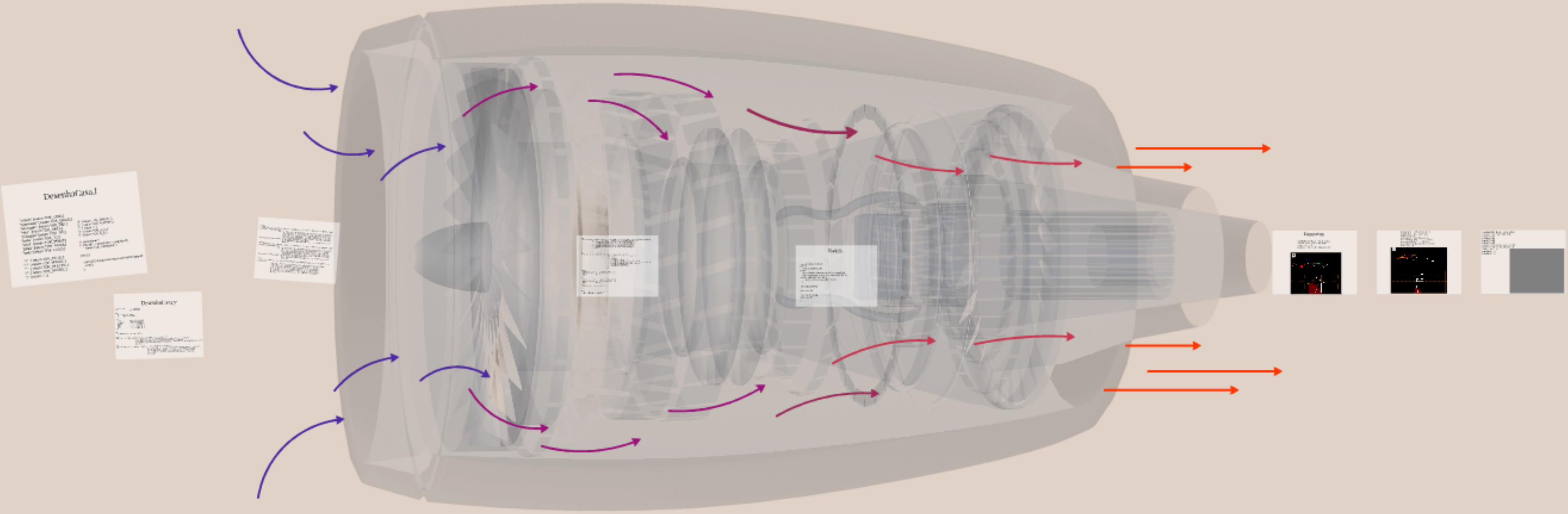
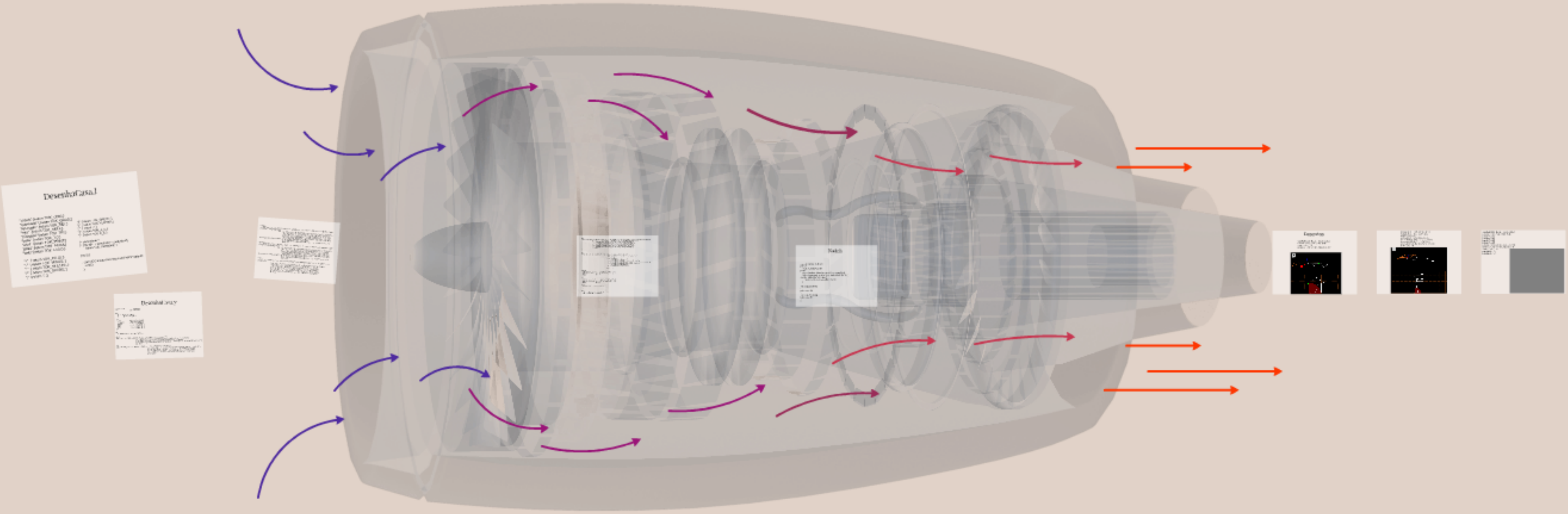


Trabalho de Compiladores

DesenhaCasas

Roberto Goulart
Carlos Francisco



Trabalho de Compiladores

DesenhaCasas

Roberto Goulart
Carlos Francisco

DesenhaCasa.l

```
"circulo" {return TOK_CIRC;}
"quadrado" { return TOK_QUAD;}
"retangulo" {return TOK_RET;}
"reta"  {return TOK_RETA;}
"triangulo" {return TOK_TRI;}
"para"  {return TOK_TO;}
"print"  {return TOK_PRINT;}
"pinta" {return TOK_PINTA;}
"lado" {return TOK_LADO;}

"+"  { return TOK_PLUS; }
"-"  { return TOK_MINUS; }
"*"  { return TOK_MULTPL; }
"/"  { return TOK_DIVIDE; }
","  {return ',' ;}
```

```
"("  { return TOK_OPENP; }
")"  { return TOK_CLOSEP; }
";"  { return ';'; }
"{"  { return TOK_A_C;}
"}"  {return TOK_F_C;}

 /* comentario */
[+-]?[0-9]+   { yylval.nint = atoi(yytext);
       return TOK_INTEGER; }

[ \t\n] {}

.  { printf("Caracter nao esperado! %s\n",yytext);
    exit(1);
  }
```

DesenhaCasa.y

```
programa : stmts                      { Program p;
                                  p.generate($1); };

stmts :
      stmts stmt {$1->append($2); }
    | stmt         {$$ = new Stmts($1);}

stmt
    :circulo ';'                    {$$ = new Capsule($1);}
    |retangulo ';'                  {$$ = new Capsule($1);}
    |quadrado   ';                  {$$ = new Capsule($1);}
    |reta ';'                       {$$ = new Capsule($1);}
    |triang ';'                     {$$ = new Capsule($1);}
    |pinta ';'                      {$$ = new Capsule($1);}
    ;

print
: TOK_PRINT arith_expr { $$ = new Print($2); };

circulo
:TOK_CIRC TOK_OPENP coordD TOK_CLOSEP ',' TOK_OPENP coordD TOK_CLOSEP {
                                              Stmts *bloco = new Stmts( new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(1)));
                                              bloco->append(new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(0)));
                                              bloco->append( new ArcMove($3->getCoordX(),$3->getCoordY(), new Float(0), $7->getCoordX(), $7->getCoordY()));
                                              $$ = bloco;            };
reta
:TOK_RETA TOK_OPENP coordD TOK_CLOSEP TOK_TO TOK_OPENP coordD TOK_CLOSEP {
                                                                  Stmts *bloco = new Stmts( new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(1)));
                                                                  bloco->append(new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(0)));
                                                                  bloco ->append(new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(0)));
                                                                  bloco->append( new LinearMove($7->getCoordX(), $7->getCoordY(), new Float(0)));
                                                                  $$ = bloco; };
```

```
retangulo
:TOK_RET TOK_OPENP coordD TOK_CLOSEP ',' TOK_OPENP coordD TOK_CLOSEP ',' TOK_OPENP coordD TOK_CLOSEP ','
TOK_OPENP coordD TOK_CLOSEP
                        {Stmts *bloco = new Stmts( new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(1)));
                             bloco->append( new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(0)));
                             bloco->append( new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(0)));
                             bloco->append( new LinearMove( $7->getCoordX(), $7->getCoordY(), new Float(0)));
                             bloco->append( new LinearMove($11->getCoordX(), $11->getCoordY(), new Float(0)));
bloco->append( new LinearMove($15->getCoordX(), $15->getCoordY(), new Float(0)));
                             bloco->append( new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(0)));
                             $$ = bloco;};
quadrado
: TOK_QUAD TOK_OPENP coordD TOK_CLOSEP ',' TOK_OPENP coordD TOK_CLOSEP ',' TOK_OPENP coordD TOK_CLOSEP ','
TOK_OPENP coordD TOK_CLOSEP
                        {Stmts *bloco = new Stmts( new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(1)));
                             bloco->append(new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(0)));
                             bloco->append(new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(0)));
                             bloco->append( new LinearMove( $7->getCoordX(), $7->getCoordY(), new Float(0)));
                             bloco->append( new LinearMove($11->getCoordX(), $11->getCoordY(), new Float(0)));
bloco->append( new LinearMove($15->getCoordX(), $15->getCoordY(), new Float(0)));
                             bloco->append( new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(0)));
                             $$ = bloco;};
triang
:TOK_TRI TOK_OPENP coordD TOK_CLOSEP TOK_TO TOK_OPENP coordD TOK_CLOSEP TOK_TO TOK_OPENP coordD
TOK_CLOSEP
                {Stmts *bloco = new Stmts( new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(1)));
                 bloco->append(new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(0)));
                   bloco->append( new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(0)));
                 bloco->append(new LinearMove($7->getCoordX(), $7->getCoordY(), new Float(0)));
                 bloco->append( new LinearMove($11->getCoordX(), $11->getCoordY(), new Float(0)));
                 bloco->append(new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(0)));
                 $$ = bloco;};
```

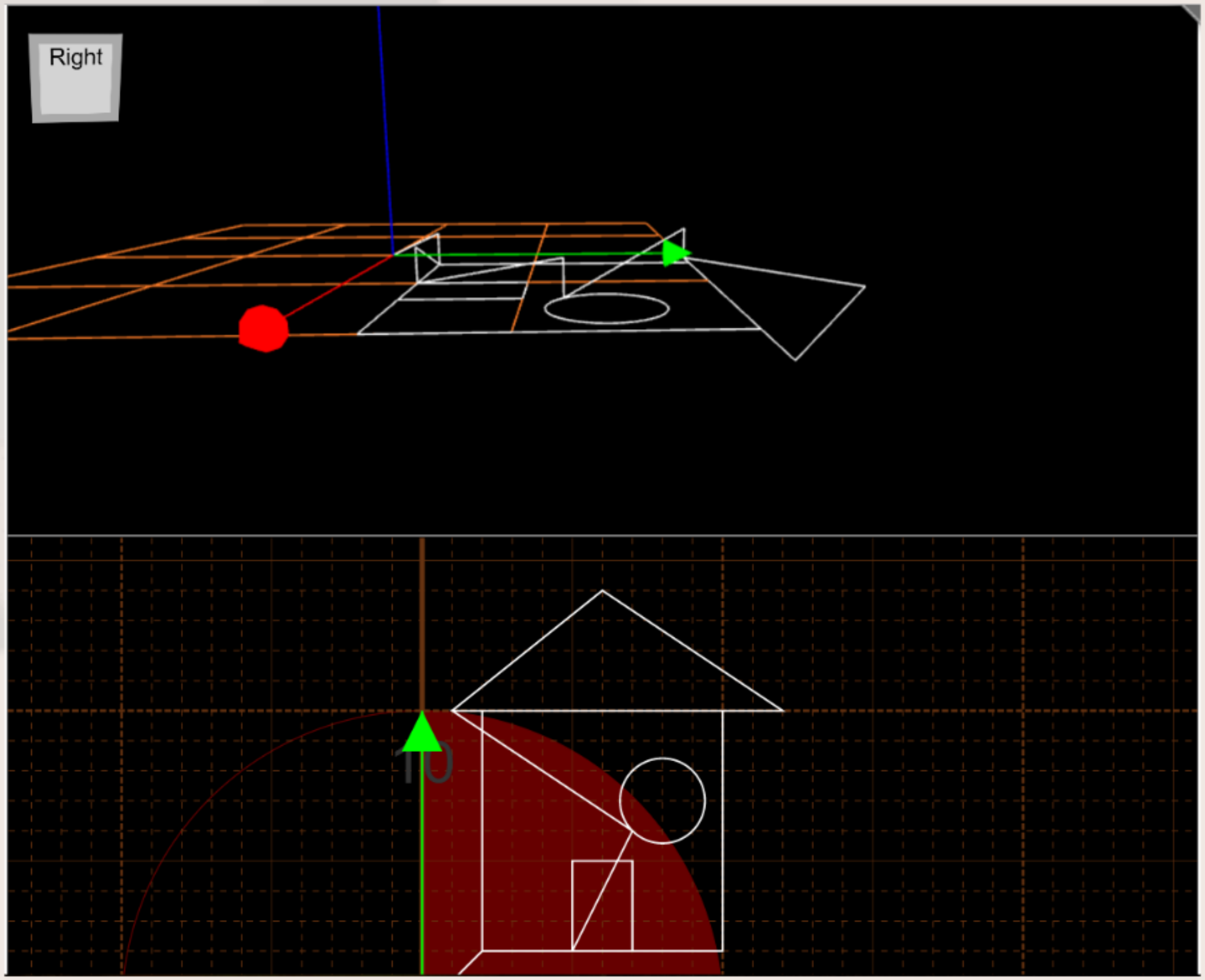
```
triang
:TOK_TRI TOK_OPENP coordD TOK_CLOSEP TOK_TO TOK_OPENP coordD TOK_CLOSEP TOK_TO TOK_OPENP coordD TOK_CLOSEP
                                   {Stmts *bloco = new Stmts( new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(1)));
                                    bloco->append(new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(0)));
                                       bloco->append( new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(0)));
                                    bloco->append(new LinearMove($7->getCoordX(), $7->getCoordY(), new Float(0)));
                                    bloco->append( new LinearMove($11->getCoordX(), $11->getCoordY(), new Float(0)));
                                    bloco->append(new LinearMove($3->getCoordX(), $3->getCoordY(), new Float(0)));
                                    $$ = bloco;}
;


pinta
    : TOK_PINTA TOK_OPENP factor ',' factor TOK_CLOSEP ',' TOK_OPENP factor TOK_CLOSEP  {
                                                                         Stmts *bloco = new Stmts(new LinearMove($3, $5, new Float(1)));

                                                                        for(int i=0; i<100; i++){
                                                                            bloco->append(new LinearMove($3, $5));
                                                                            bloco->append(new LinearMove($3, new BinaryOp($5, '+', $9)));
                                                                            bloco->append(new LinearMove(new BinaryOp($3, '+', $9), new BinaryOp($5, '+', $9)));
                                                                            bloco->append(new LinearMove(new BinaryOp($3, '+', $9), $5));
                                                                            bloco->append(new LinearMove($3, $5));
                                                                            $9 = new BinaryOp($9, '-', new Float(0.03));
                                                                         }
                                                                        $$ = bloco;}

arith_expr
    : arith_expr TOK_PLUS expr          {$$ = new BinaryOp($1, '+', $3);}
    | arith_expr TOK_MINUS expr         {$$ = new BinaryOp($1, '-', $3);}
    | factor                            {$$ = $1;}
    ;

expr
    : expr TOK_MULTPL factor                {$$ = new BinaryOp($1, '*' , $3);}
    | expr TOK_DIVIDE factor              {$$ = new BinaryOp($1, '/' , $3);}
    |factor                             {$$ = $1;}
    ;
coordD
    : factor ',' factor {$$ = new Coord($1 , $3);}

    ;

factor
    : TOK_OPENP arith_expr TOK_CLOSEP {$$ = $2; }
    | TOK_INTEGER  { $$ = new Float($1); }
    ;
```

Node.h

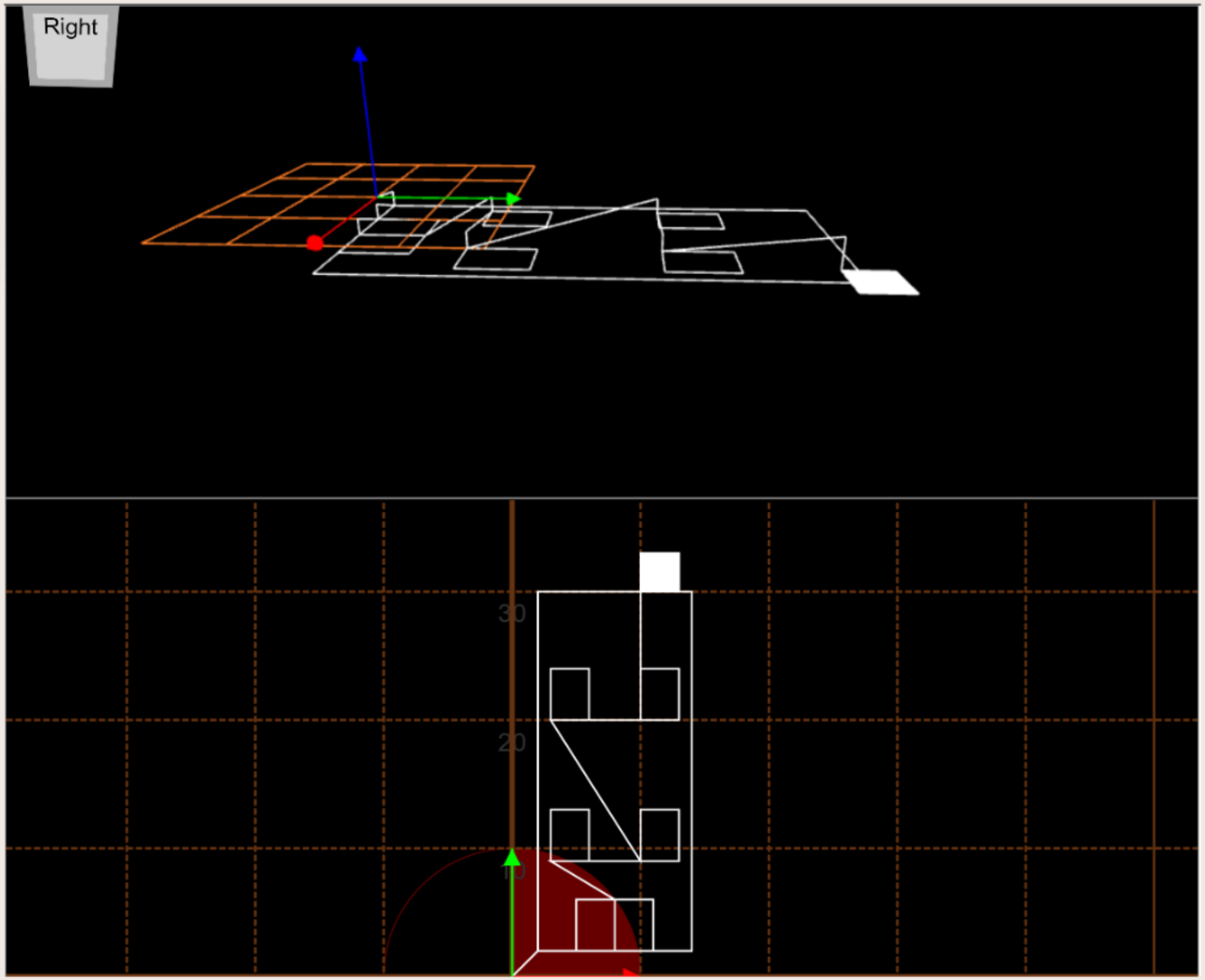
```
class Coord: public Node {
private:
    Node *coordX,*coordY;
public:
    Coord(Node* x, Node* y): coordX(x), coordY(y) {}
    Value *generate(Function *func, BasicBlock *block) {
coordX->generate(func, block);
        return coordY->generate(func, block);
    }

Node* getCoordX()
{
return coordX;
}
Node* getCoordY(){
return coordY;
}
};
```

Desenhos

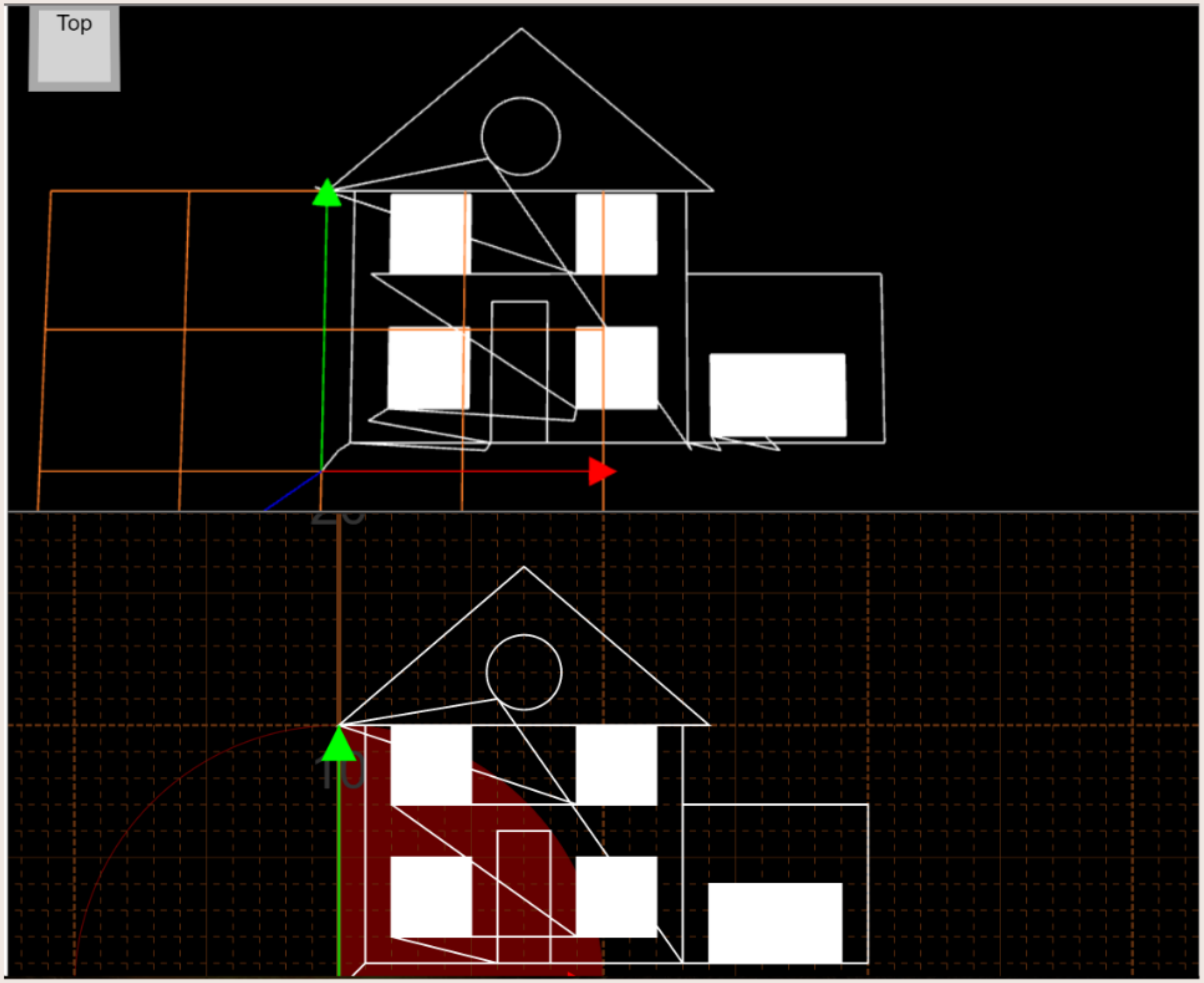
quadrado(2,2) , (10,2) , (10,10) , (2,10);
retangulo(5,2) , (7,2) , (7,5) , (5,5);
circulo(7,6) , (1,1);
triangulo(1,10) para (12,10) para (6,14);

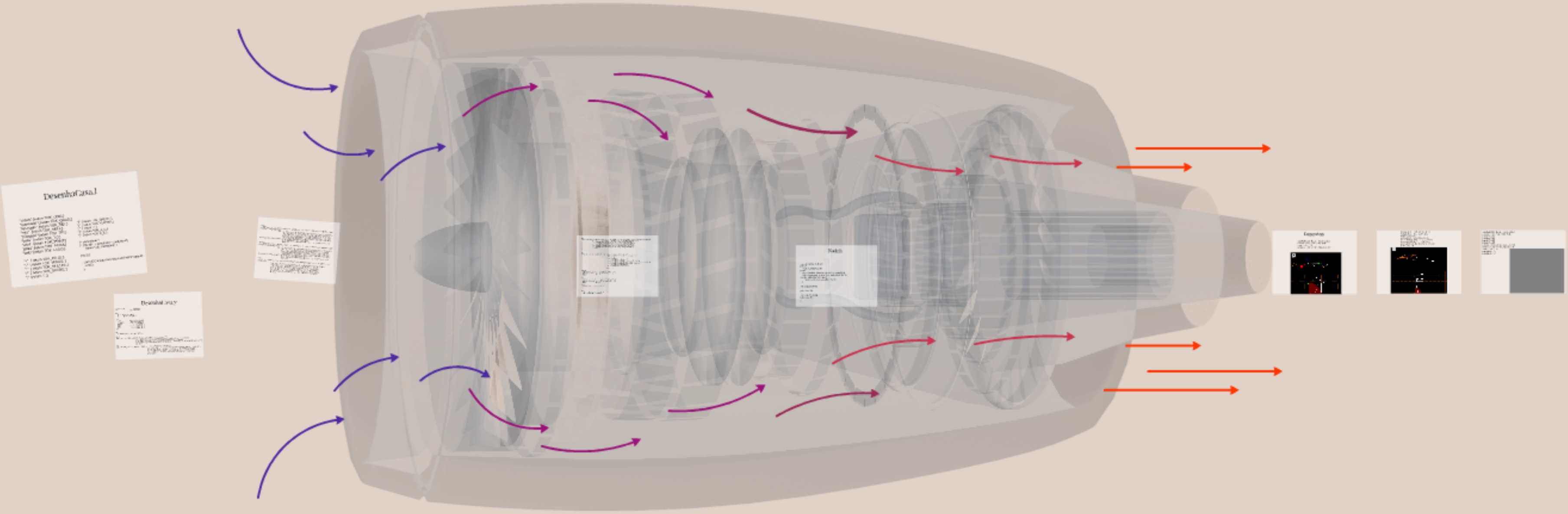


```
retangulo(2,2) , (14,2) , (14,30) , (2,30);
retangulo(5,2) , (11,2) , (11,6) , (5,6);
reta(8,2) para (8,6);
quadrado(3,9) , (6,9) , (6,13) , (3,13);
quadrado(10,9) , (13,9) , (13,13) , (10,13);
quadrado(3,20) , (6,20) , (6,24) , (3,24);
quadrado(10,20) , (13,20) , (13,24) , (10,24);
pinta(10,30) , (3);
```



```
quadrado(1,1) , (13,1) , (13,10) , (1,10);
retangulo(6,1) , (8,1) , (8,6) , (6,6);
pinta(2,2) , (3);
pinta(9,2) , (3);
pinta(2,7) , (3);
pinta(9,7) , (3);
retangulo(13,1) , (20,1) , (20,7) , (13,7);
triangulo(0,10) para (14,10) para (7,16);
circulo(6,11) , (1,1);
pinta(14,1) , (3);
pinta(16,1) , (3);
```





Trabalho de Compiladores

DesenhaCasas

Roberto Goulart
Carlos Francisco