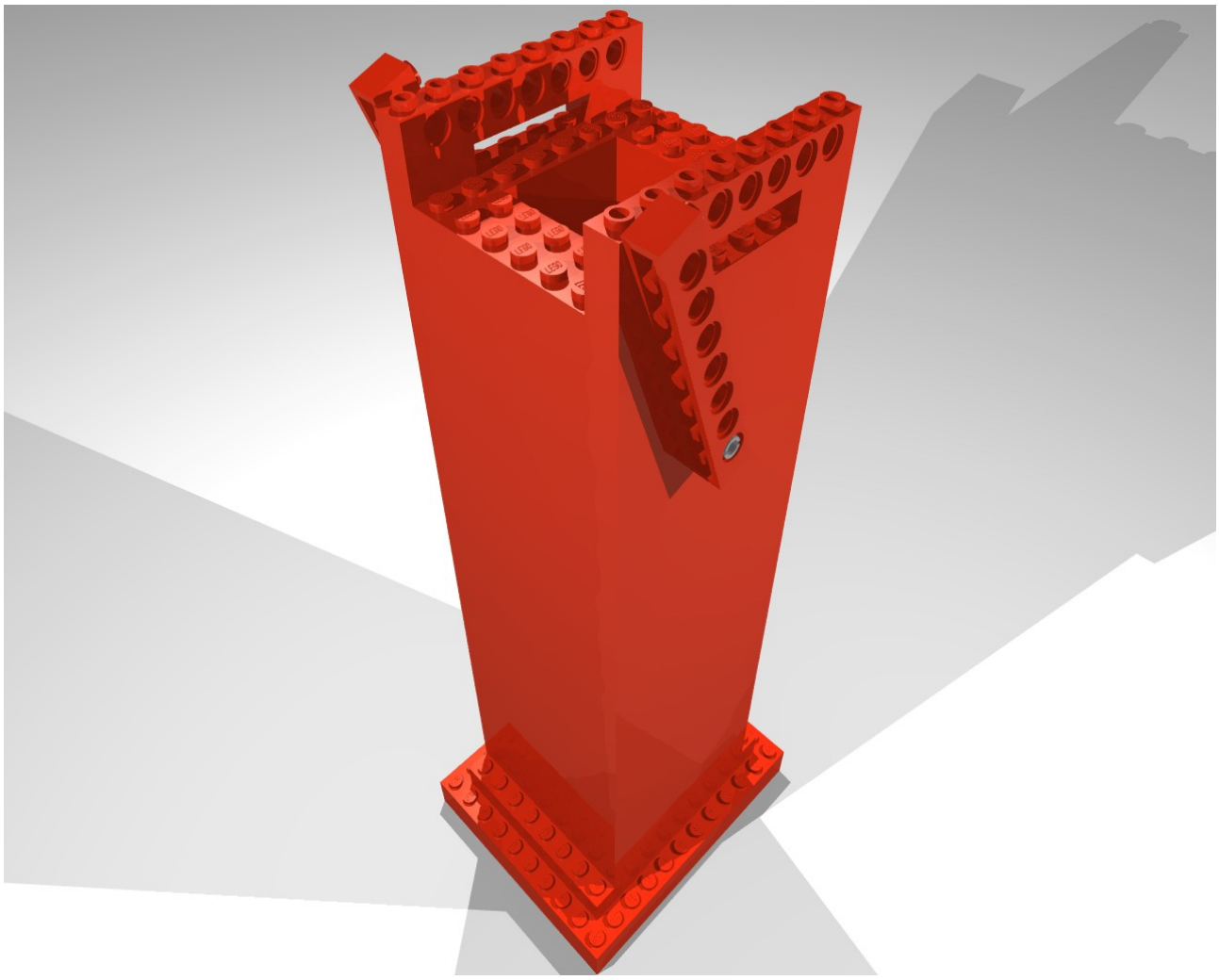
Christopher Polhemsgymnasiet 2002. Handledare Torbern Stålek.



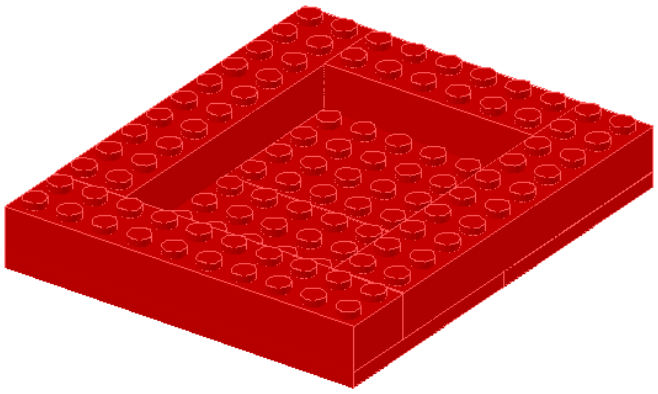




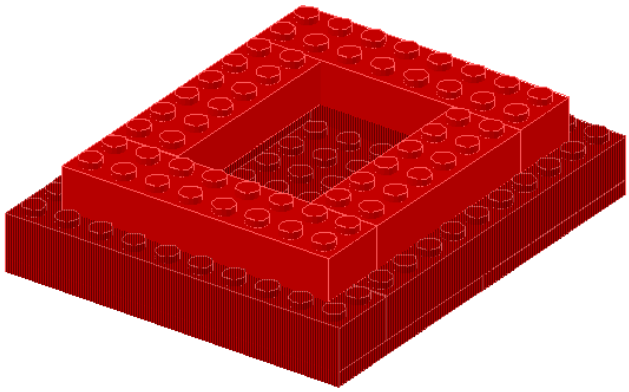
4x Hörnpelare



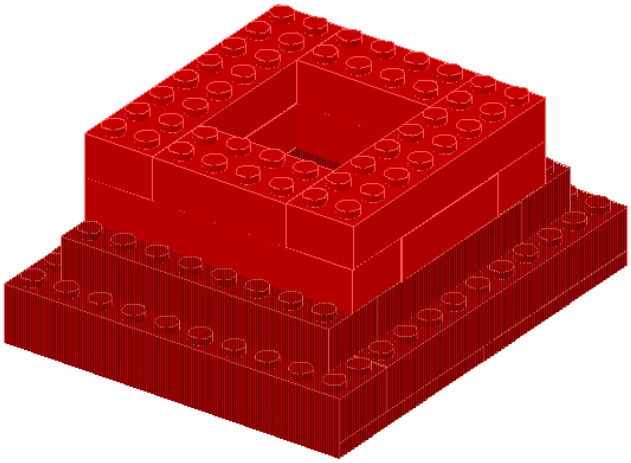
1



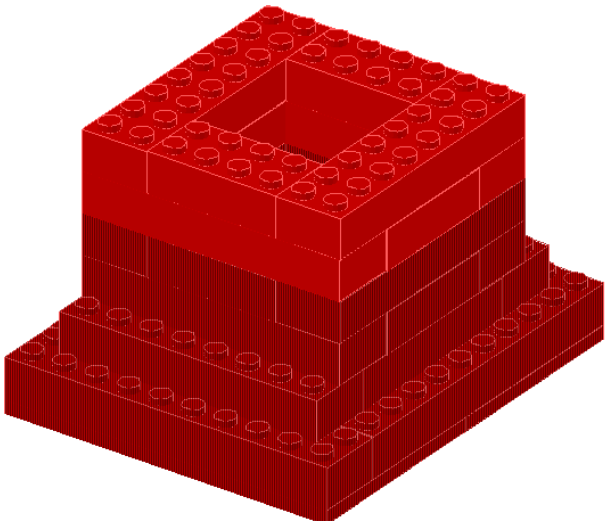
2



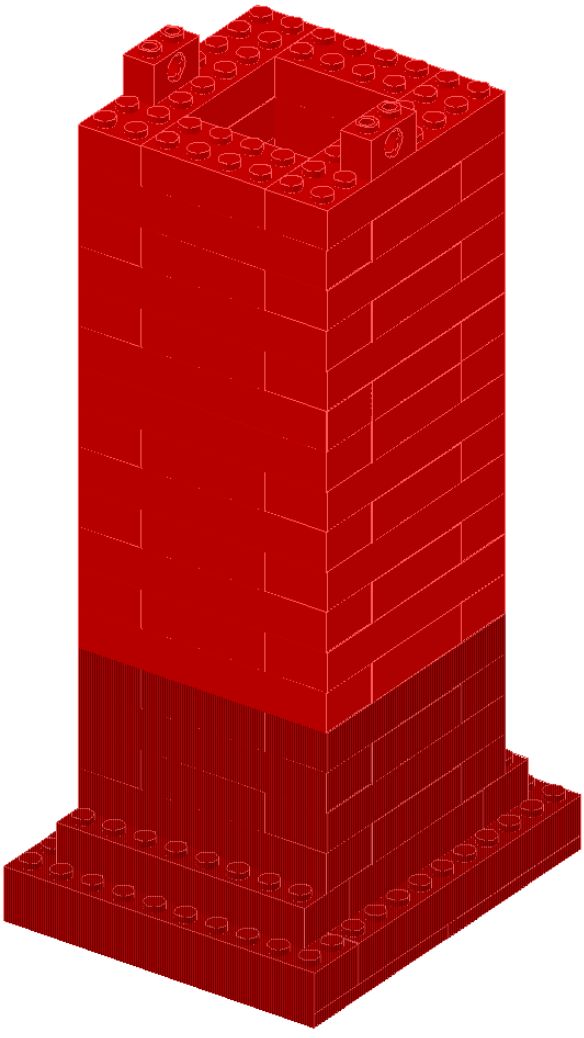
3



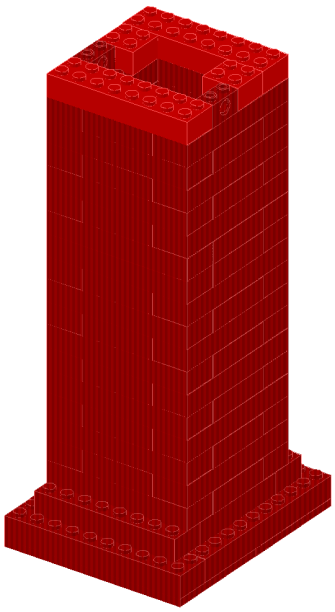
4



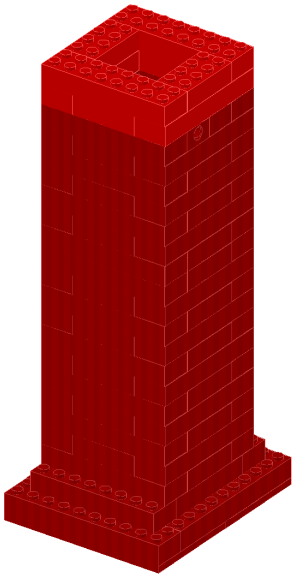
5



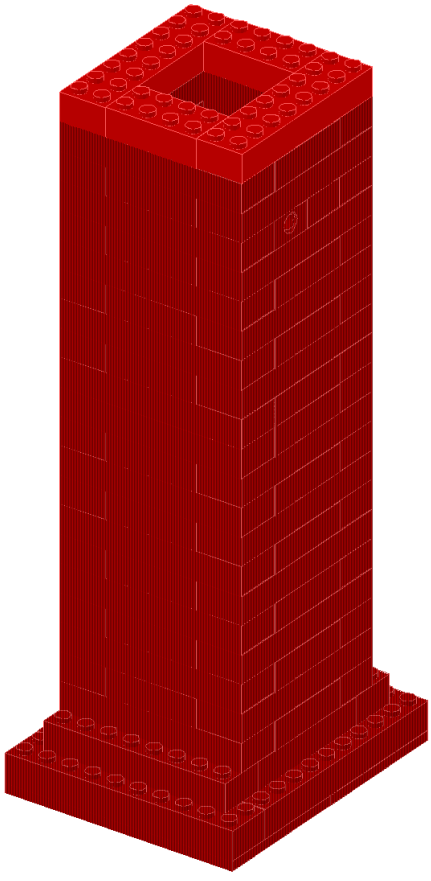
6



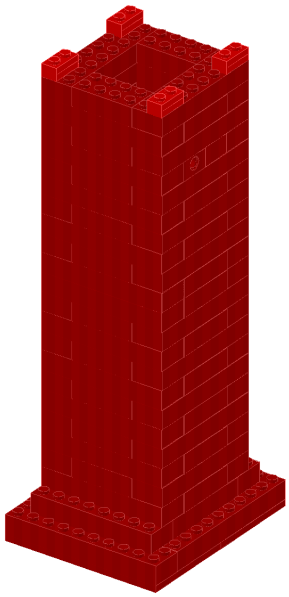
1



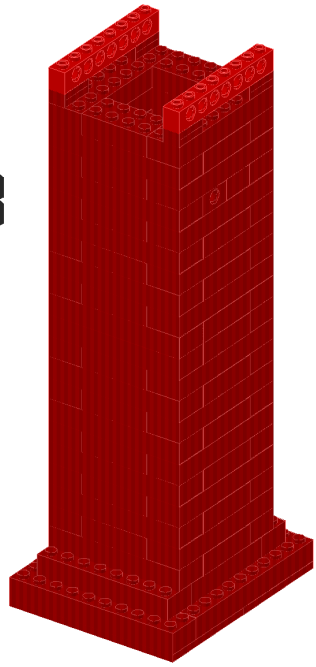
8



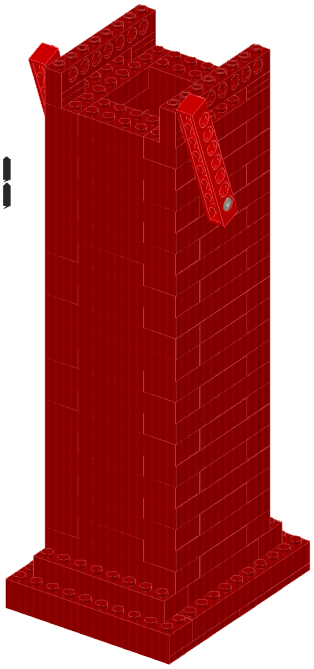
9



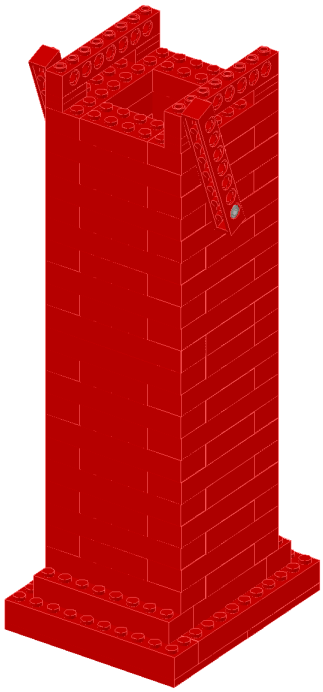
10



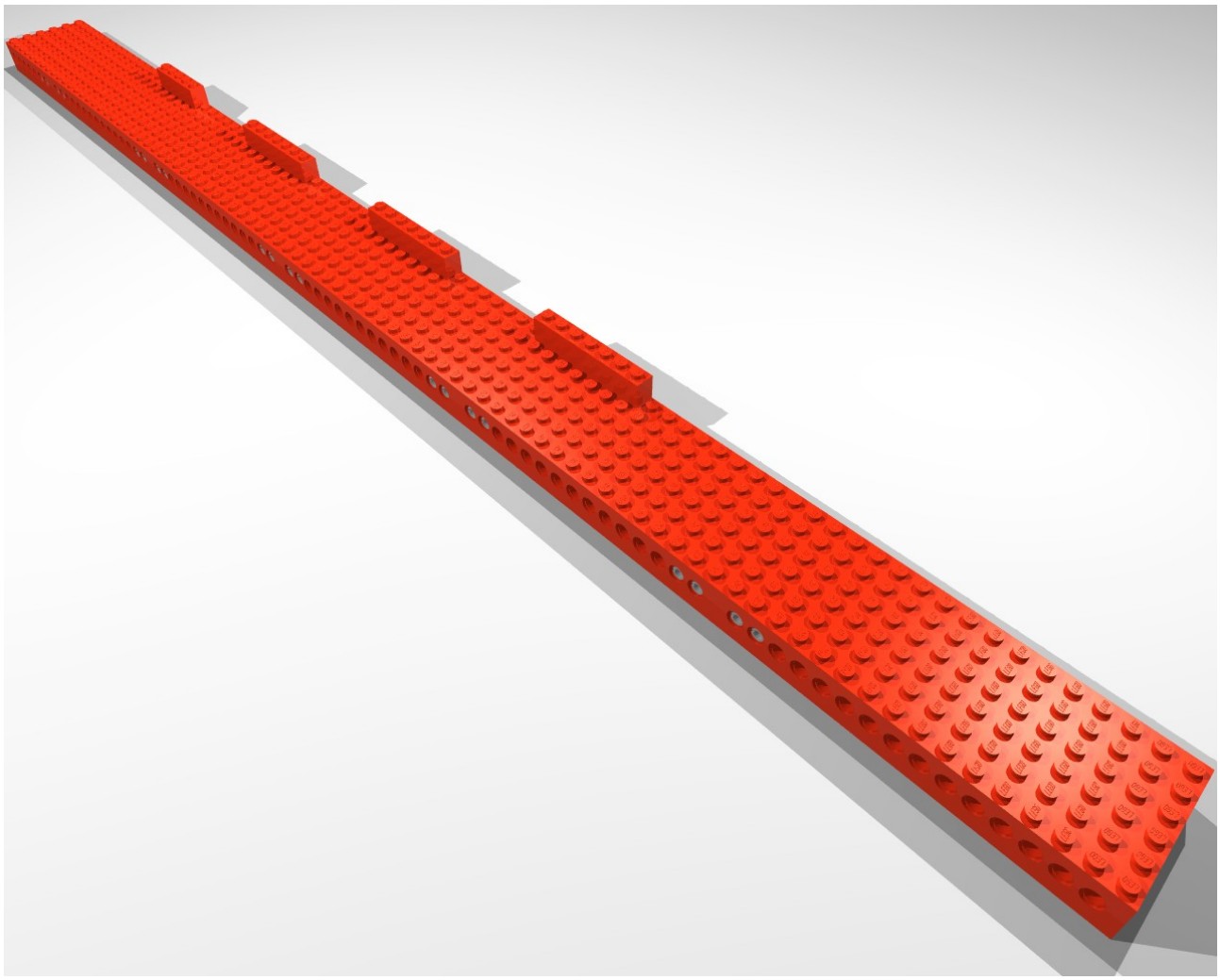
11



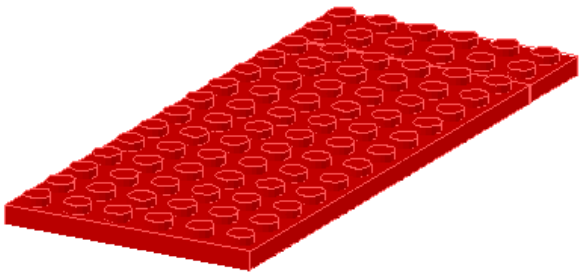
12



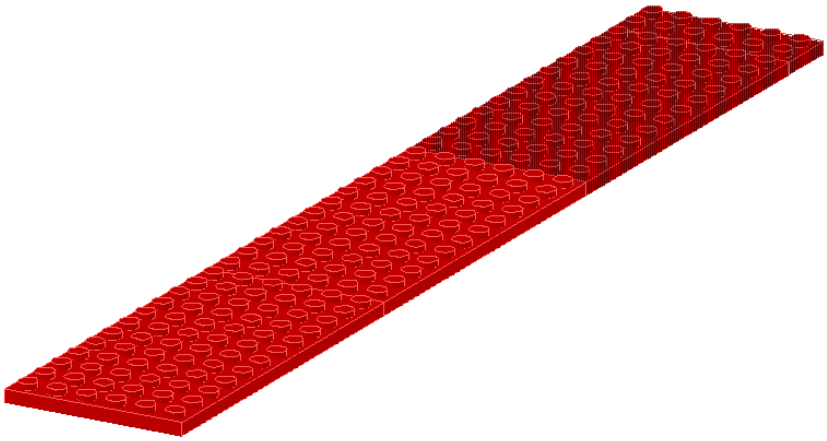
1x Kantbro med sensorbitar
1x Kantbro utan sensorbitar



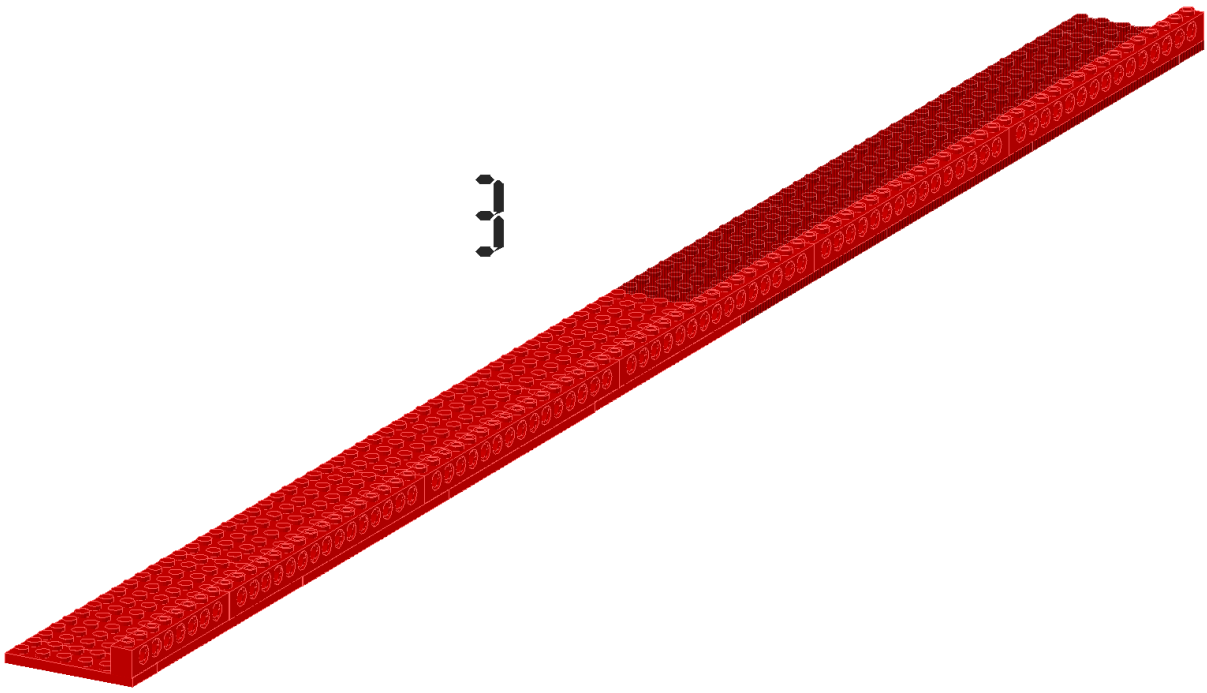
1



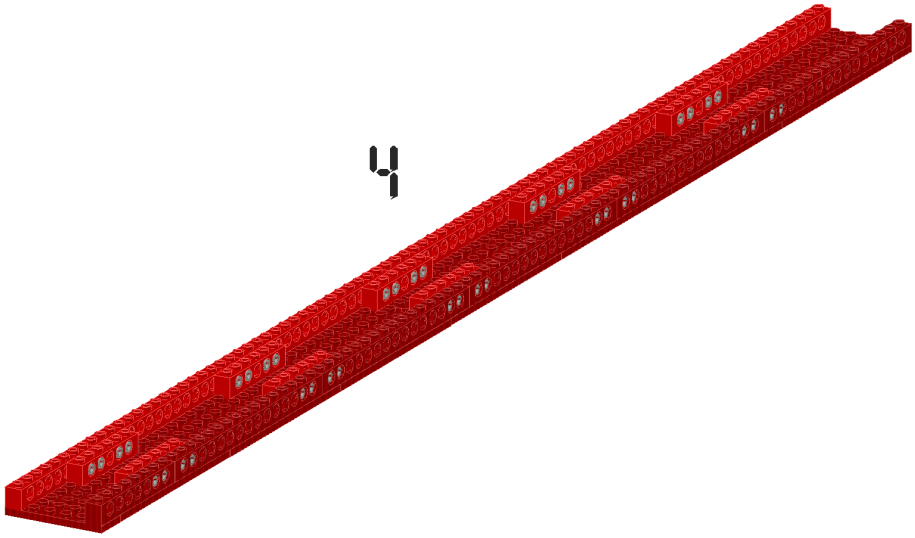
2



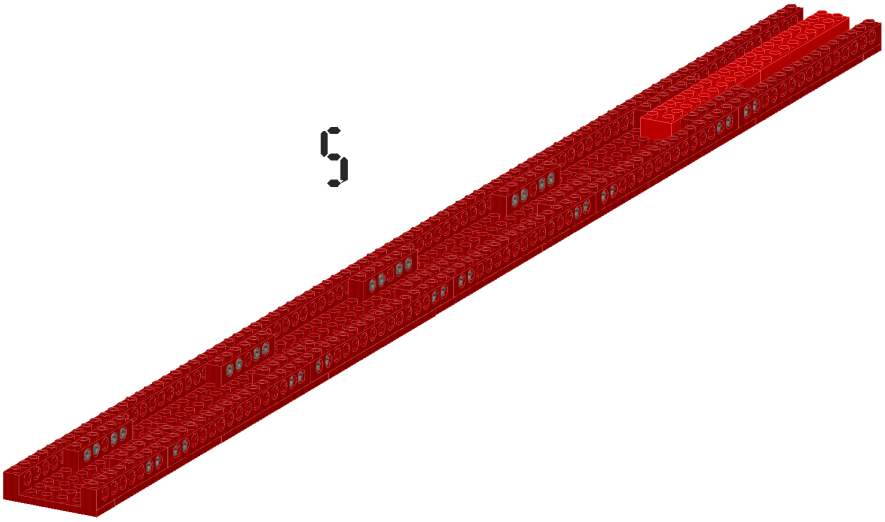
3



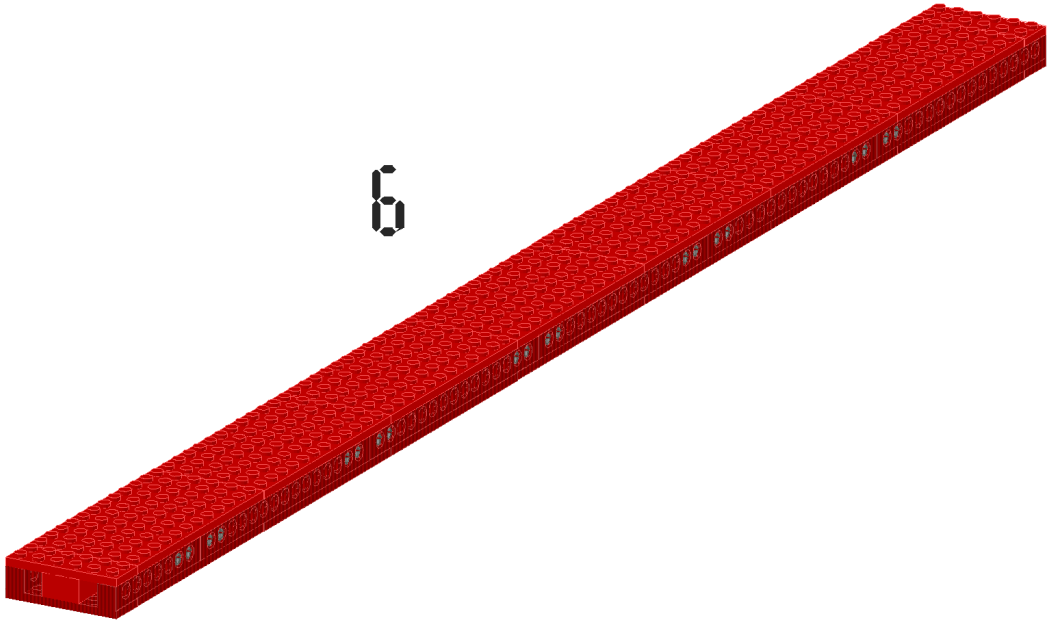
4



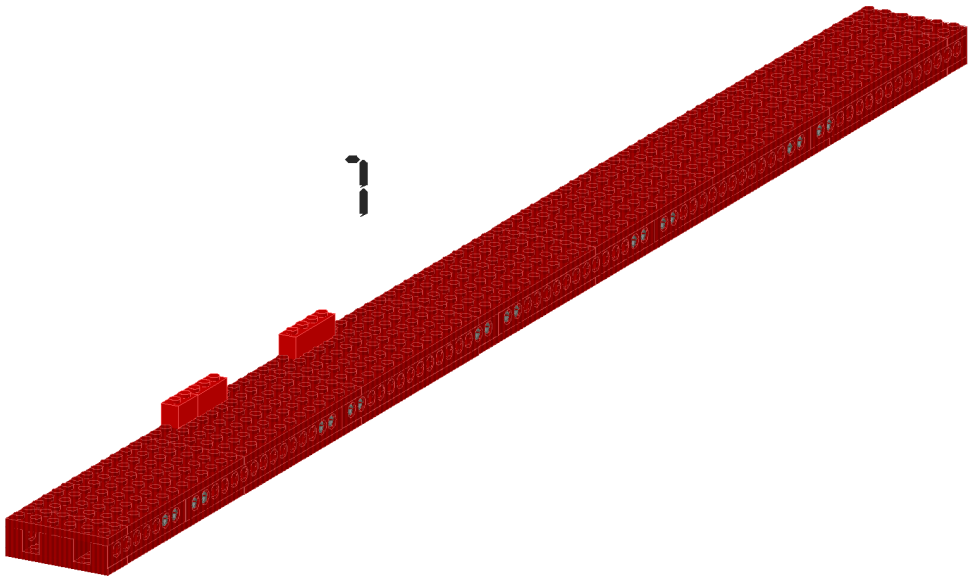
5



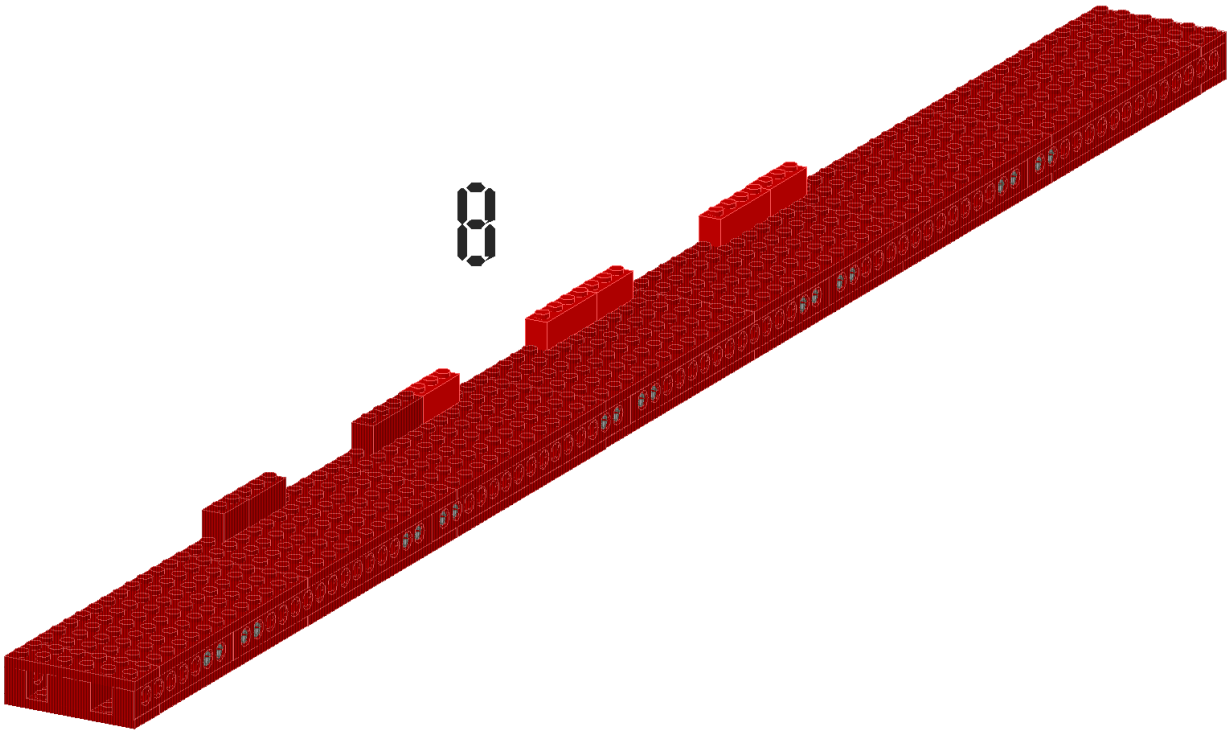
6



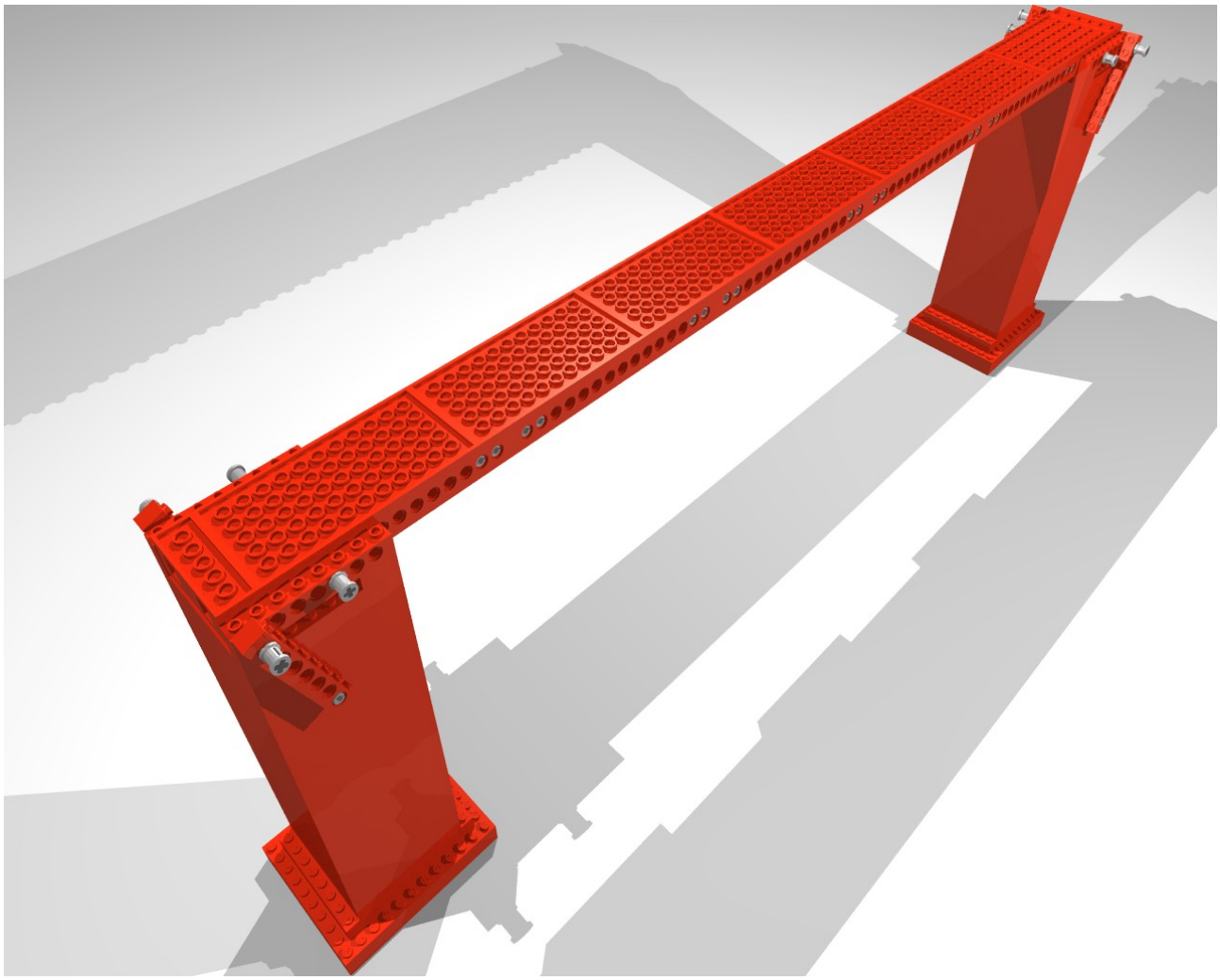
1



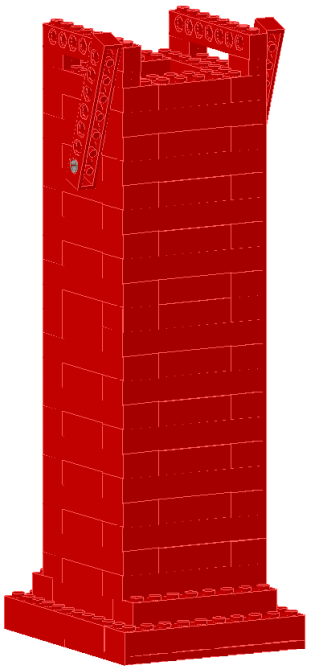
8



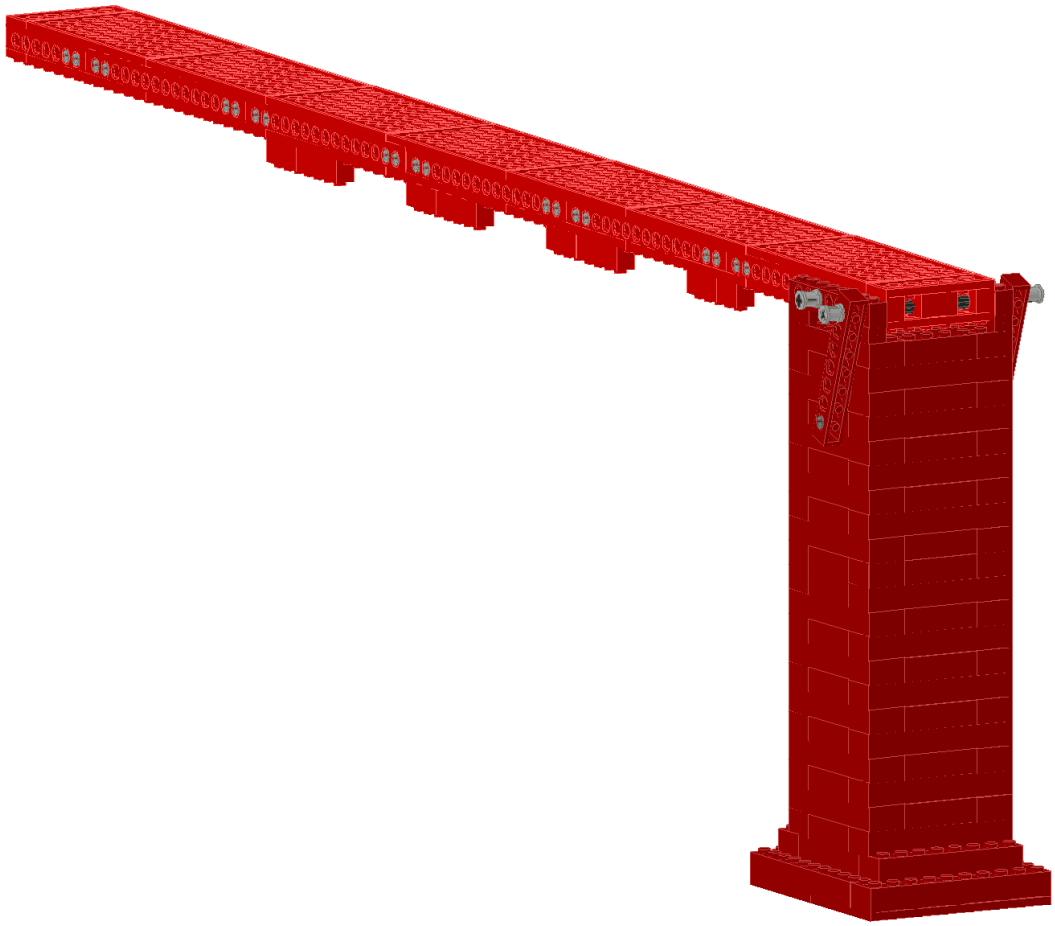
1x Stome med sensorbitar
1x Stome utan sensorbitar



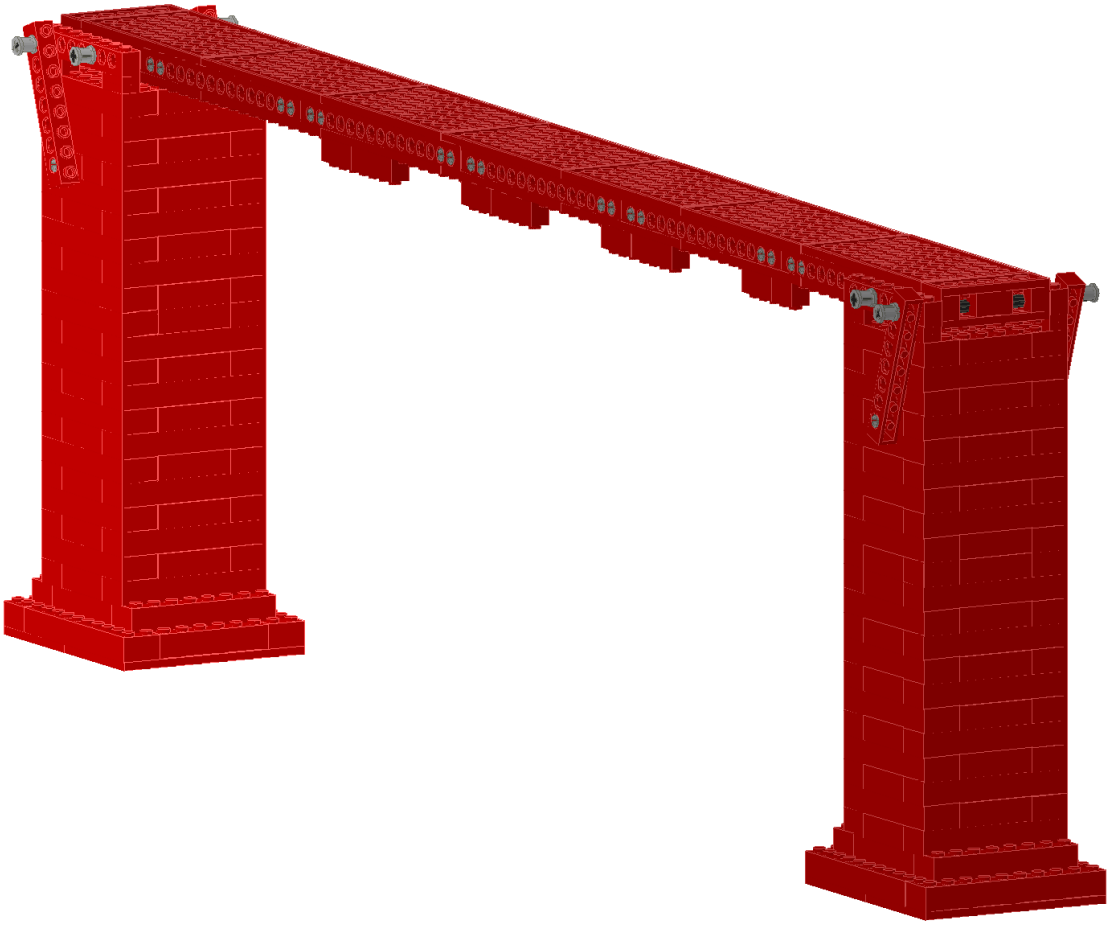
1



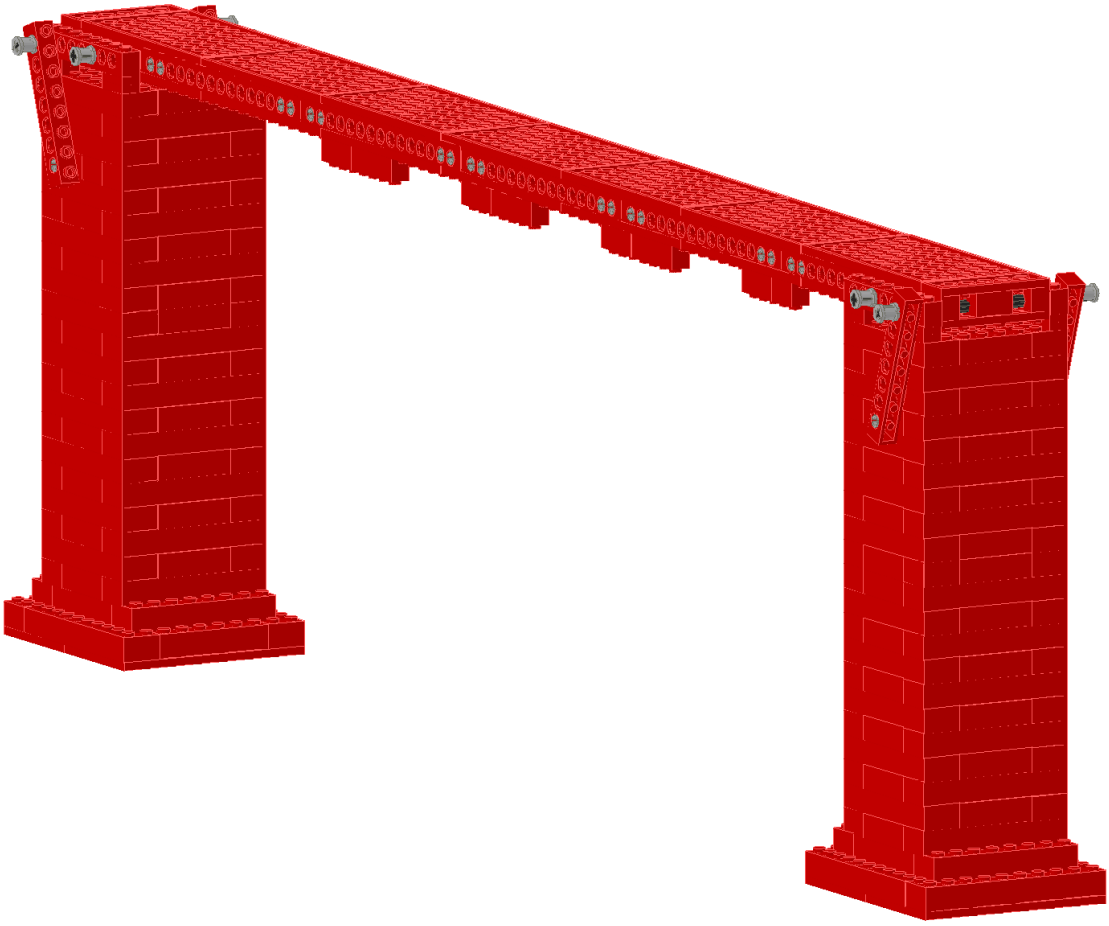
2



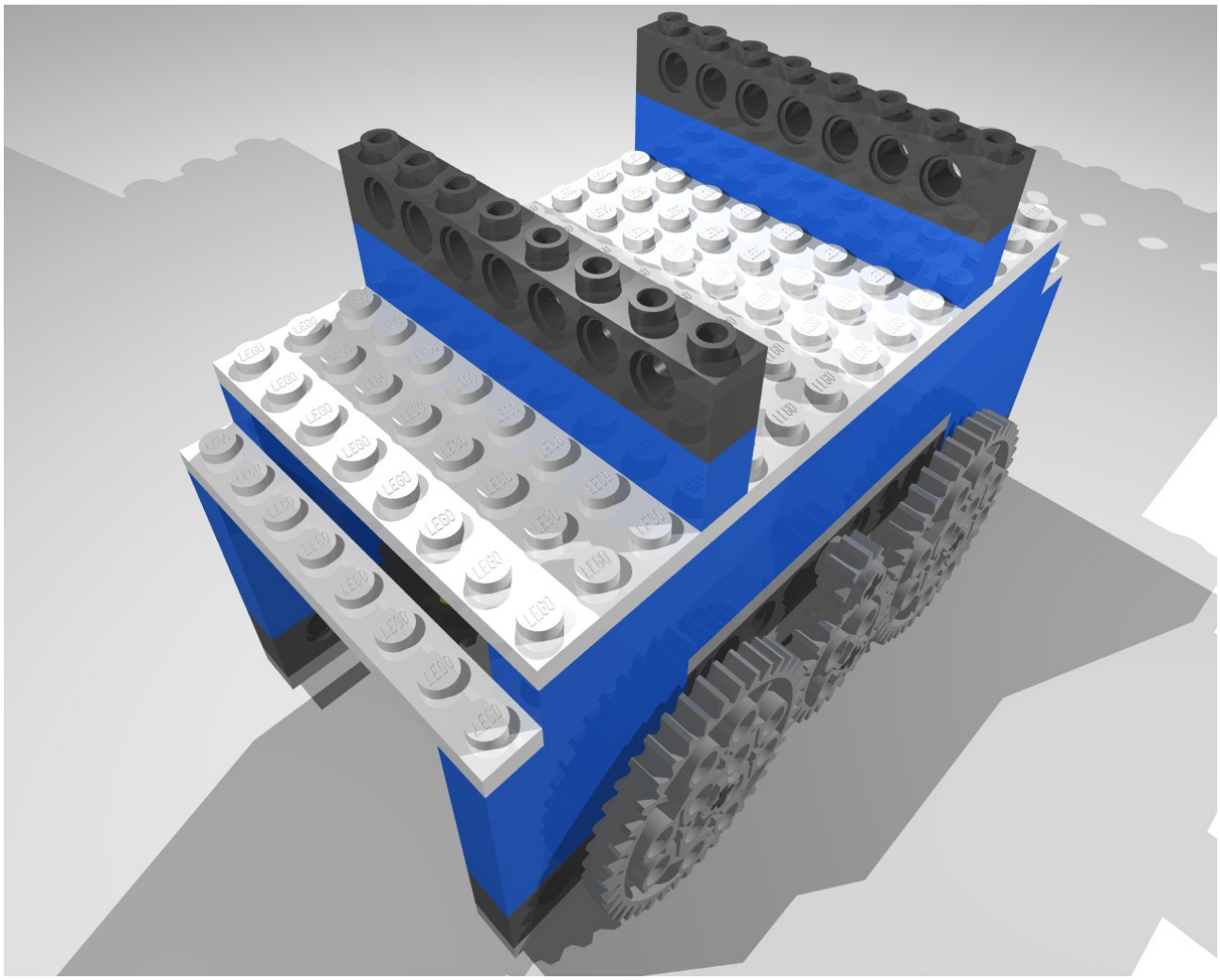
3



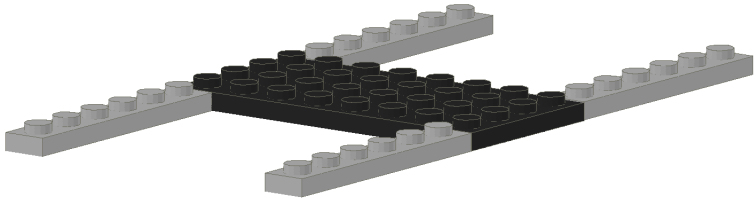
4



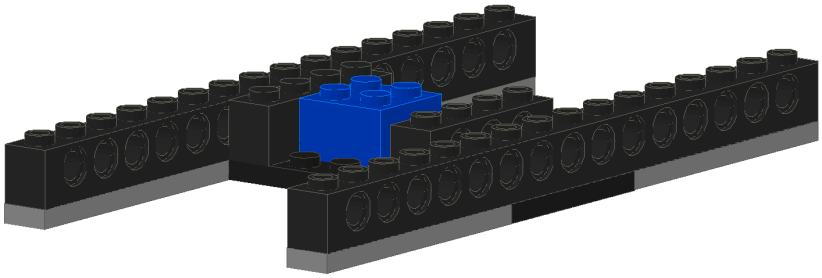
1x Vagn Y



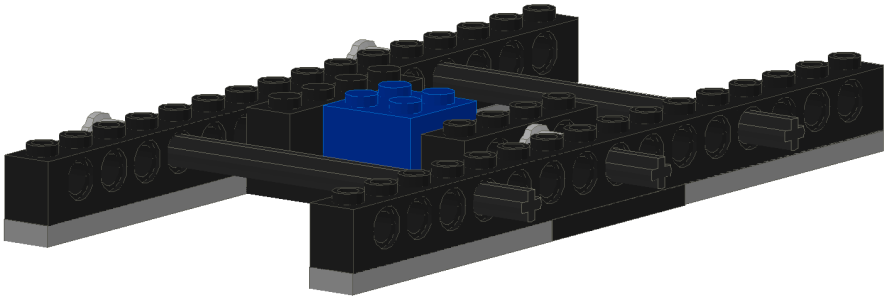
1



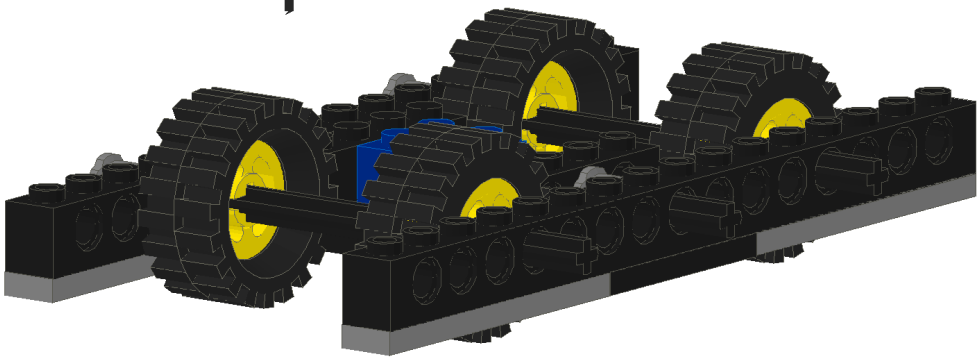
2



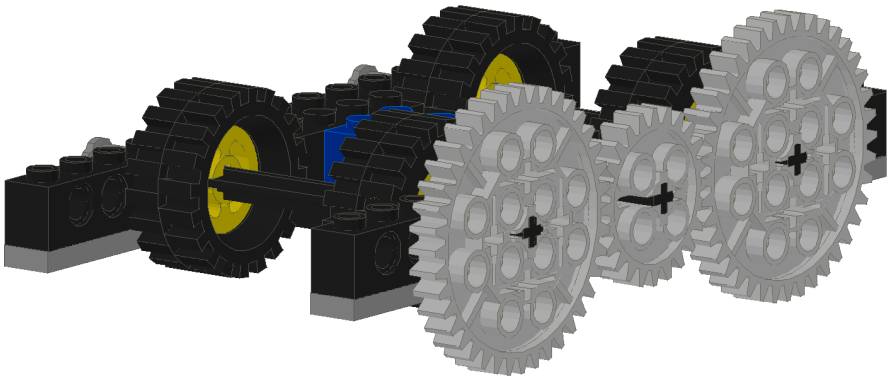
3



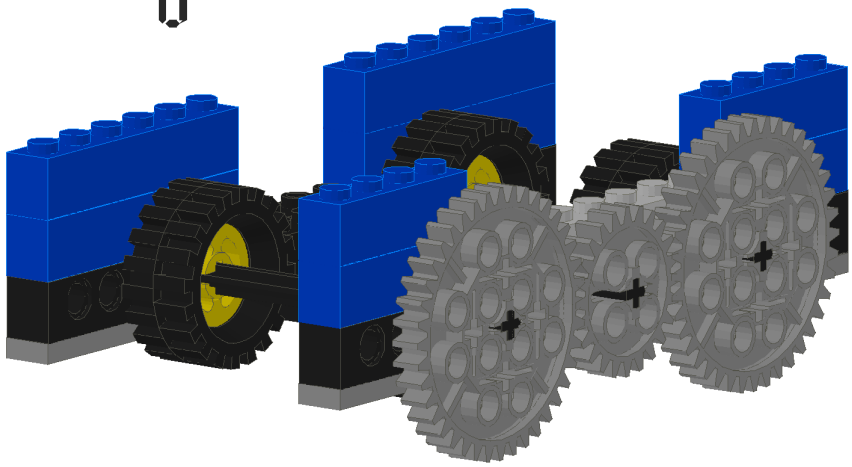
4



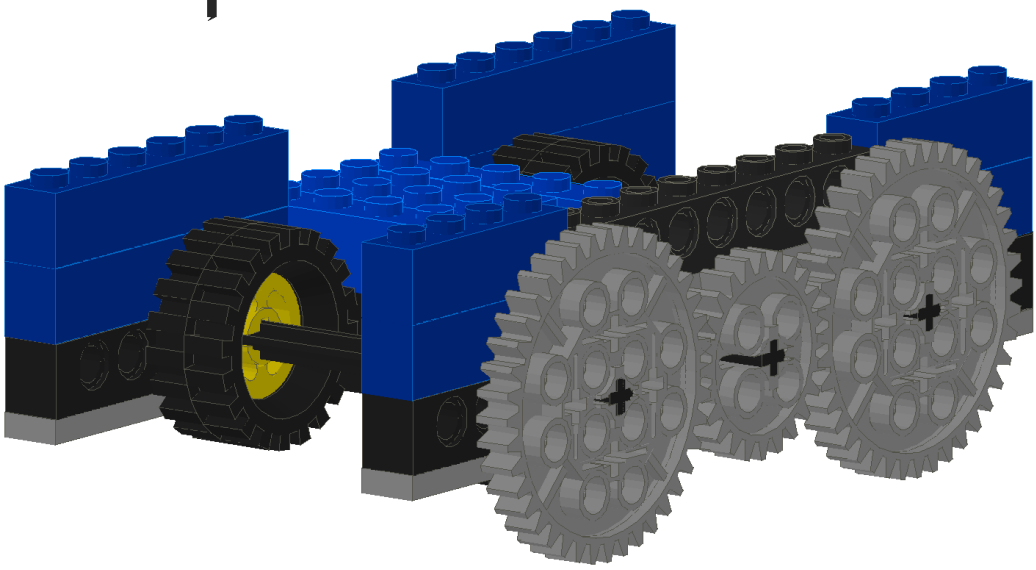
5



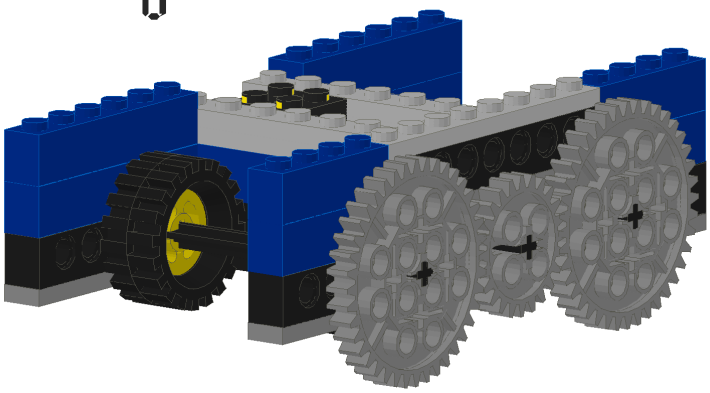
6



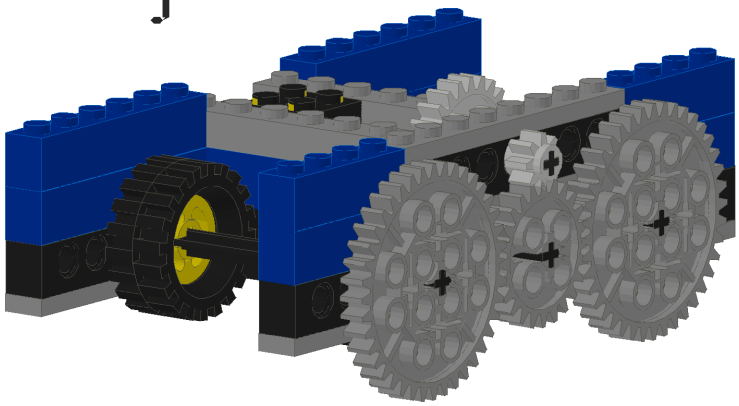
7



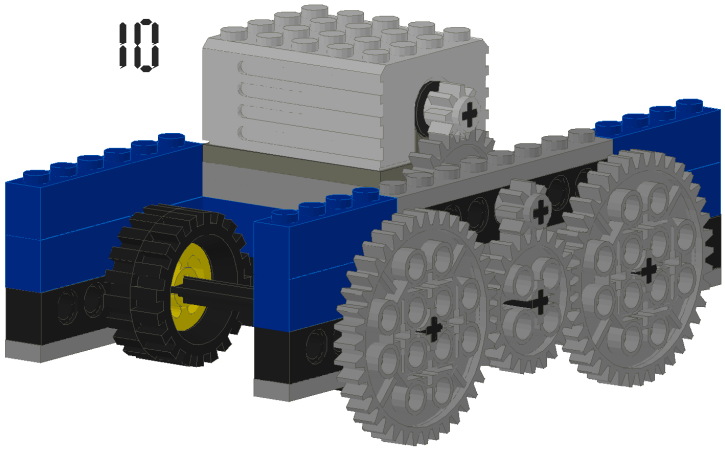
8



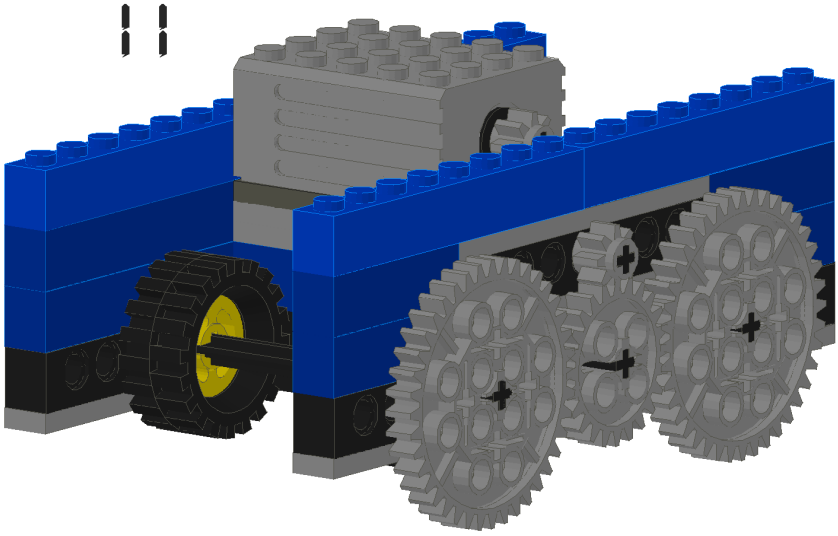
9



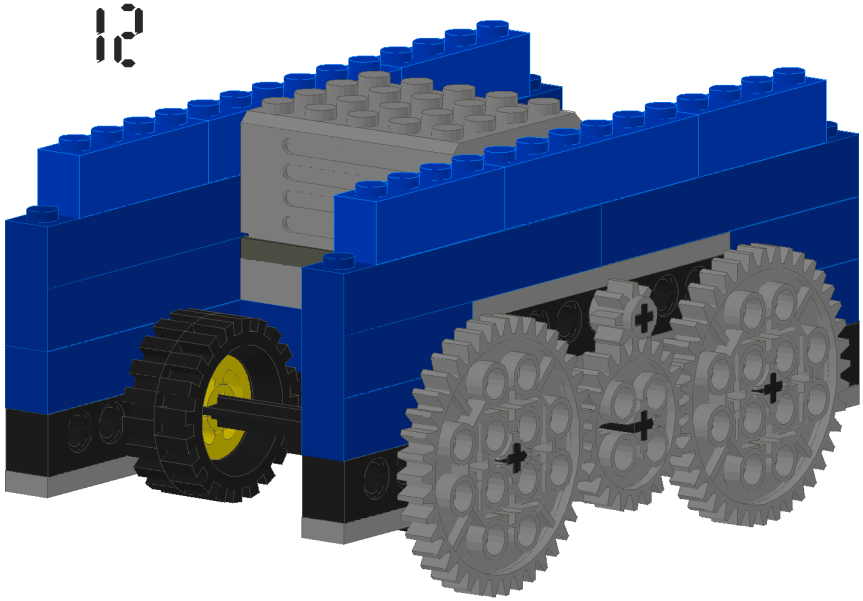
10



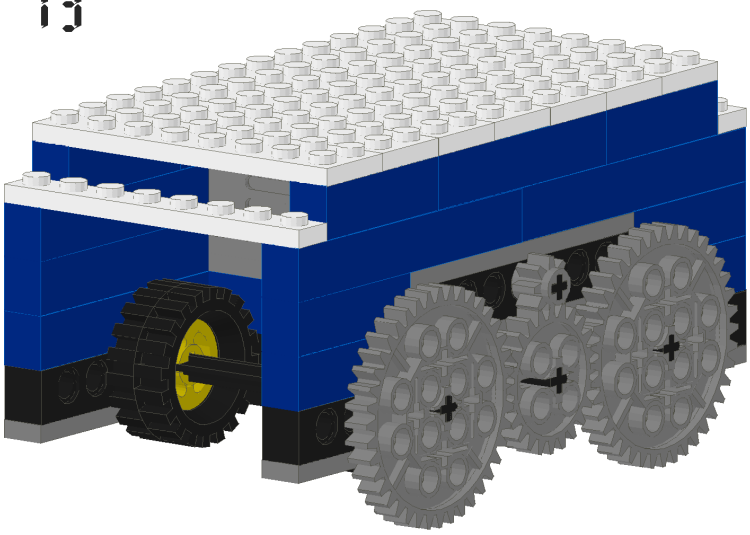
11



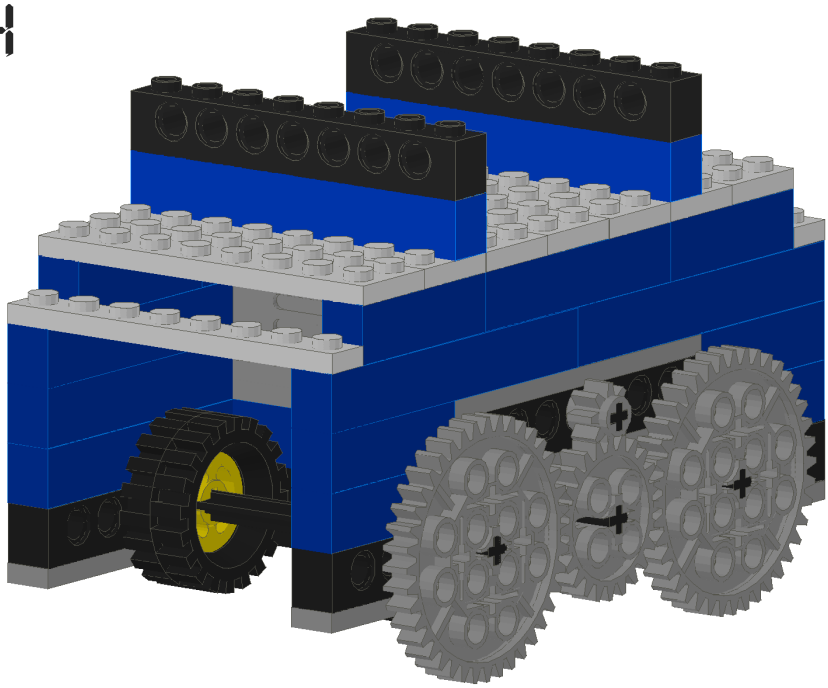
12



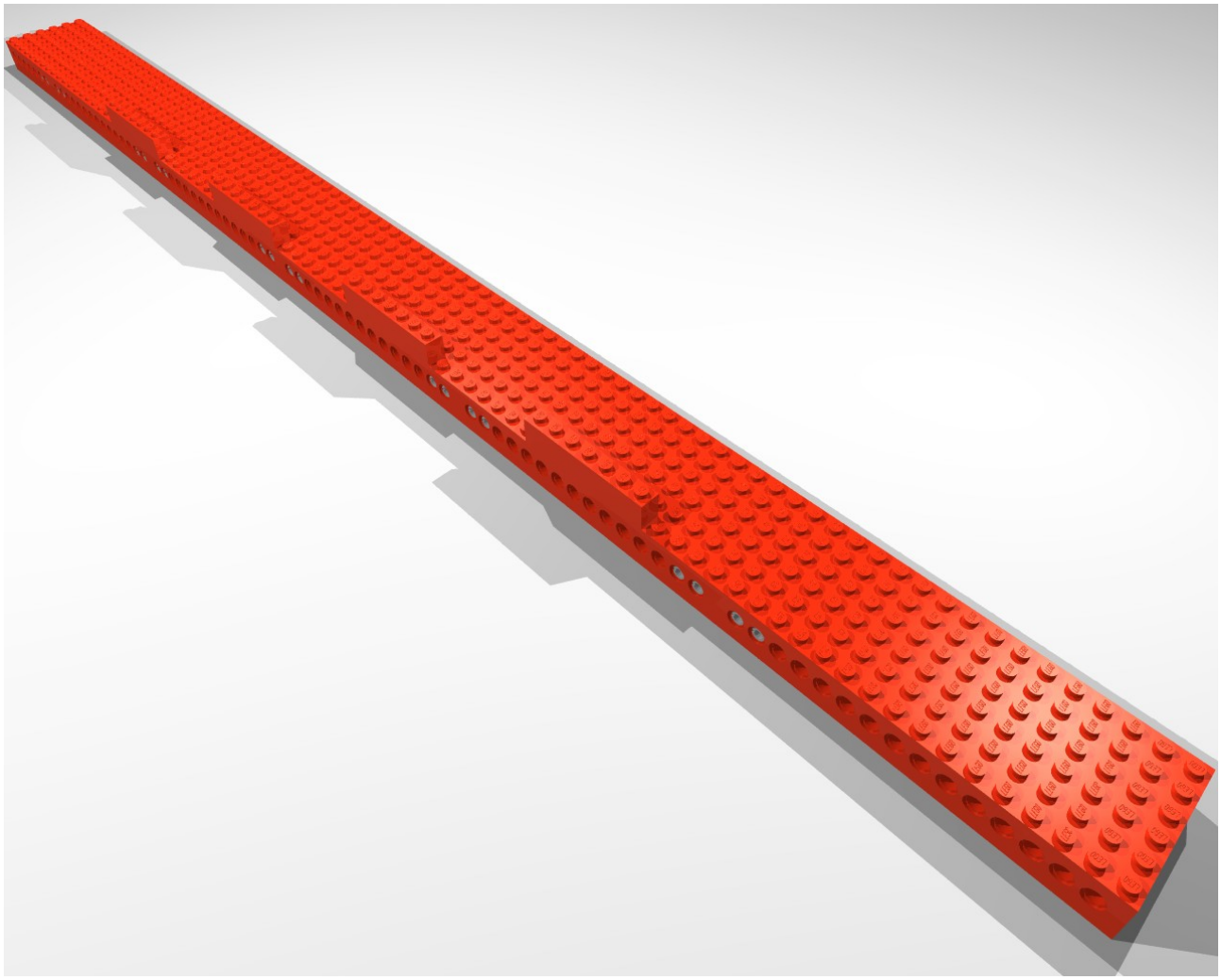
13



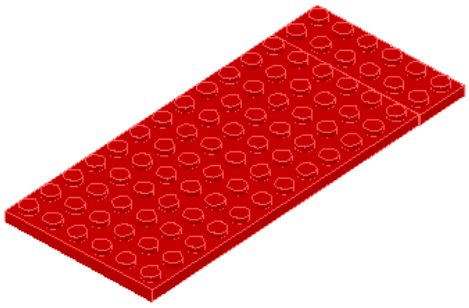
14



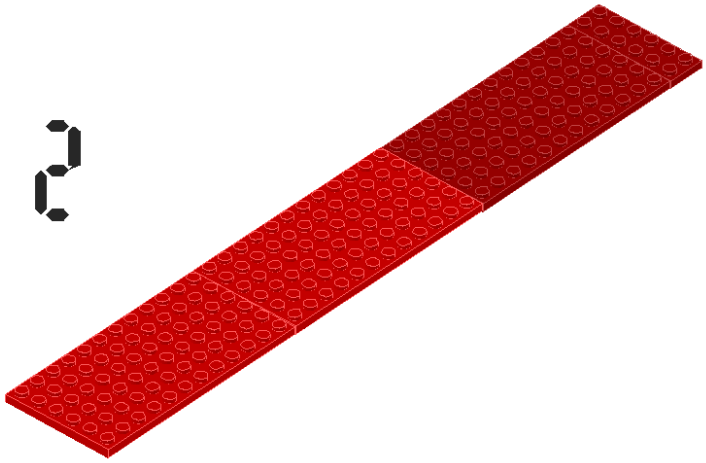
Tvärgående bro



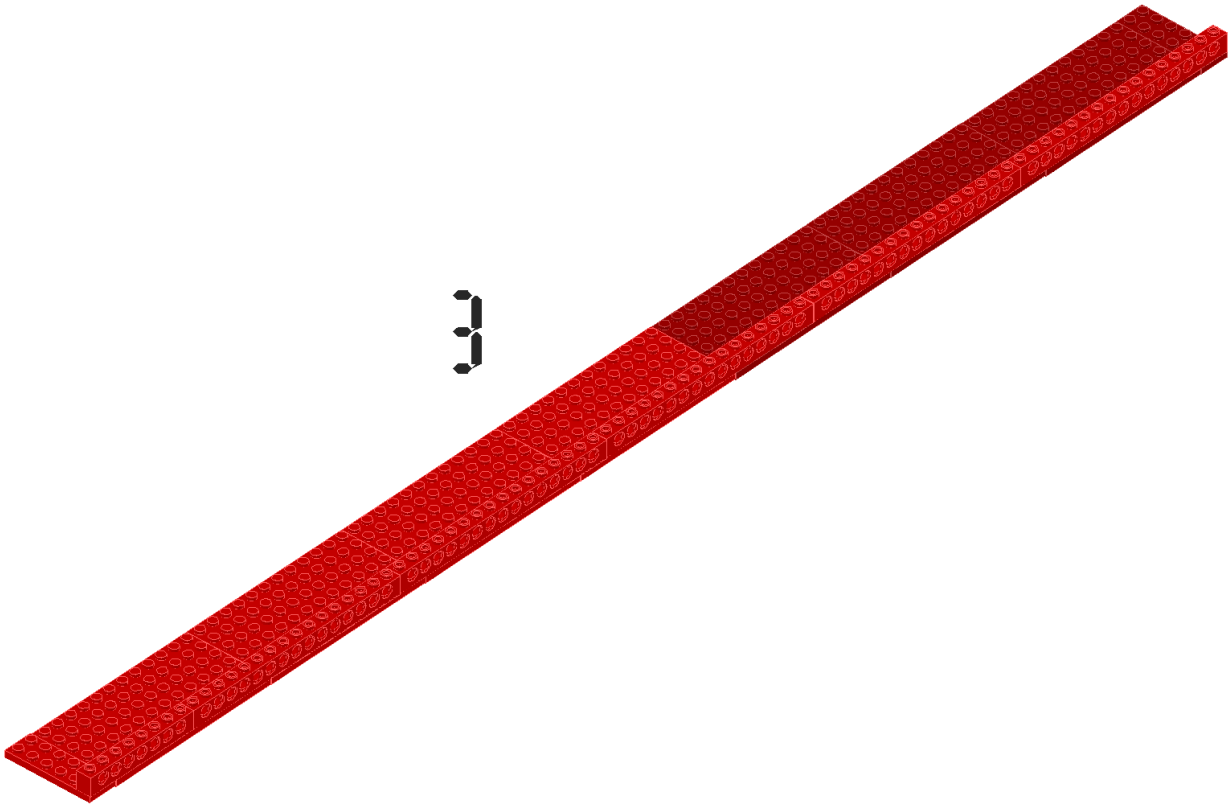
1



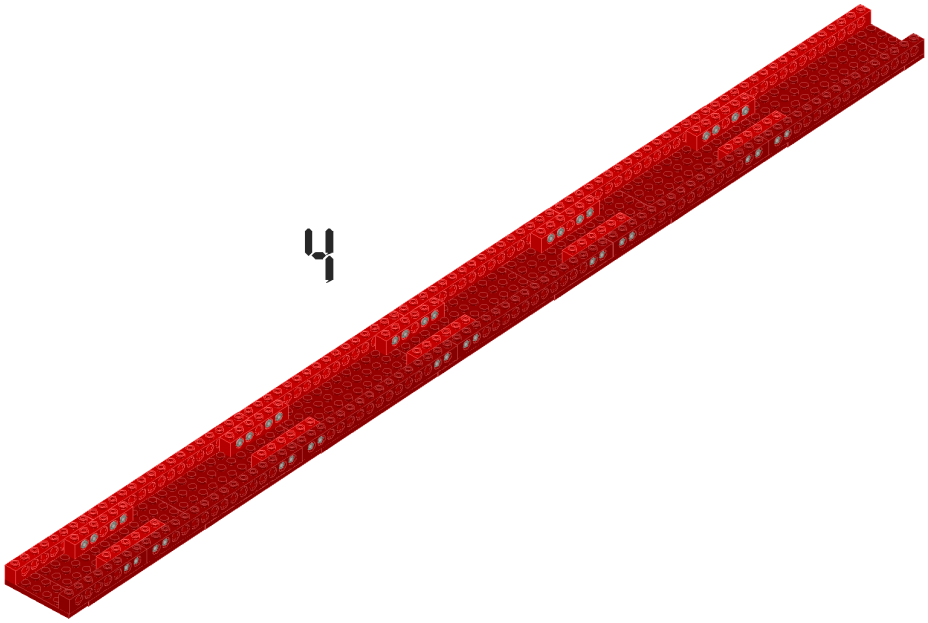
2



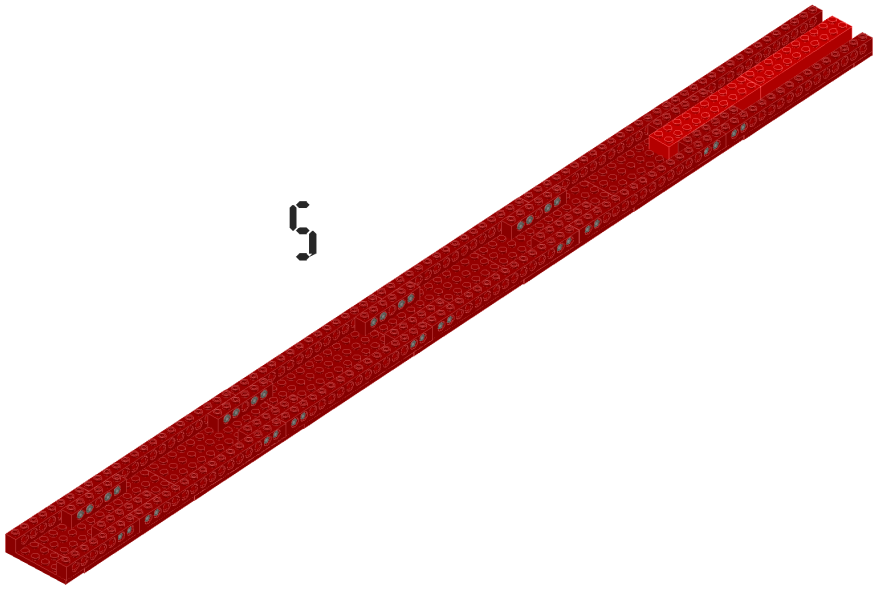
3



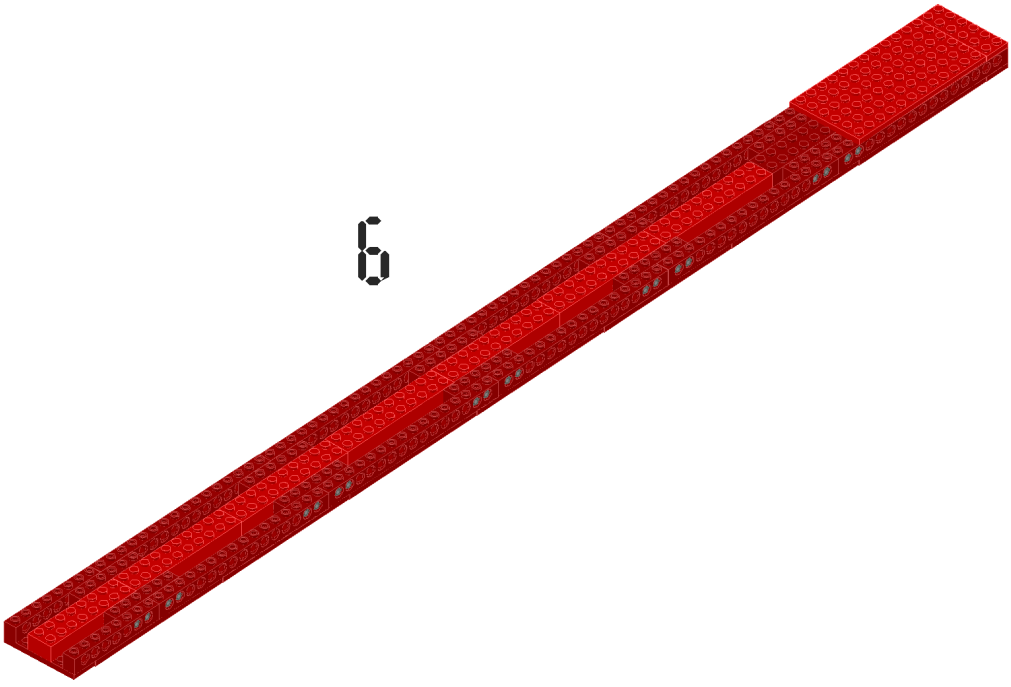
4



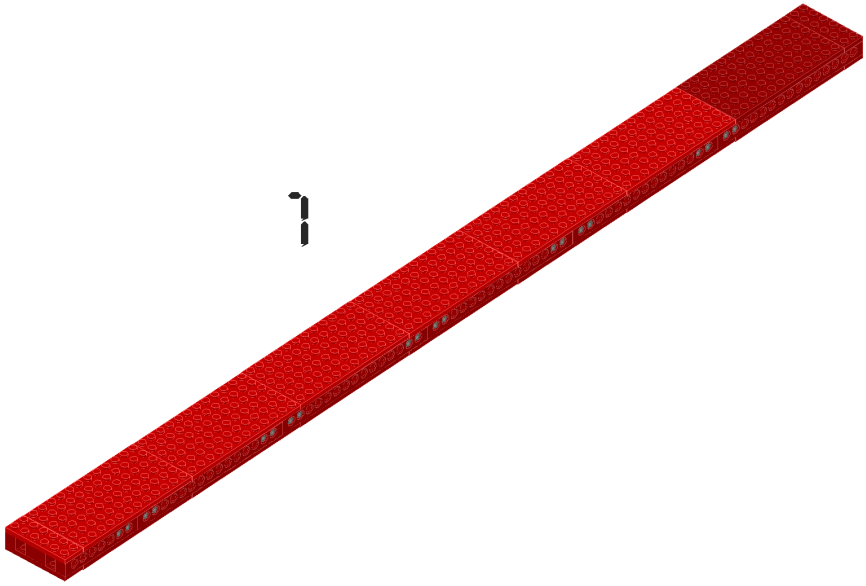
5



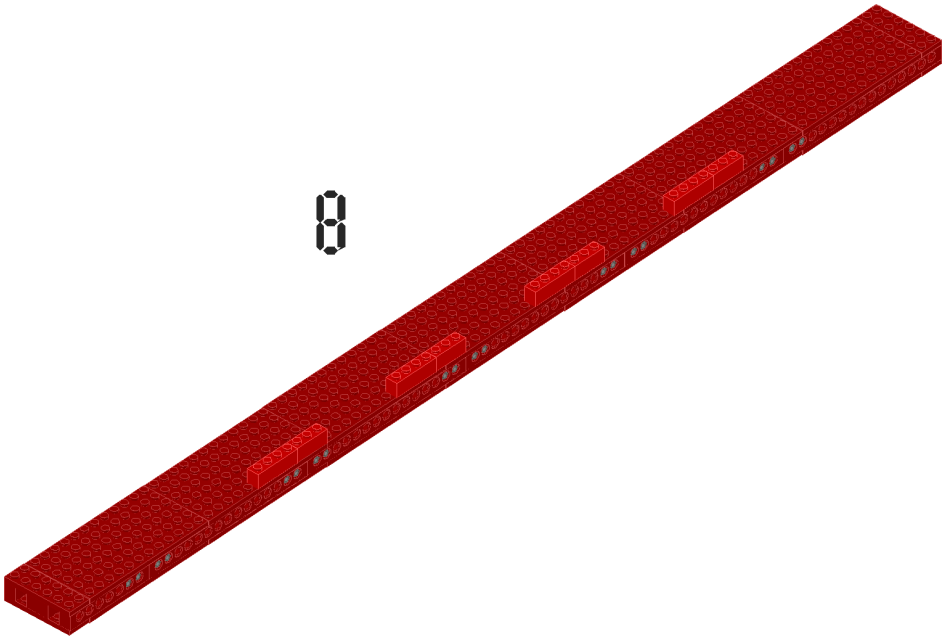
6



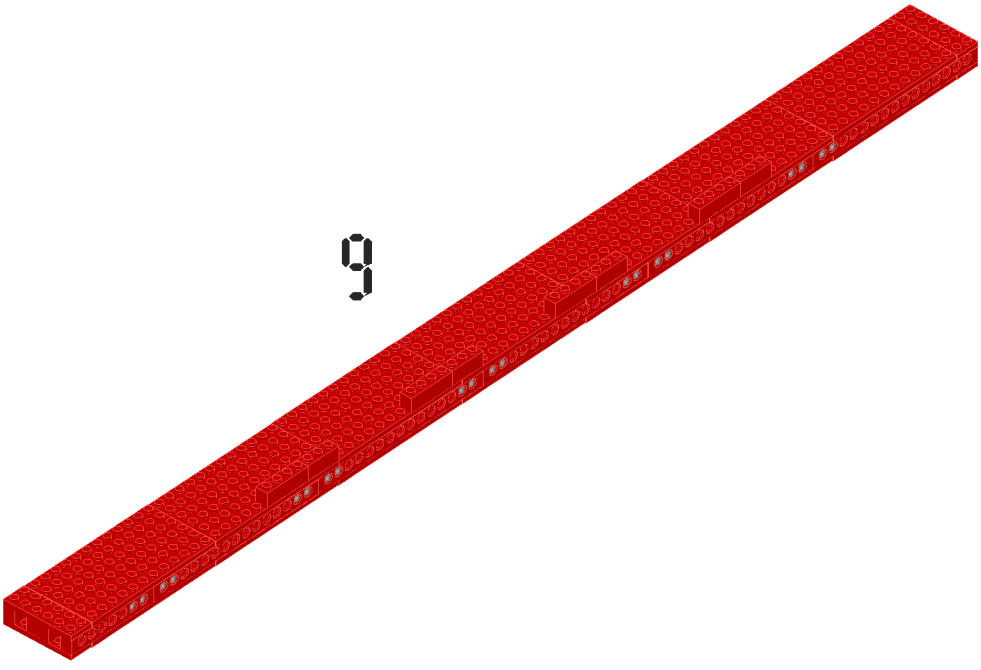
1



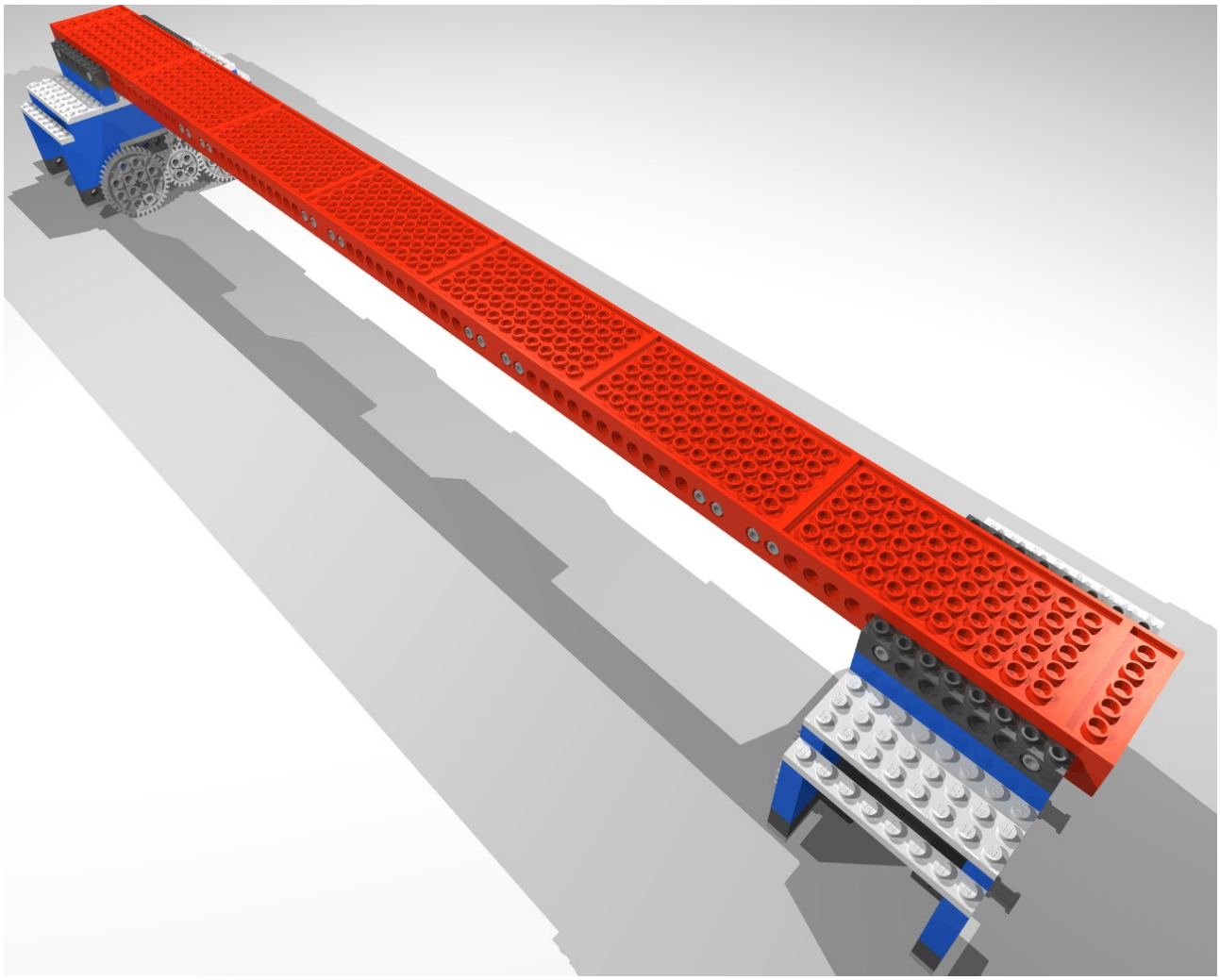
8



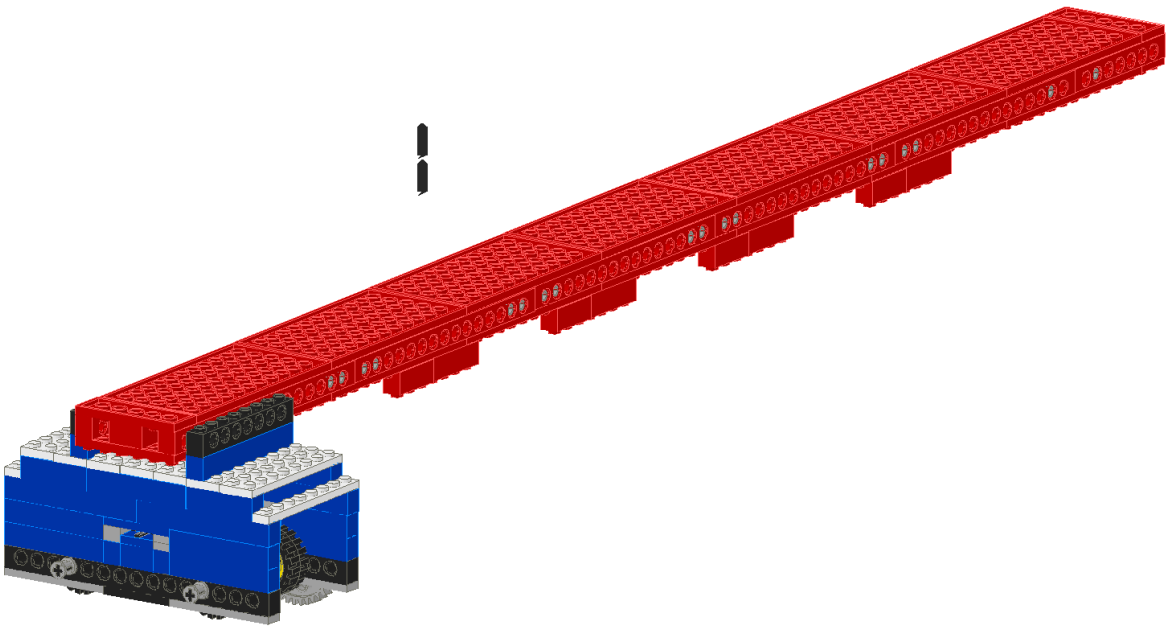
9



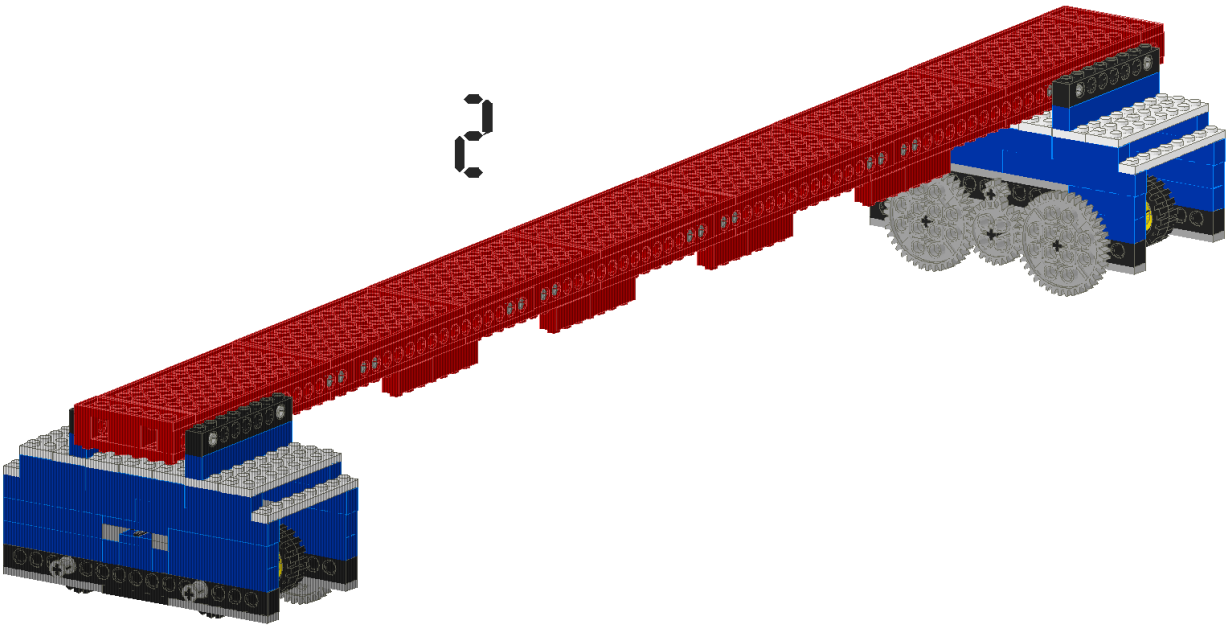
1x Bro med vagnar



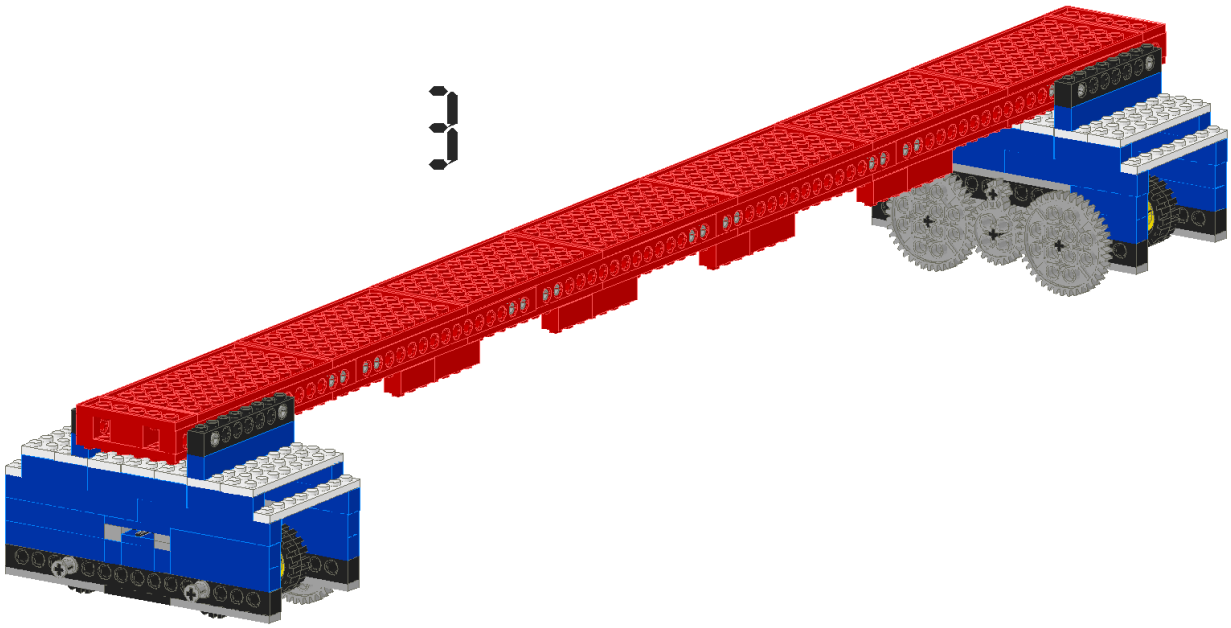
1



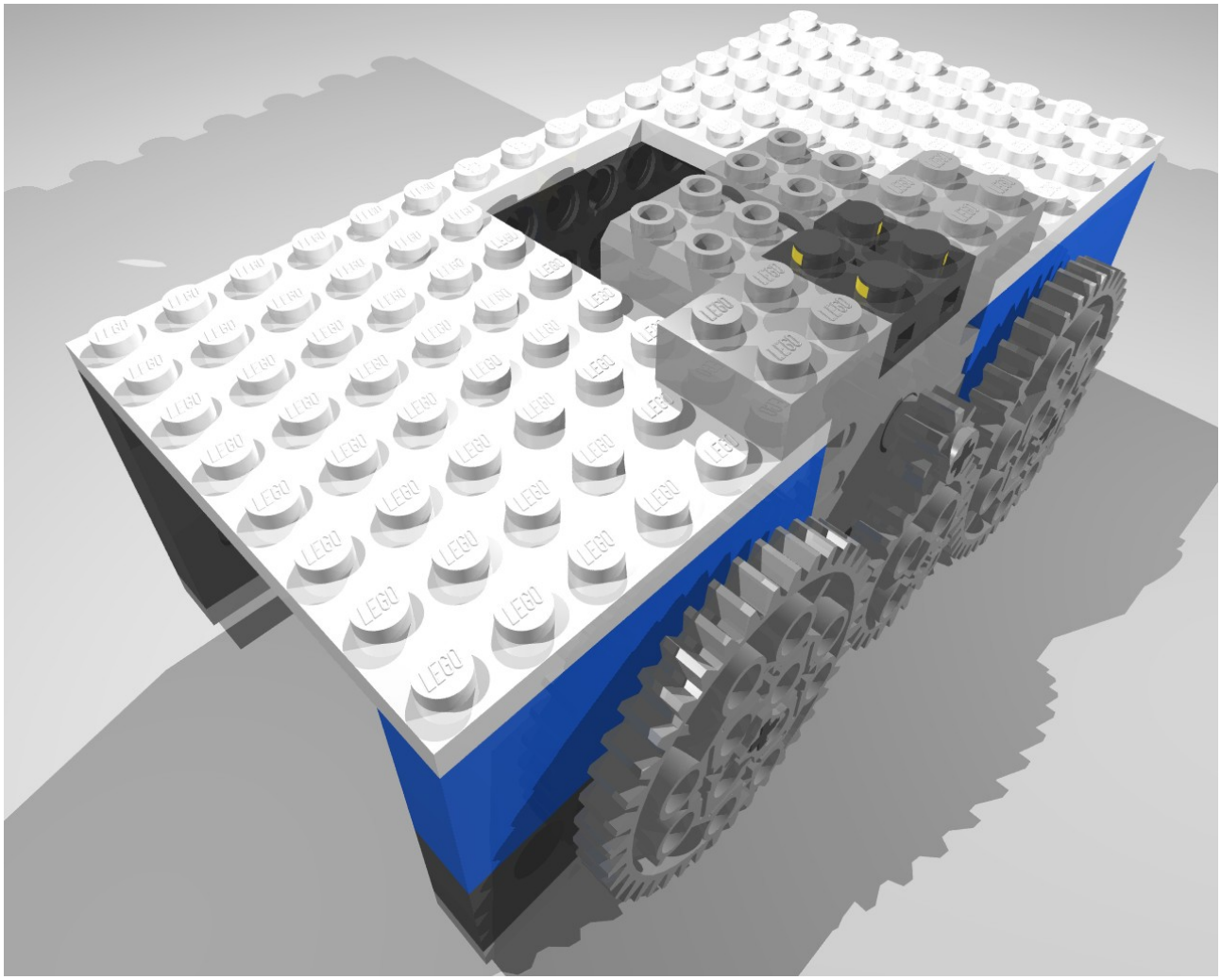
2



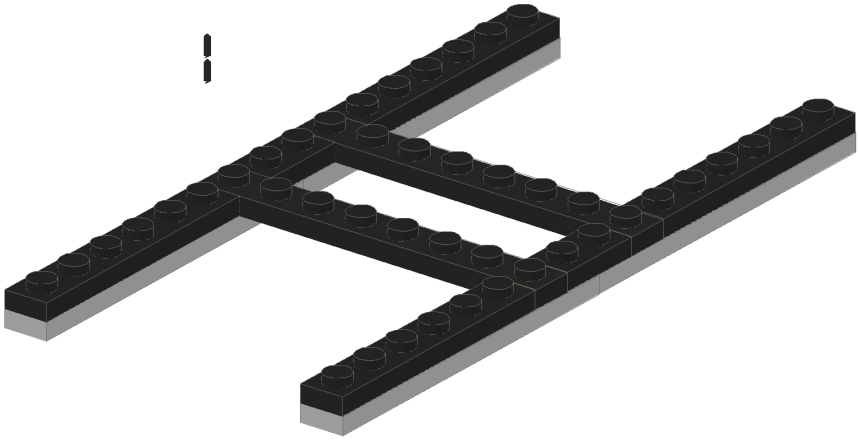
3



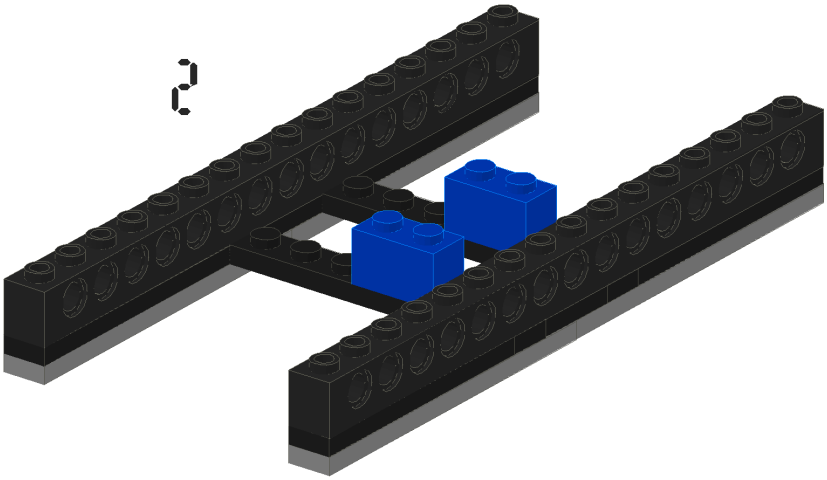
1x Vagn X



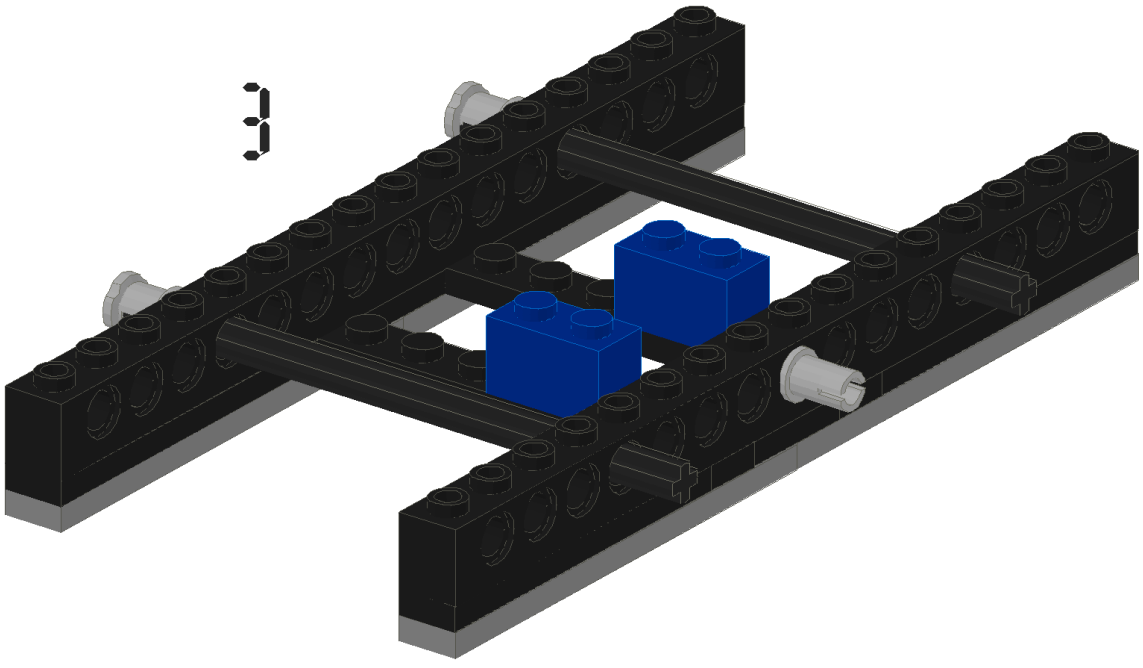
1



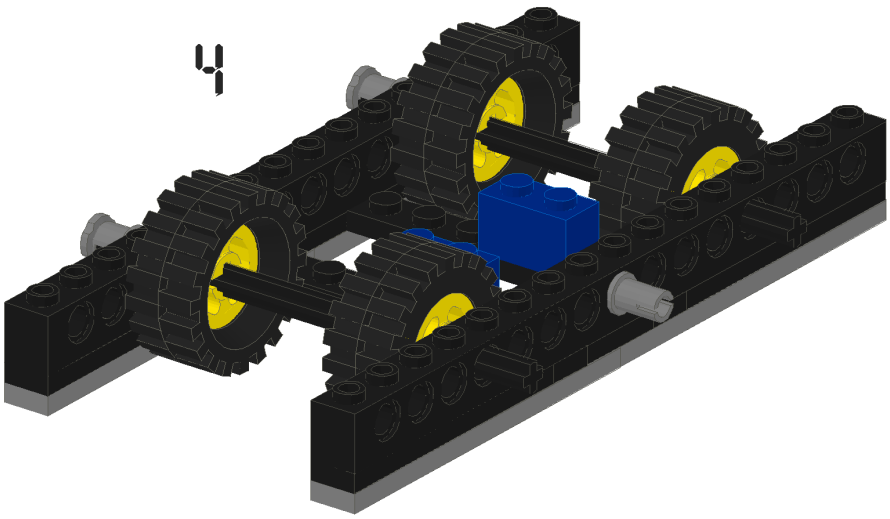
2



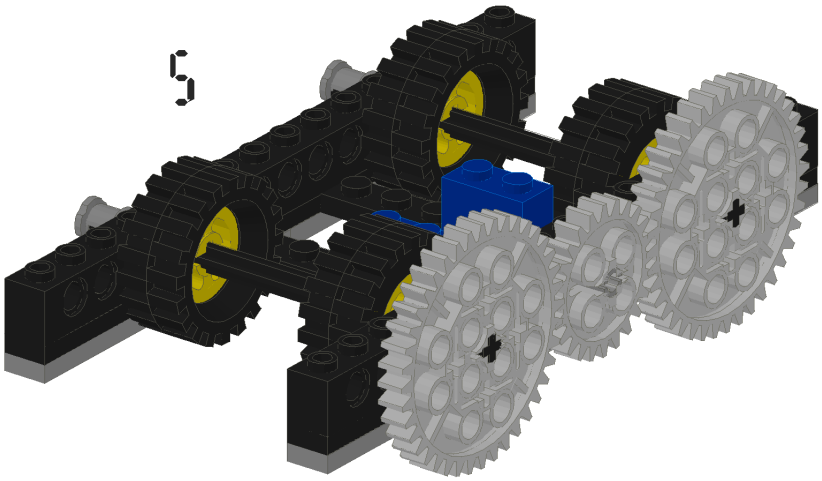
3



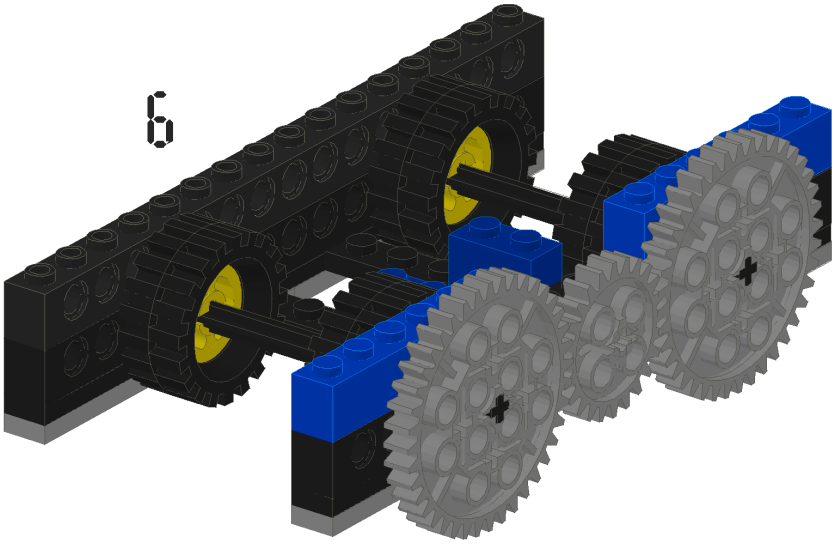
4



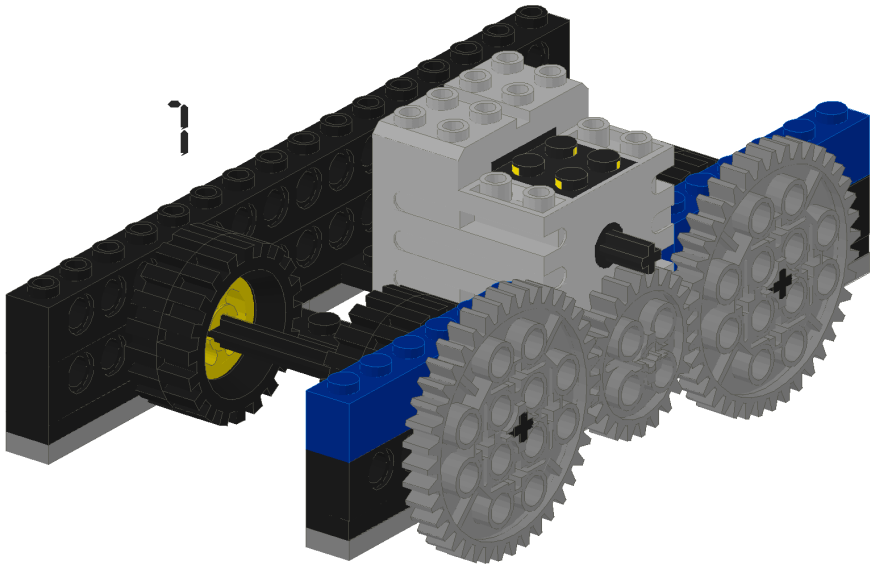
5



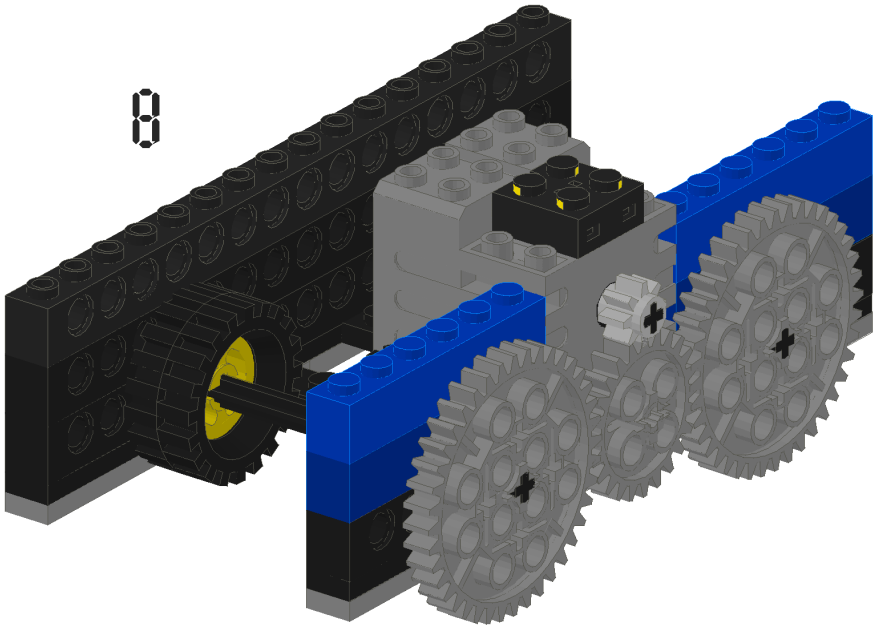
6



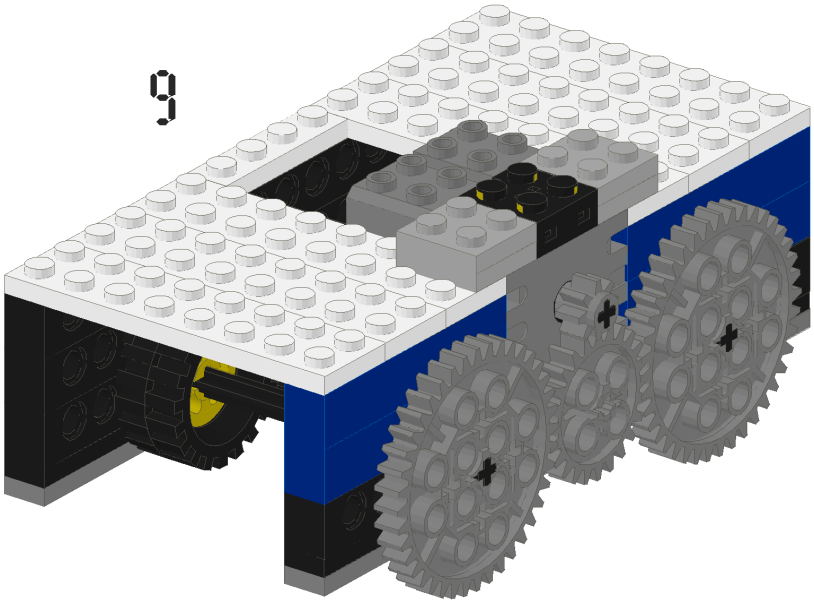
7



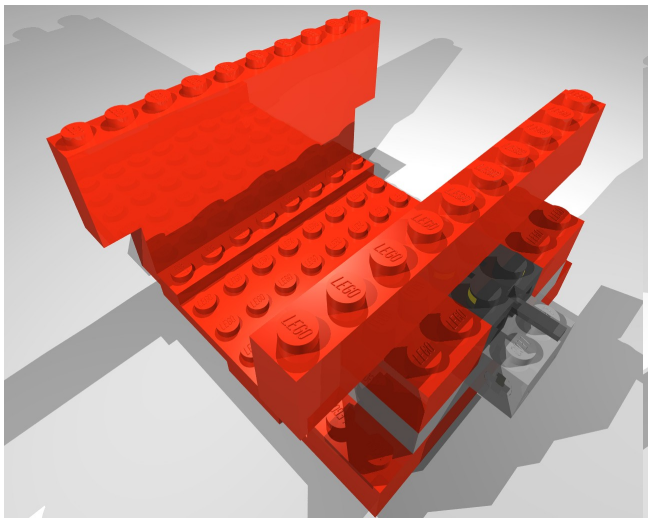
8



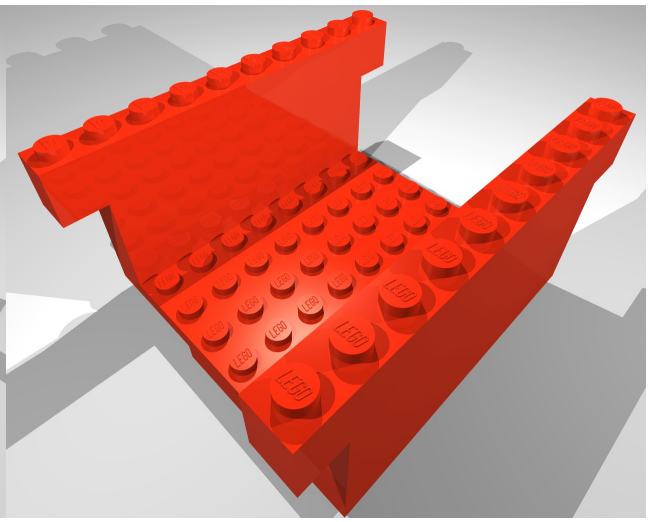
9



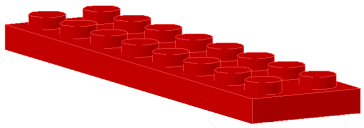
1x Vagn X under



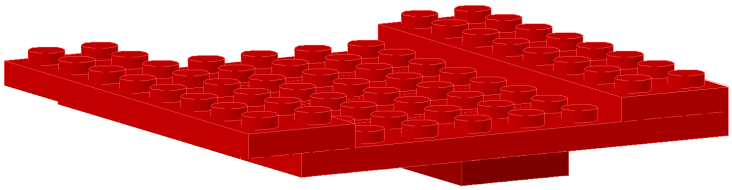
1x Vagn Y under



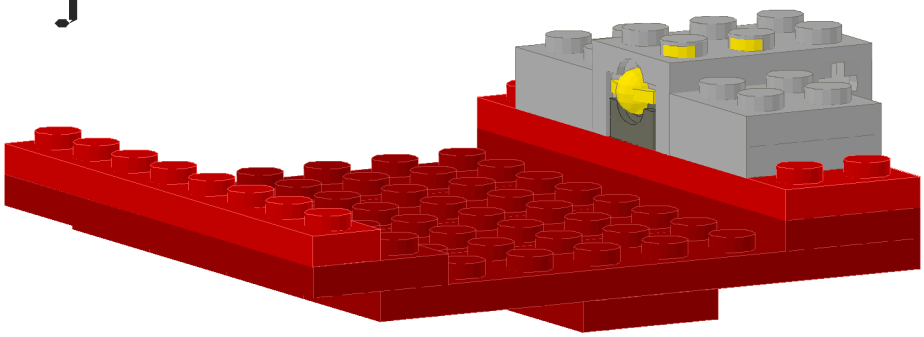
1



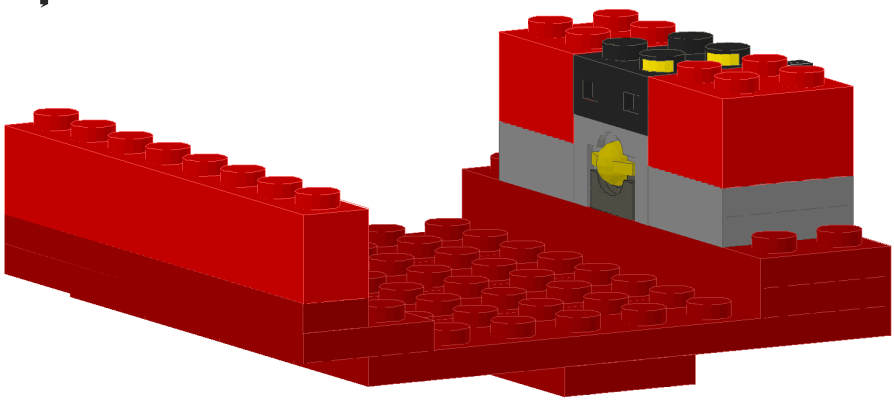
2



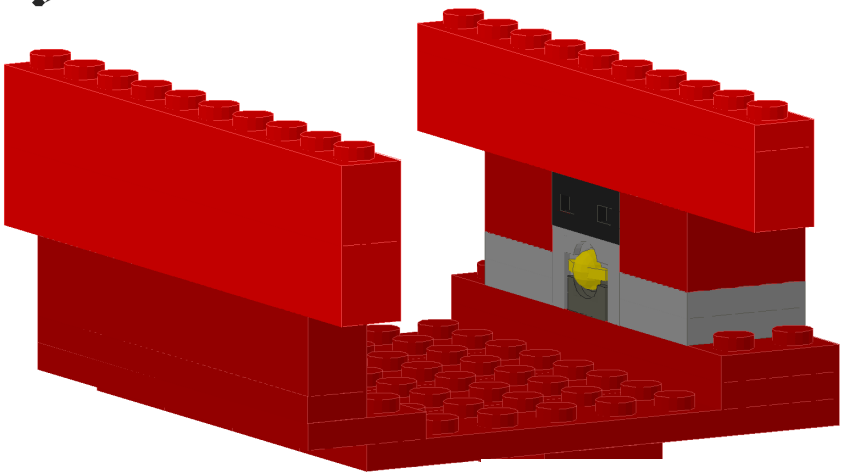
3



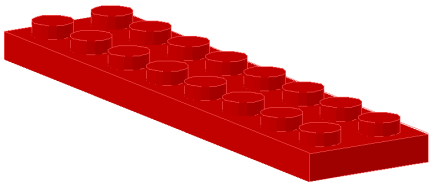
4



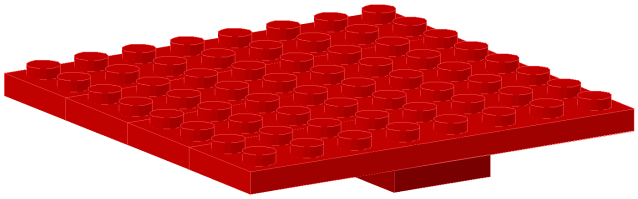
5



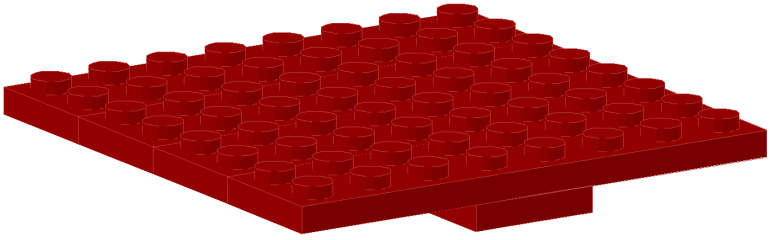
1



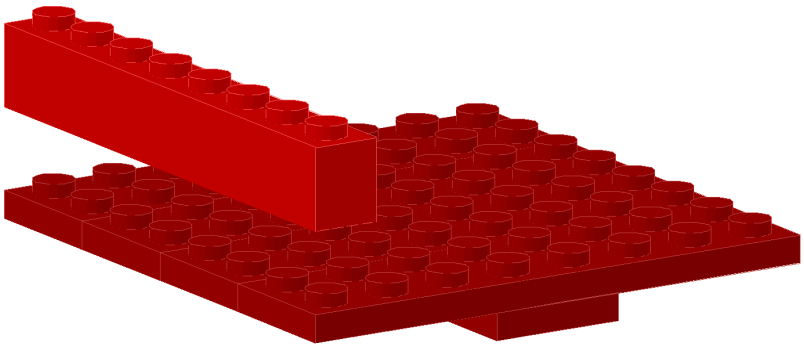
2



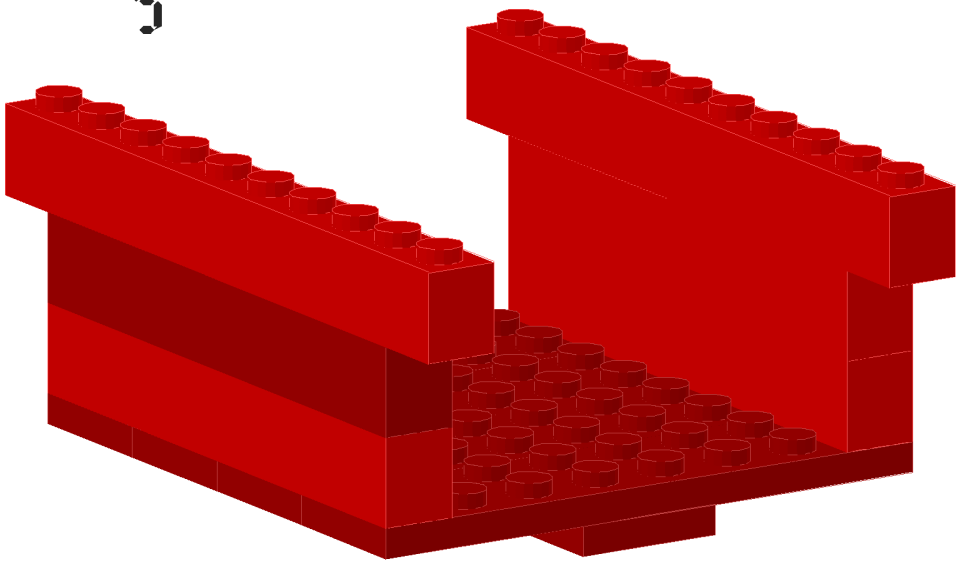
3



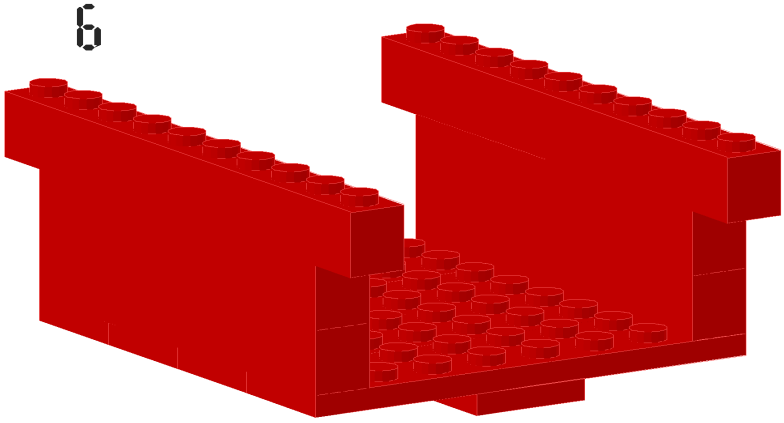
4



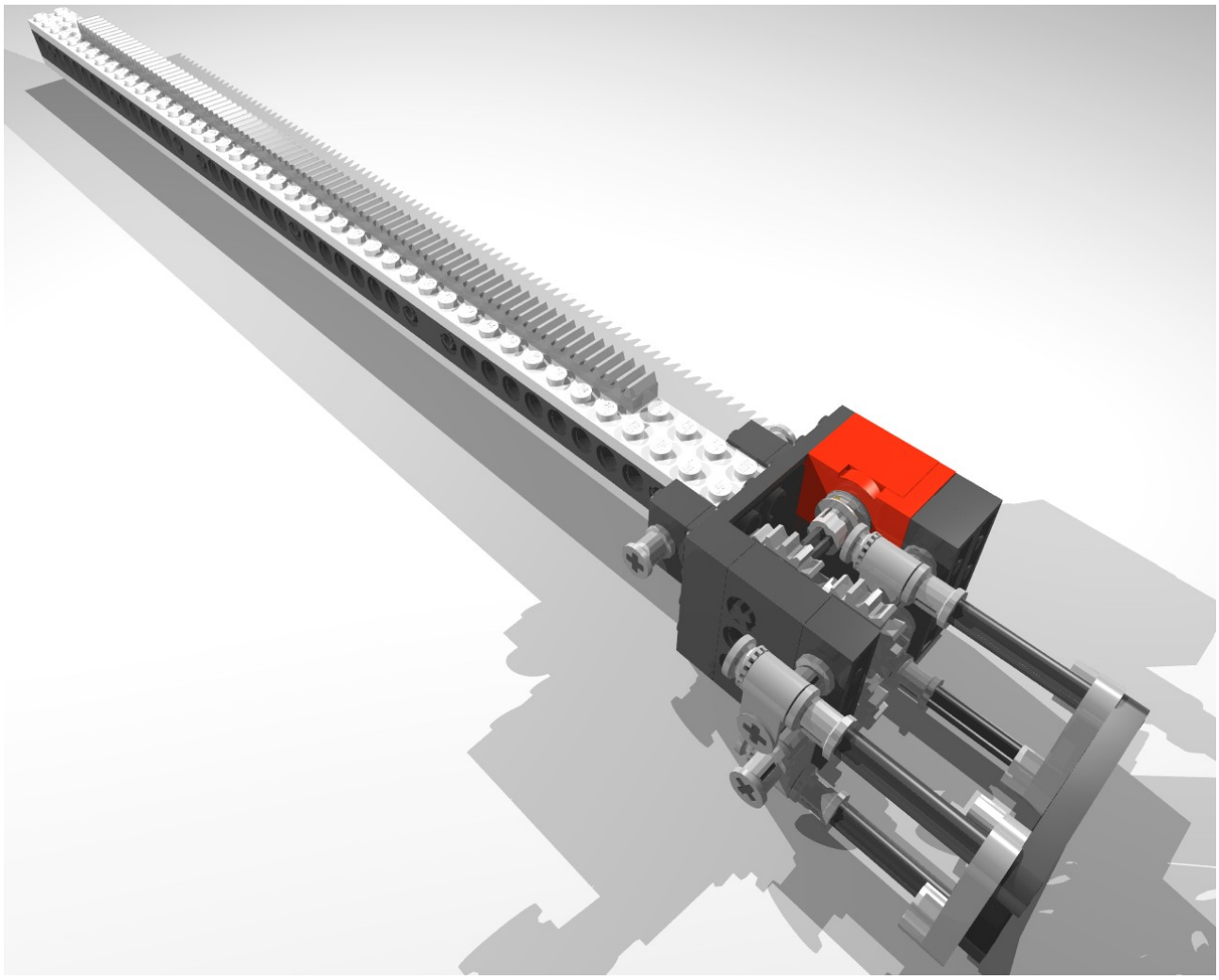
5



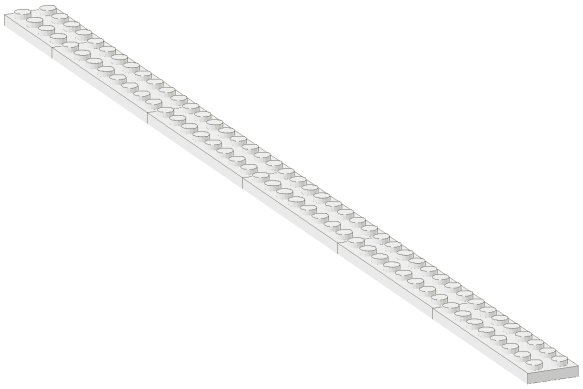
6



1x Kran



1

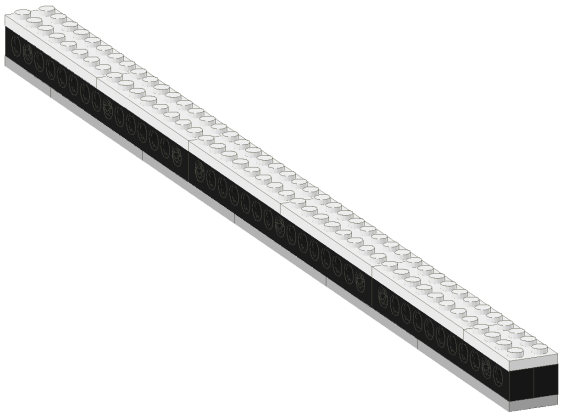


2

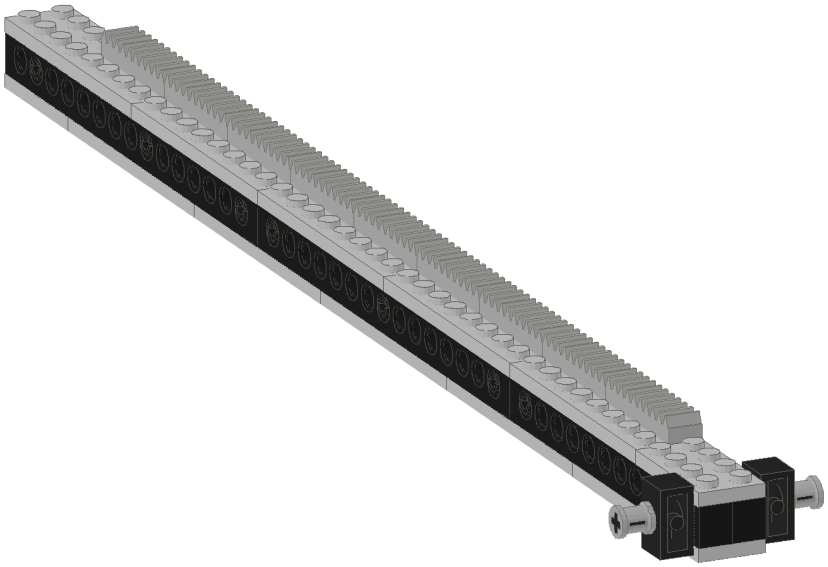
3

4

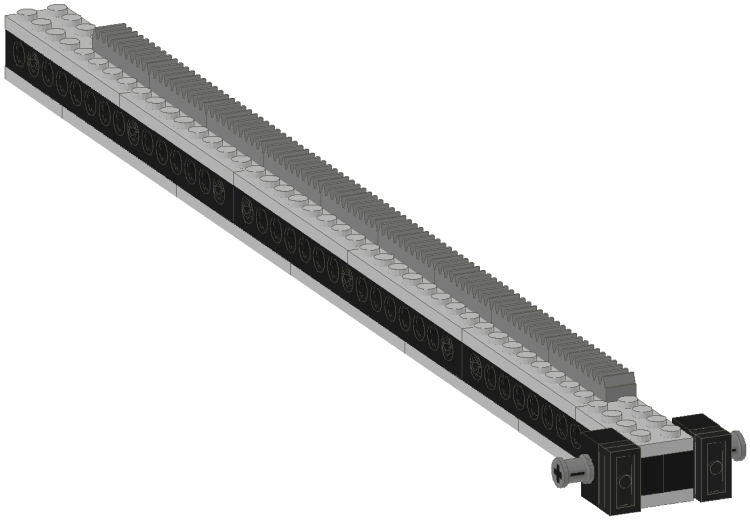
5



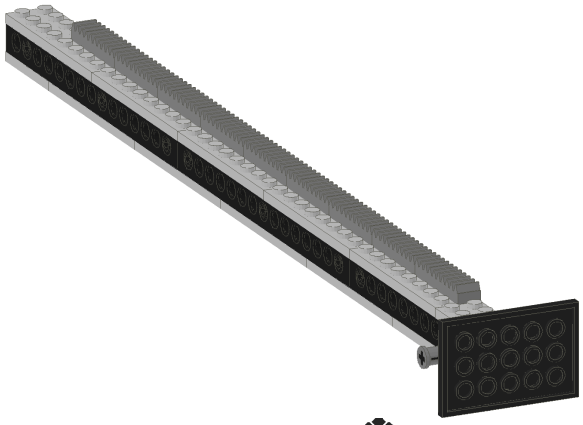
6



7

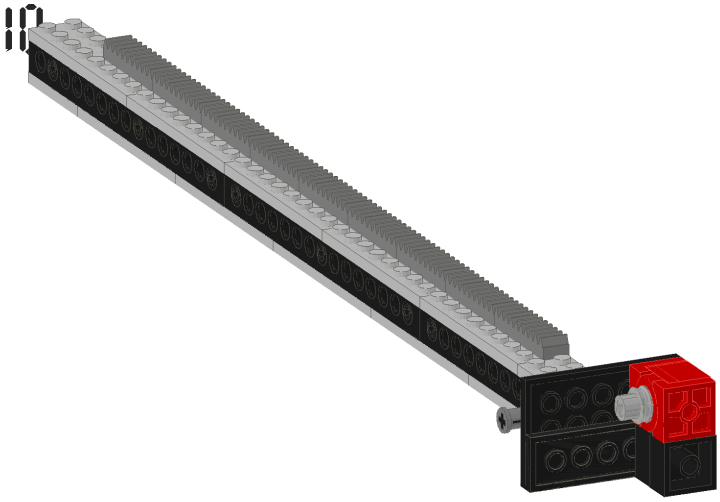


8

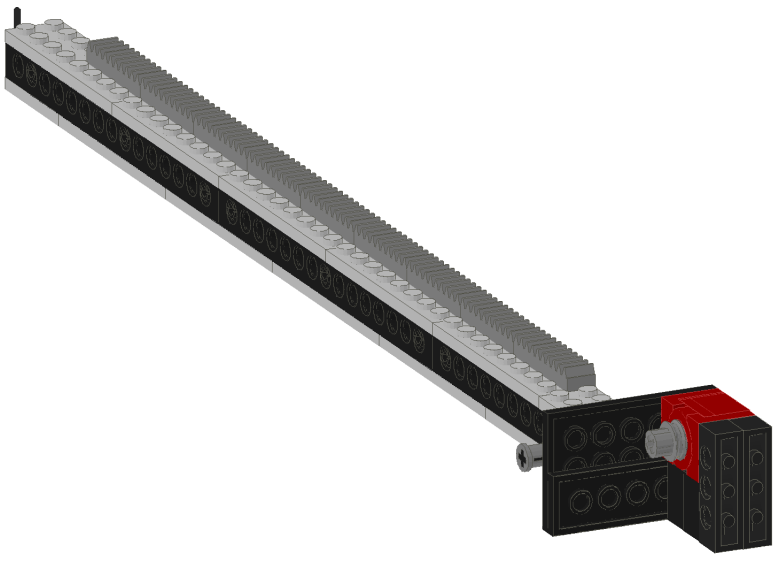


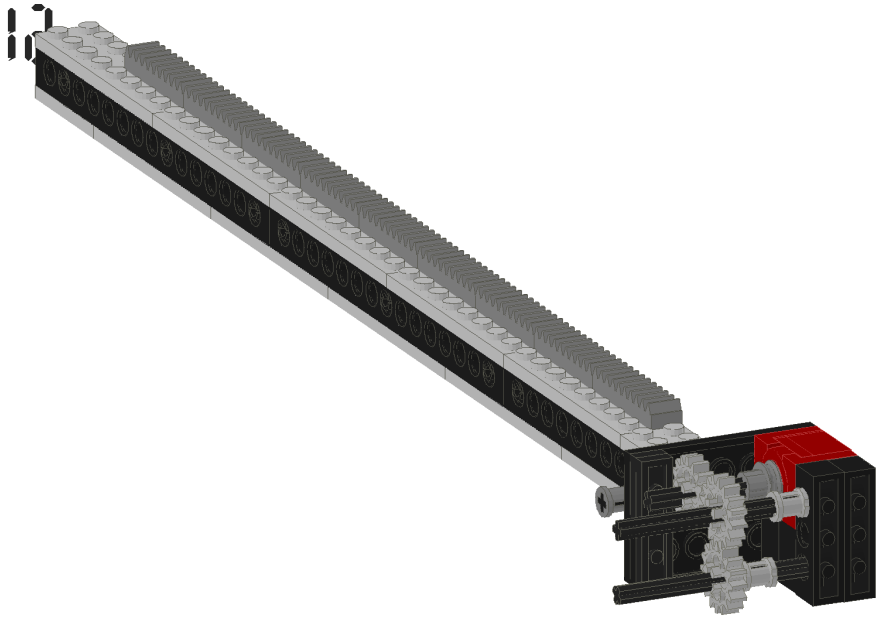
9

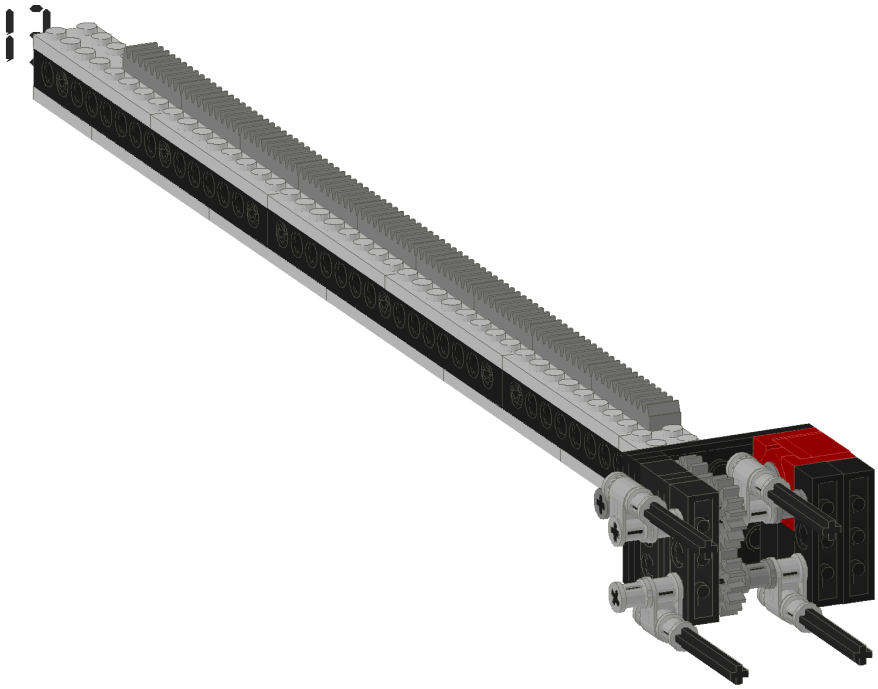
10

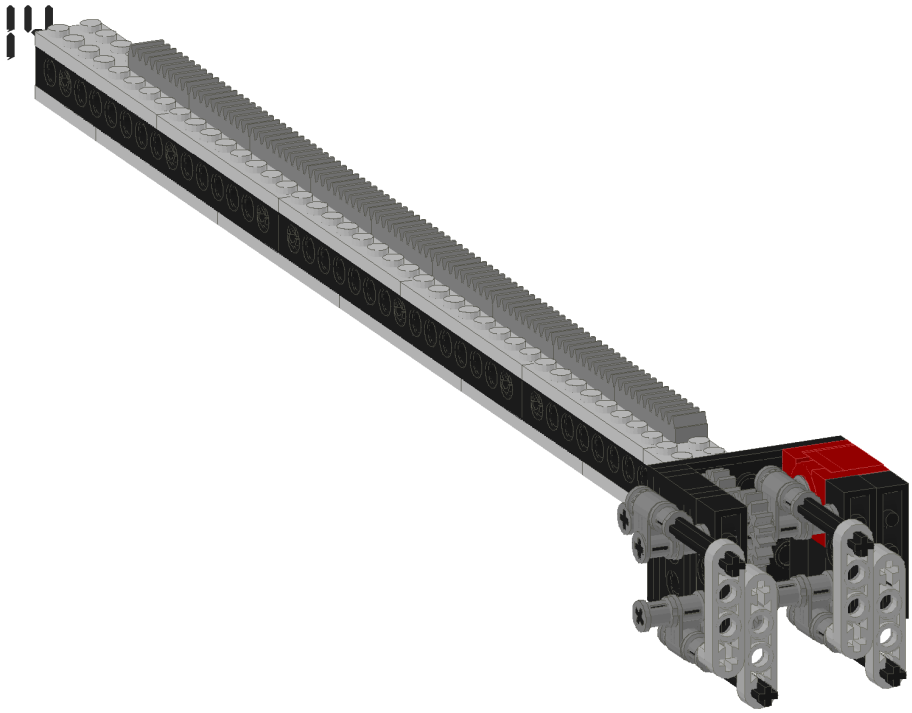


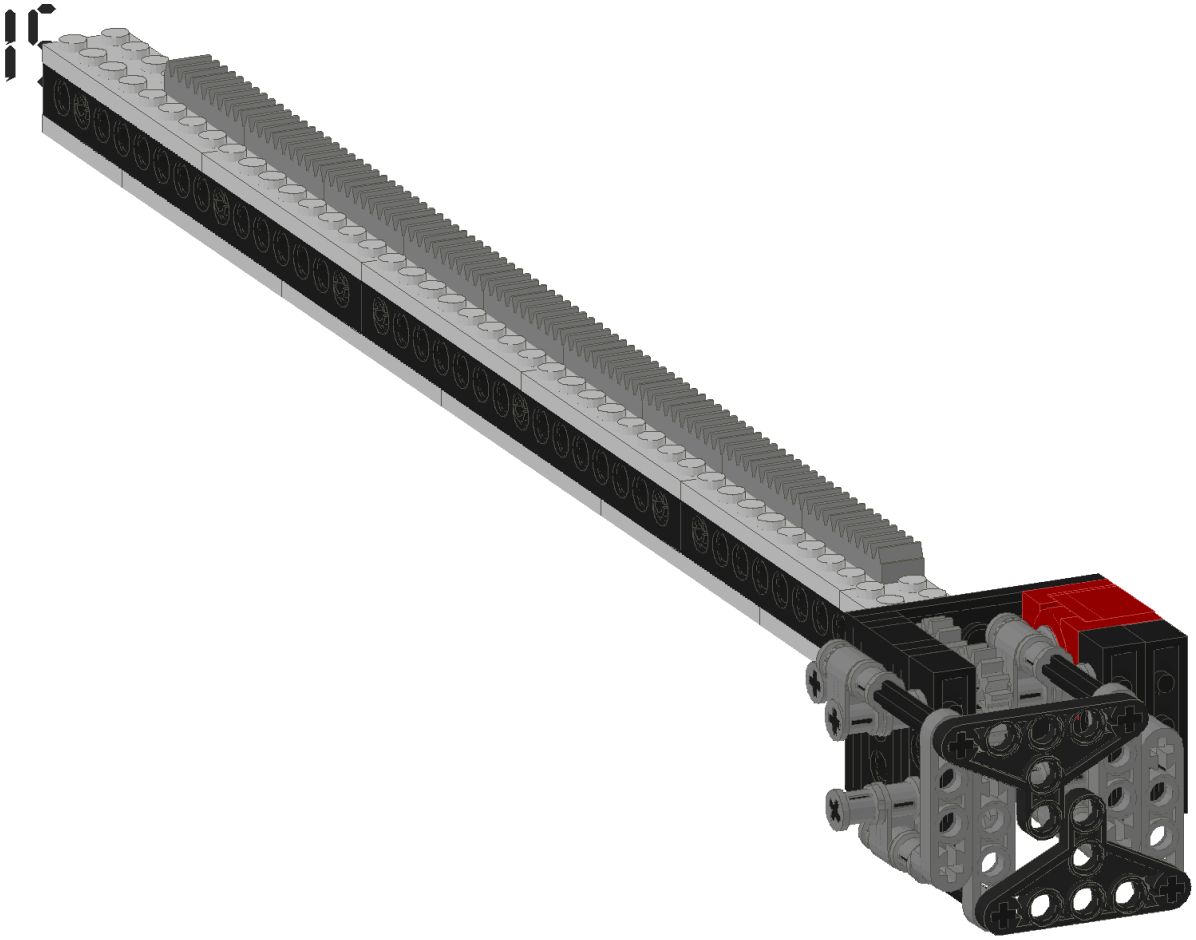
11

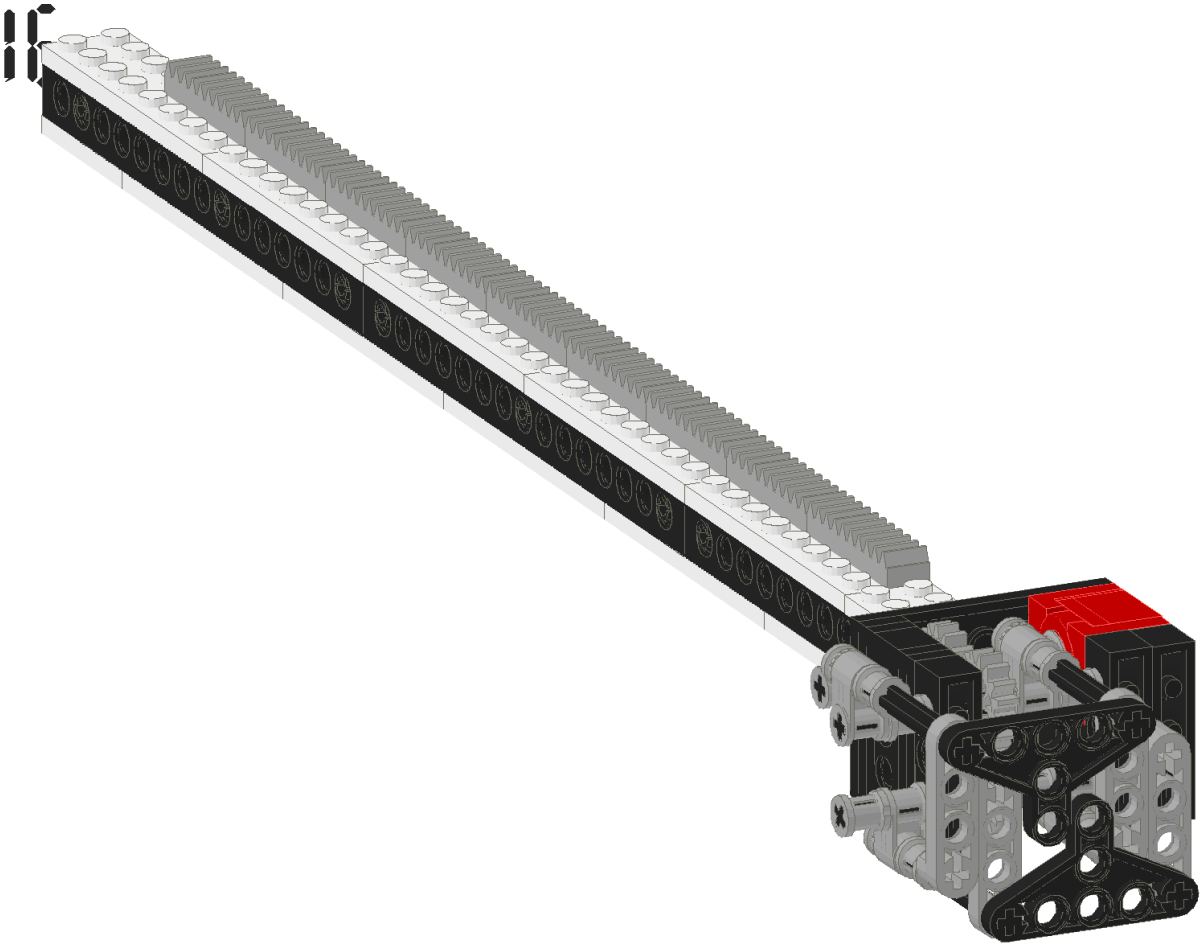




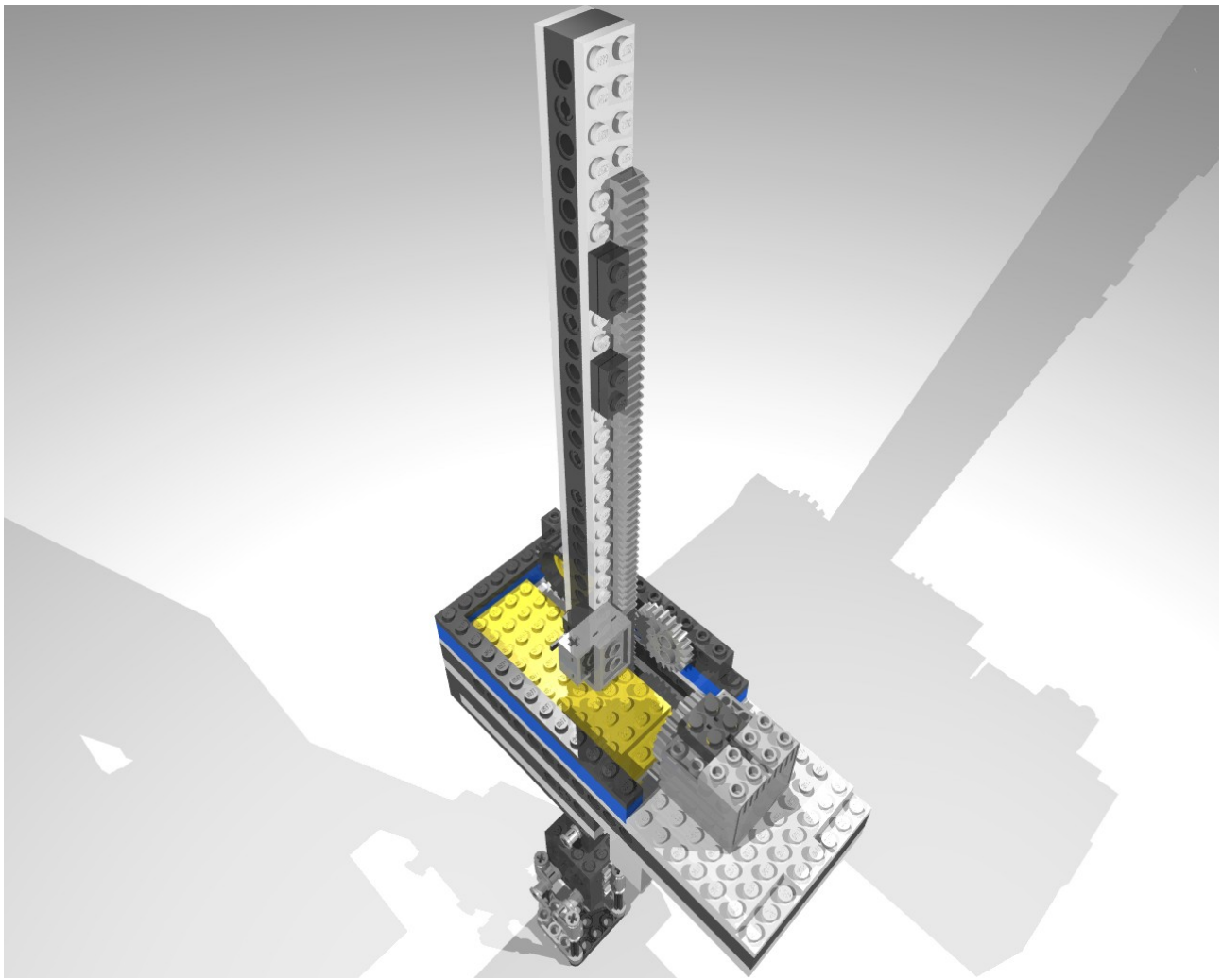




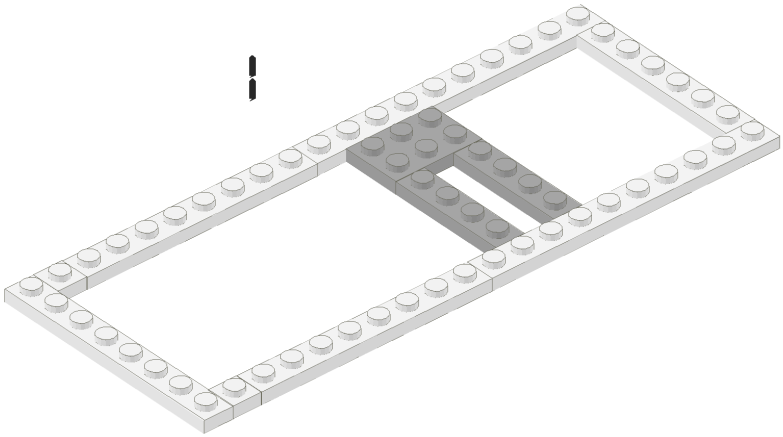




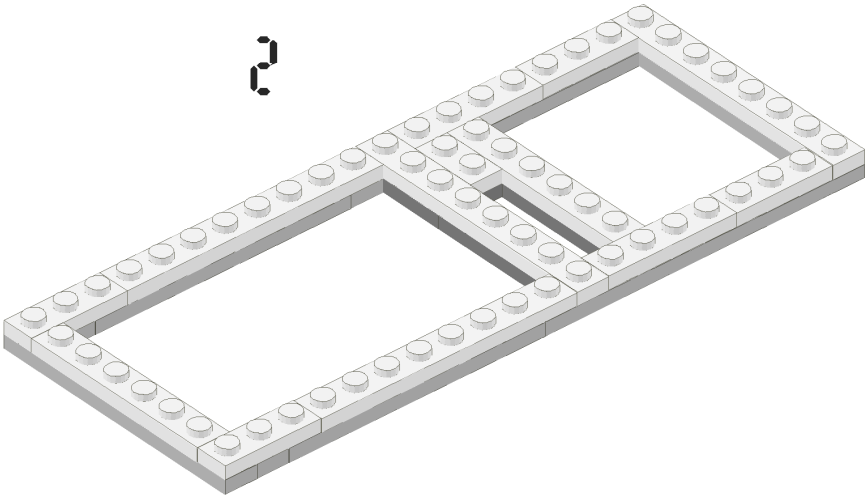
1x Kranhuvud



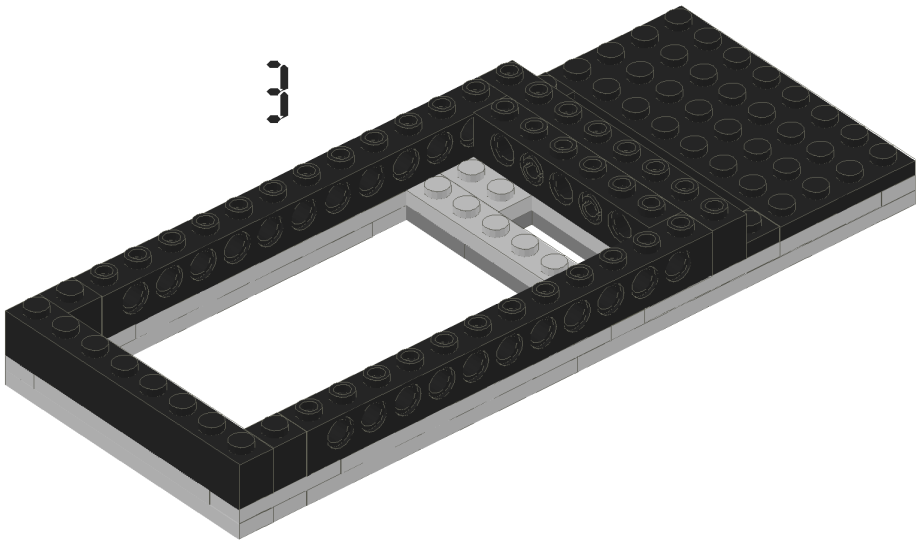
1



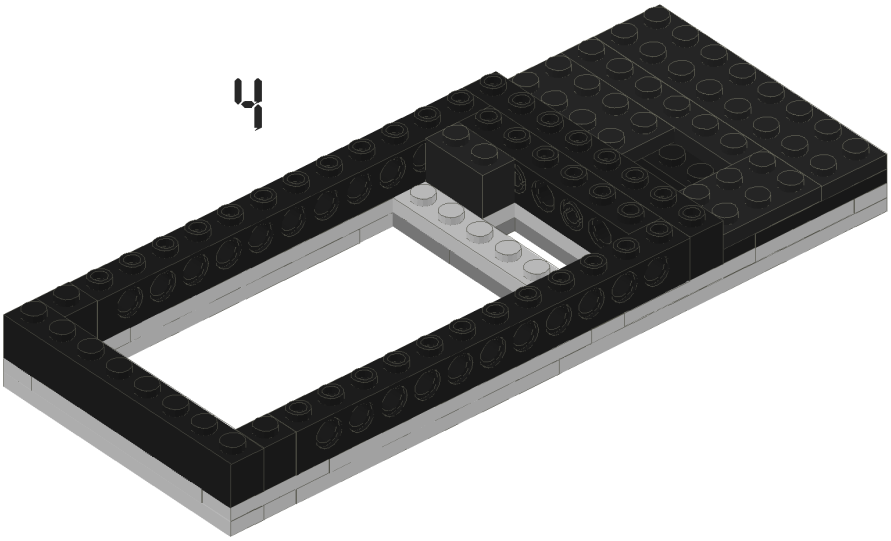
2



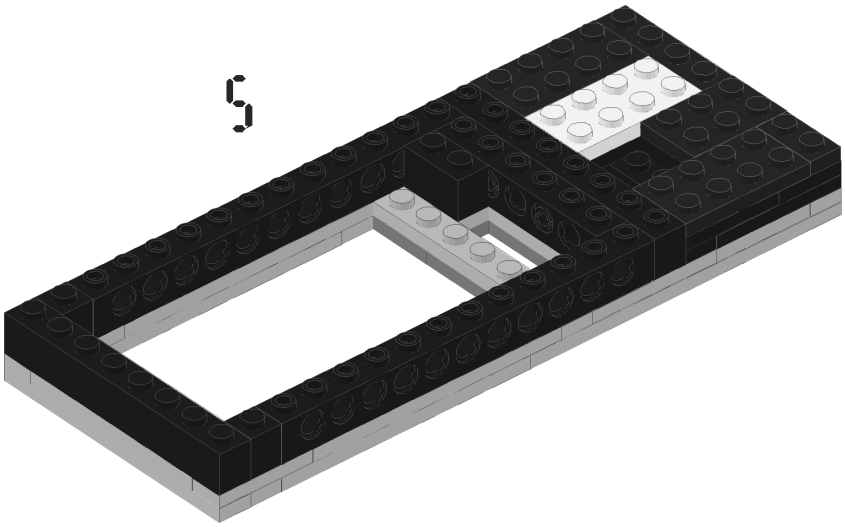
3



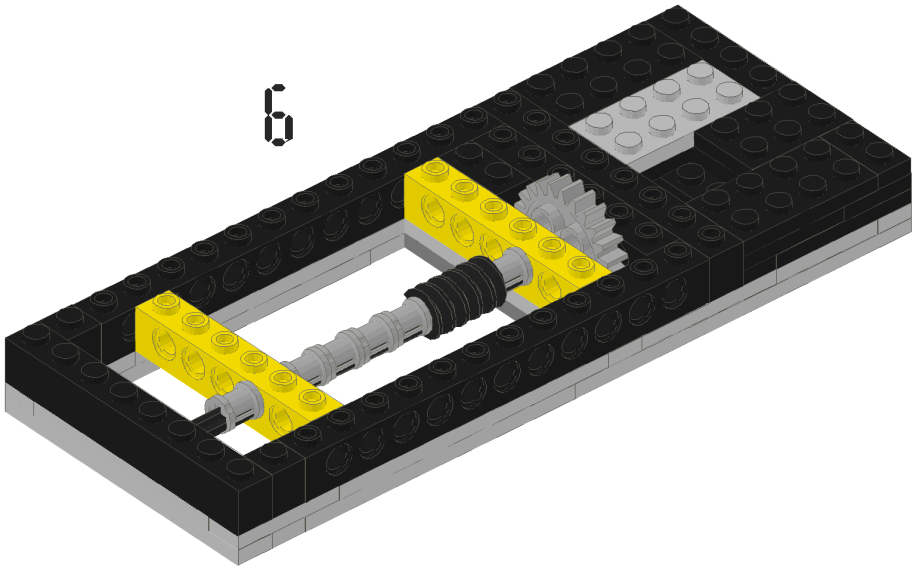
4



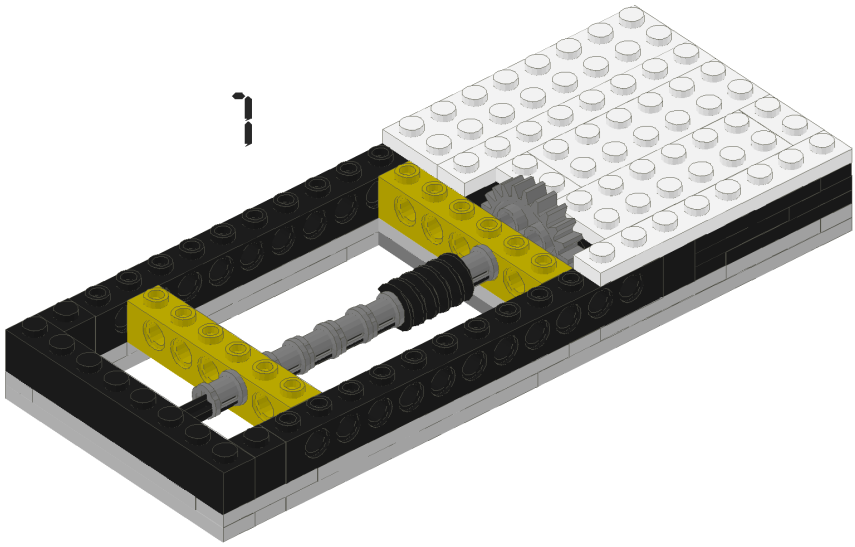
5



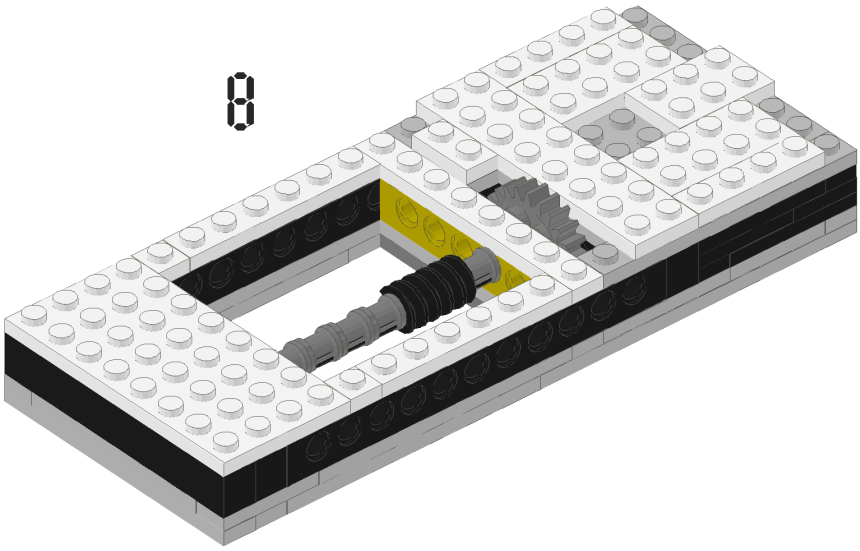
6



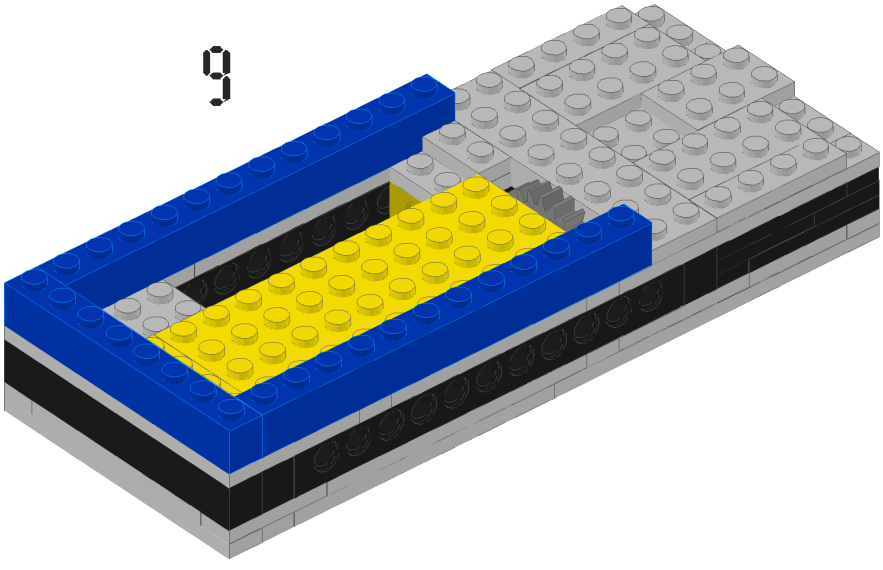
1



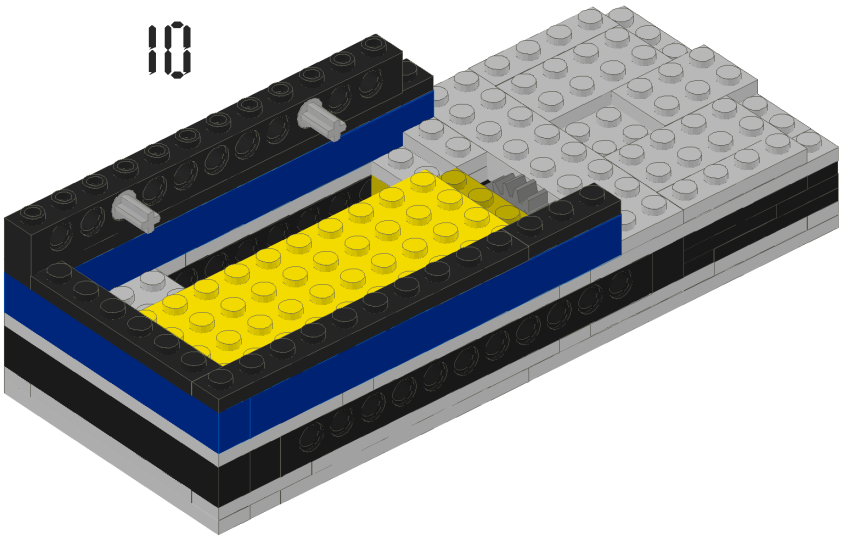
8



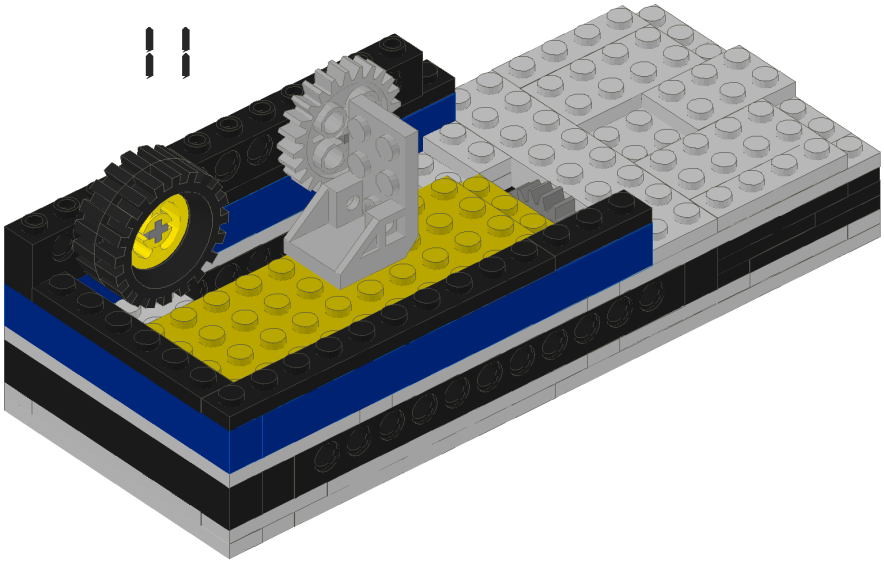
9



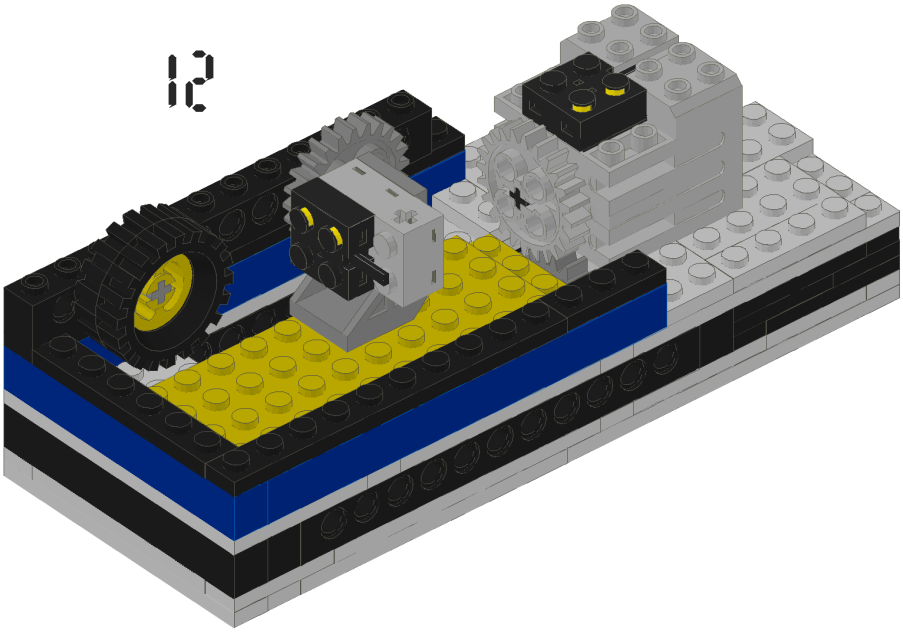
10



11



12



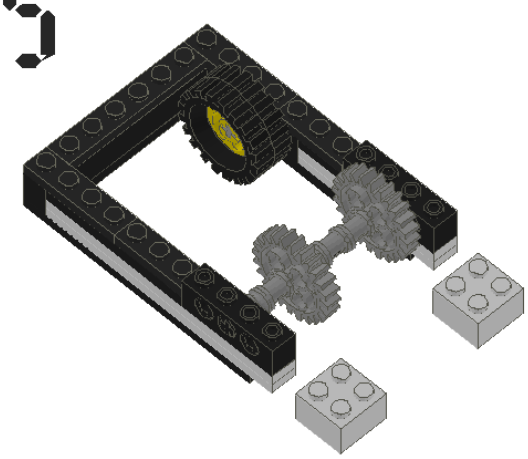
1

2

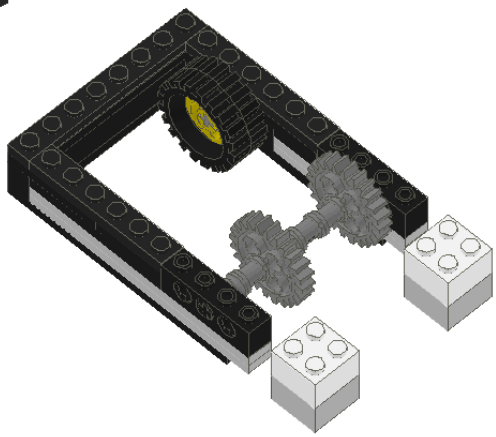
3

4

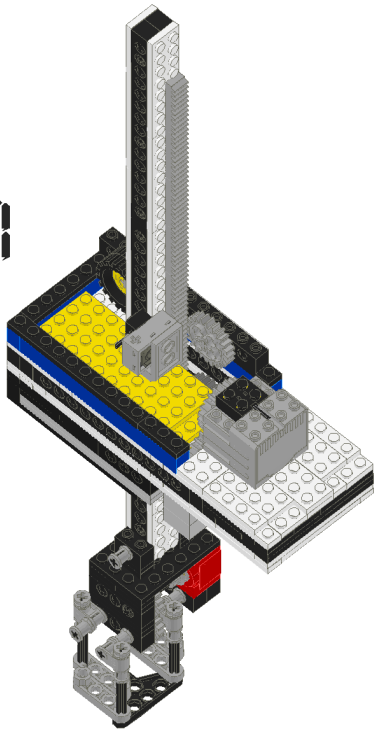
5



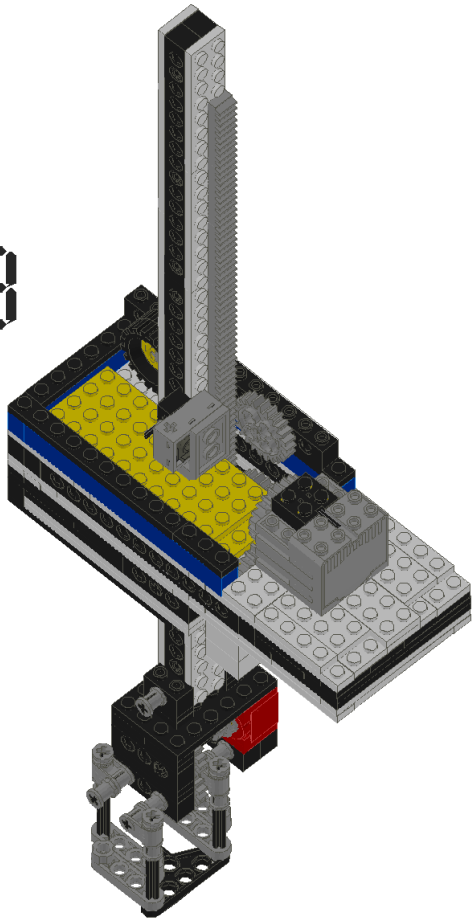
6



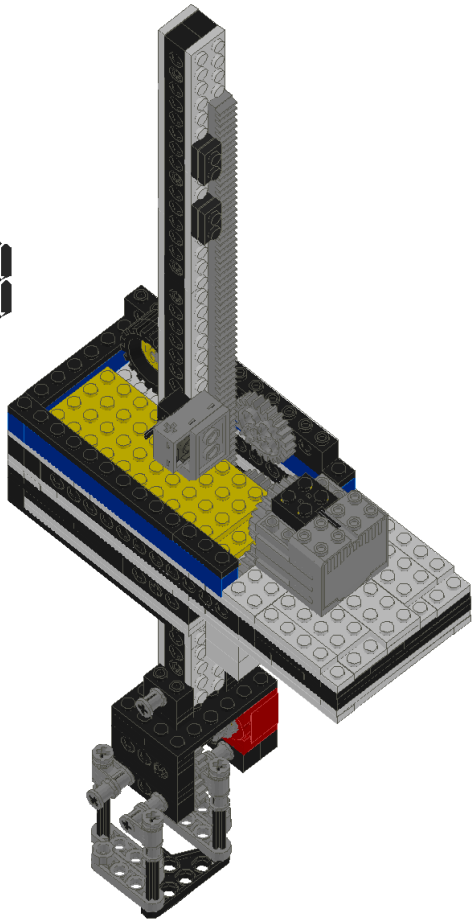
7



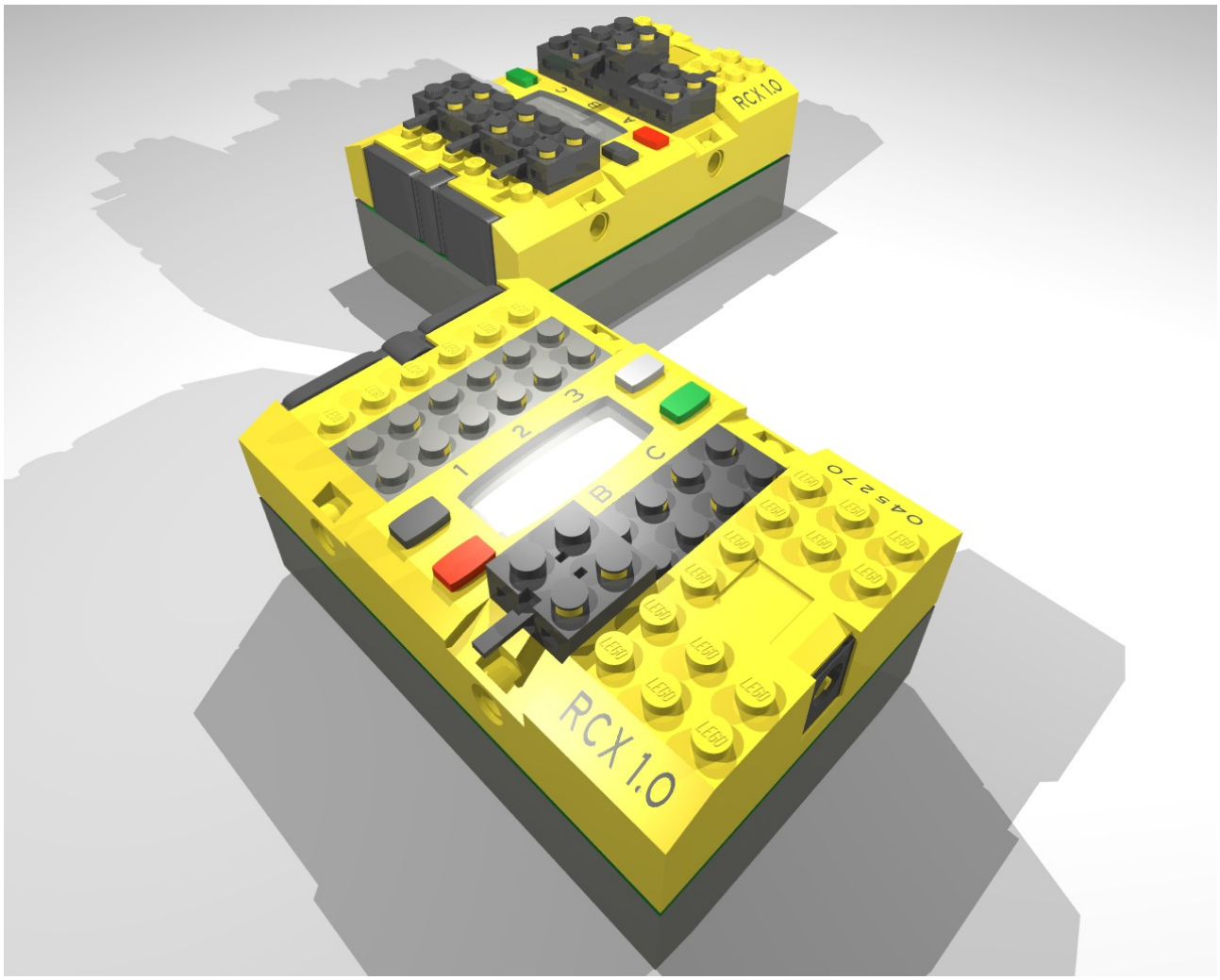
8



9



RCX-kopplingar



Observera placering. RCX1 ska både ha kontakt med RCX2 och datorsändaren.

RCX 1		**RCX 2**	
Sensor 1	1x Trycksensor vagn X	Sensor 1	1x Minimotor klo
Sensor 2	1x Trycksensor vagn Y	Sensor 2	
Sensor 3	1x Trycksensor kran Z	Sensor 3	
Motor A	1x Motor vagn X	Motor A	
Motor B	2x Motor vagn Y	Motor B	
Motor C	1x Motor kran Z	Motor C	

Modifiering av källkoden till Woodin

'Till ett program, BlitzIn, som används för att spela schack på Internet finns ett insticksprogram kallat Woodin. Woodin är skapat av DGT Projects som gjort sensorbrädet. Insticksprogrammet kommunicerar med BlitzIn och skickar dragen man utför på brädet. Motståndarens drag läses upp i högtalarna av Woodin. Genom att källkoden till woodin finns tillgänglig på Internet kan vi modifiera den så att där den läser upp dragen skickar den meddelande till roboten om vilket drag som ska utföras. Här följder den del av koden som ändrats.

Lägg märket till `legoOut.Format("nqc.exe -msg %d -msg %d -msg %d -msg 151",squereFrom, squereTo+64,pjasen);` här anropas nqc.exe som kommunicerar med RCX:en. Fyra meddelande skickas; från-ruta, till-ruta, vilken pjäs som ska flyttas samt utför-draget-kommandot.

```
DLLFUNCTION int __stdcall AnnounceMove(char *move, char piececode, unsigned int
special, bool status)
{
    char movecopy[6];
    int j;
    char f;
    int squereTo=-1;
    int squereFrom=-1;
    int ejDrag=0;


    if ((f>='a')&&(f<='h'))
    {
        if (j==0) squereFrom = (f-'a');
        if (j==2) squereTo = (f-'a');

        if (j==0 && long_notation)
            soundlist[soundlistindex]=42+(f-'a');
        if (j==2) soundlist[soundlistindex]=42+(f-'a');
    }

    if ((f>='1')&&(f<='8'))
    {
        if (j==1) squereFrom +=(7-(f-'1'))*8;
        if (j==3) squereTo +=(7-(f-'1'))*8;

        if (j==1 && long_notation) soundlist[soundlistindex++]+=(7-(f-'1'))*8;
        if (j==3) soundlist[soundlistindex++]+=(7-(f-'1'))*8;
    }

    if (j==1)
    {
        if (special & TAKES) soundlist[soundlistindex++]=S_TAKES;
    }

    if (special & PROMOTION)
    {
        if (piececode==ROOK) soundlist[soundlistindex++]=S_ROOK;
        if (piececode==KNIGHT) soundlist[soundlistindex++]=S_KNIGHT;
        if (piececode==BISHOP) soundlist[soundlistindex++]=S_BISHOP;
        if (piececode==QUEEN) soundlist[soundlistindex++]=S_QUEEN;
    }

    if (special & ENPASSANT) soundlist[soundlistindex++]=S_ENPASSANT;
    if (special & MAT) soundlist[soundlistindex++]=S_MAT;
    if (special & PAT) soundlist[soundlistindex++]=S_PAT;
    soundlist[soundlistindex]=0;
```

```
    if(!ejDrag)
    {
        int pjasen=0;
        if(piececode==PAWN)
            pjasen=128;
        if(piececode==KNIGHT)
            pjasen=129;
        if(piececode==BISHOP)
            pjasen=130;
        if(piececode==ROOK)
            pjasen=131;
        if(piececode==QUEEN)
            pjasen=132;
        if(piececode==KING)
            pjasen=133;

        CString legoOut;

        legoOut.Format("nqc.exe -msg %d -msg %d -msg %d -msg 151",squereFrom,
squereTo+64,pjasen);
        WinExec(legoOut,SW_HIDE);
    }

    ProcessAnnounces(true);
    return 0;
}
```

Källkoden till RCX1

Koden är skriven i NQC, Not Quite C som är ett C-liknande språk för RCX:er. Både kompilator och det som behövs därtill är fria program.

Roboten kräver två RCX:er, den första har hand om i princip hela programmet utom öppnande och stängande av klon som ligger på RCX2. Då RCX1 vill öppna eller stänga klon skickas ett meddelande till RCX2. Ett meddelande består av en byta, dvs ett tal mellan 0 och 255. I #define-listan hittar du vad varje tal motsvarar. Överst deklareras hur motorerna och sensorerna ska kopplas.

Koden är strukturerad så att det ska gå lätt att följa med vad som händer. Huvudslingan i main() snurrar hela tiden tills man stänger av programmet. Den väntar på ett meddelande och utför det som ska göras. Vanligtvis går programmet så här:

1. Väntar på meddelande
2. Får ett meddelande från 0 till 63. Sätter variabeln för från-rutan till meddelandets värde.
3. Får ett meddelande från 64 till 127. Sätter variabeln för till-rutan till meddelandets värde – 64.
4. Får ett meddelande från 128 till 148. Sätter variabeln frö vilken pjäs som ska flyttas.
5. Får meddelande 151. Börjar utföra draget. Se doMove();
6. Flyttar x-leden till från-rutans linje. Genom att räkna antal förändringar då x-sensorn går längs med kanten hittar den till rätt linje oavsett om vart den går ifrån. Se moveXTo();
7. Samma princip för y-leden.
8. Ta upp en pjäs. Se pickUp(); Flyttar z-leden till rätt höjd, stänger klon och tar upp z-leden.
9. Flyttar x- och y-lederna till till-rutan, kör putDown(); som sänker z-leden till rätt höjd, öppnar klon och tar upp z-leden igen.
10. doMove(); klar. Flyttar tillbaks x- och y-lederna till hörnet.
11. Går till 1.

```
#define SENSOR_X    SENSOR_1
#define SENSOR_Y    SENSOR_2
#define SENSOR_Z    SENSOR_3
#define MOTOR_X     OUT_A
#define MOTOR_Y     OUT_B
#define MOTOR_Z     OUT_C

#define PAWN_Z      4
#define ROOK_Z      3
#define BISHOP_Z    3
#define KNIGHT_Z    3
#define QUEEN_Z     2
#define KING_Z      1

#define REVERSE 0
#define FORWARD 1
#define __NOTETIME   10
#define __WAITTIME   12

#define LIGHT 43
#define DARK 41

// Messages & Piece/special move information
// MOVES             0
// ...             ...
// MOVES           127

#define PAWN       128
#define KNIGHT     129
#define BISHOP     130
#define ROOK       131
#define QUEEN      132
#define KING       133

#define DO_MOVE    151
#define PLAY_TUNE  152
//#define PLAY_SOUND 153
// RCX2            154
// RCX2            155
#define SEND_OPEN  156
#define SEND_CLOSE 157

// other defines
#define DROP_SQUARE_X -1;
#define DROP_SQUARE_Y -1;
#define HOME_SQUARE 0;

// Globals
int fromSquareX=0;
int fromSquareY=0;
int toSquareX=0;
int toSquareY=0;

int thePiece=0;

int vehicleX=0;
int vehicleY=0;
int xWay=REVERSE;
int yWay=REVERSE;
int zWay=REVERSE;
int pointZ=0;

// Main task
task main()
```

```
{
  int message;
  SetSensor(SENSOR_X, SENSOR_TOUCH);
  SetSensor(SENSOR_Y, SENSOR_TOUCH);
  SetSensor(SENSOR_Z, SENSOR_TOUCH);
  while(true)
  {
    // Wait for message
    ClearMessage();
    while(Message()==0);
    PlayTone(523,4*__NOTETIME);
    message=Message();

    // processMessage
    // Message 0-127 Move-message
    if(message<128)
    {
      calculateMove(message);
    }

    // Message 128-148 Piece
    if(message>PAWN-1 && message < KING+1)
    {
      thePiece = message;
    }

    // Message 151-255 other messages
    if(message > 150)
    {
      switch(message)
      {
        case DO_MOVE:
          doMove();
        moveToSquare(0);
        break;

        case PLAY_TUNE:
        break;

        case SEND_OPEN:
                SendMessage(154);
        break;

        case SEND_CLOSE:
                SendMessage(155);
        break;

        case 200:
               openClaw();
        break;
        case 201:
               closeClaw();
        break;
      }
    }
  }
}

// Move to quare
void moveToSquare(int nr)
{
  int y=nr/8;
  int x=nr-y*8;
  moveYTo(y);
```

```
  moveXTo(x);
}


void openClaw()
{
  SendMessage(SEND_OPEN);
}


void closeClaw()
{
  SendMessage(SEND_CLOSE);
}

// calculateMove gets a coded move by a number 0-127.
void calculateMove(int codedMove)
{
  int toMove=0;
  int x; int y;
  //check if from or to
  if(codedMove > 64)
  {
    codedMove-=64;
    toMove=1;
  }

  y=codedMove/8;
  x=codedMove-(y*8);

  if(toMove==1)
  {
    toSquareX = x;
    toSquareY = y;
  }
  else
  {
    fromSquareX = x;
    fromSquareY = y;
  }

  return;
}

// do the move on the board,
// not o-o, takes etc. Just moves a
// piece from a square to another
void doMove()
{
  // Go to fromSquare
  moveXTo(fromSquareX);
  moveYTo(fromSquareY);

  // Pick up the piece
  pickUp();

  // Go to toSquare
  moveXTo(toSquareX);
  moveYTo(toSquareY);

  // put down piece
  putDown();
  return;
}
```

```
// Pick up a piece
void pickUp()
{
  Wait(4*__WAITTIME);
  PlayTone(262,1*__NOTETIME);
  PlayTone(294,1*__NOTETIME);
  PlayTone(330,1*__NOTETIME);
  Wait(4*__WAITTIME);

  moveZToPiece(thePiece);
  //close
  closeClaw();
  moveZTo(0);
}
// Put down a piece
void putDown()
{
  Wait(4*__WAITTIME);
  PlayTone(330,1*__NOTETIME);
  PlayTone(294,1*__NOTETIME);
  PlayTone(262,1*__NOTETIME);
  Wait(4*__WAITTIME);

  moveZToPiece(thePiece);
  //open
  openClaw();
  moveZTo(0);
}

void moveZToPiece(int piece)
{
  if(piece == KING)
    moveZTo(KING_Z);
  if(piece == QUEEN)
    moveZTo(QUEEN_Z);
  if(piece == ROOK)
    moveZTo(ROOK_Z);
  if(piece == BISHOP)
    moveZTo(BISHOP_Z);
  if(piece == KNIGHT)
    moveZTo(KNIGHT_Z);
  if(piece == PAWN)
    moveZTo(PAWN_Z);
}

// Moves the Z-arm to the right height
void moveZTo(int row)
{
  int changes=0;
  int theSensor=SENSOR_Z;
  int toMove=0;

  if(row > pointZ)
  {
    toMove = row-pointZ;
    OnFwd(MOTOR_Z);
    if(zWay==REVERSE)
      toMove++;
    zWay=FORWARD;
  }
  if(row < pointZ)
  {
    toMove = pointZ-row;
```

```
    OnRev(MOTOR_Z);
    if(zWay==FORWARD)
      toMove++;
    zWay=REVERSE;
  }

  while(changes < toMove)
  {
    if((SENSOR_Z > 0 && theSensor == 0) || (SENSOR_Z == 0 && theSensor > 0))
    {
      PlayTone(430,10);
      changes++;
      theSensor=SENSOR_Z;
    }
  }
  pointZ=row;
  Off(MOTOR_Z);

}

// Moves the X-vehicle to the right row
void moveXTo(int row)
{
  int changes=0;
  int theSensor=SENSOR_X;
  int toMove=0;
  if(row > vehicleX)
  {
    toMove = row-vehicleX;
    OnFwd(MOTOR_X);
    if(xWay==REVERSE)
      toMove++;
    xWay=FORWARD;
  }
  if(row < vehicleX)
  {
    toMove = vehicleX-row;
    OnRev(MOTOR_X);
    if(xWay==FORWARD)
      toMove++;
    xWay=REVERSE;
  }
  while(changes < toMove)
  {
    if((SENSOR_X > 0 && theSensor == 0) || (SENSOR_X == 0 && theSensor > 0))
    {
      PlayTone(430,10);
      changes++;
      theSensor=SENSOR_X;
    }
  }
  vehicleX=row;
  Off(MOTOR_X);
}
// Moves the Y-vehicle to the right row
void moveYTo(int row)
{
  int changes=0;
  int theSensor=SENSOR_Y;
  int toMove=0;

  if(row > vehicleY)
  {
    toMove = row-vehicleY;
```

```
    OnFwd(MOTOR_Y);
    if(yWay==REVERSE)
      toMove++;
    yWay=FORWARD;
  }
  if(row < vehicleY)
  {
    toMove = vehicleY-row;
    OnRev(MOTOR_Y);
    if(yWay==FORWARD)
      toMove++;
    yWay=REVERSE;
  }
  while(changes < toMove)
  {
    if((SENSOR_Y > 0 && theSensor == 0) || (SENSOR_Y == 0 && theSensor > 0))
    {
      PlayTone(430,10);
      changes++;
      theSensor=SENSOR_Y;
    }
  }
  vehicleY=row;
  Off(MOTOR_Y);
}
```

Källkoden till RCX2

Programmet ligger bara och väntar på ett meddelande. Är meddelandet 156 eller 157 öppnar respektive stänger RCX:en klon.

```
#define MOTOR    OUT_A
#define SEND_OPEN  156
#define SEND_CLOSE 157

// Main task
task main()
{
  int message;

  while(true)
  {
    // Wait for message
    ClearMessage();
    while(Message()==0);

    PlayTone(523,20);
    message=Message();

    if(message==SEND_OPEN)
    {
      Fwd(MOTOR);
      OnFor(MOTOR,50);
    }
    if(message==SEND_CLOSE)
    {
      OnRev(MOTOR);
    }
  }
}
```

Använda program

LeoCAD
CAD-program för lego. Fri programvara. www.leocad.org

POV-RAY
3D-rendering. Gratisprogram. www.povray.org

GIMP, GNU Image Manupulation Program
Grafikhantering. Fri programvara. www.gimp.org

MLCAD
Ett till CAD-program för lego. Gratisprogram. www.lm-software.com/mlcad/

L3P
Konverterar lego-filer till POV-ray-filer. Gratisprogram. http://home16.inet.tele.dk/hassing/l3p.html

LDRAW
Innehåller legobitarna till legocadprogrammen. Fri programvara. www.ldraw.org

Star Office
Ordbehandlare. Gratisprogram. www.sun.com