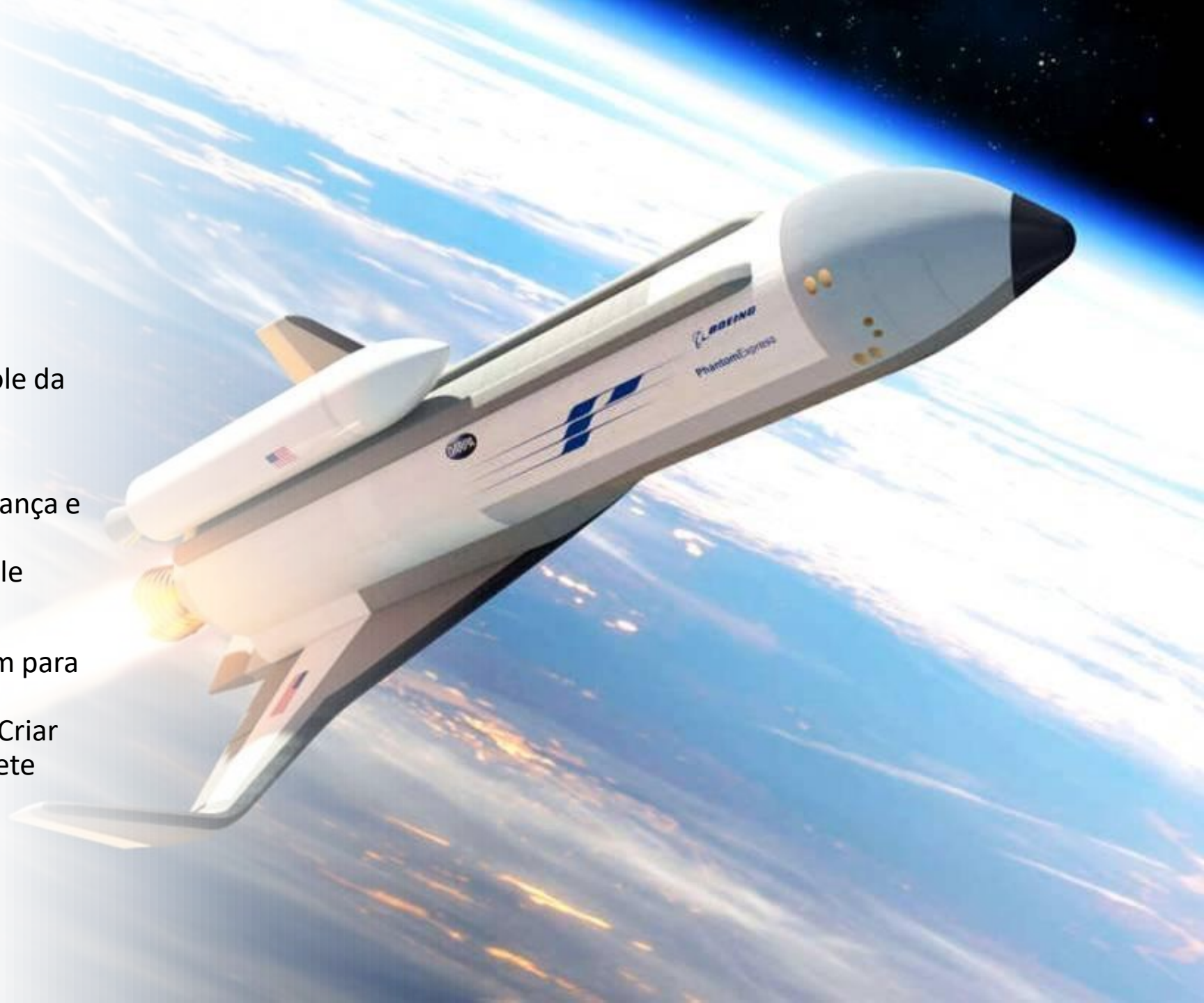
Lógica da Computação – APS – Void Language

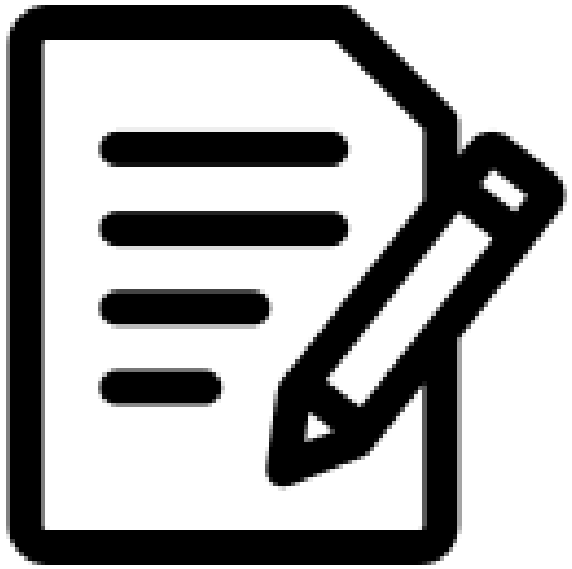
Aluno: Diogo dos Reis Duarte

Motivação

- Simplificar os códigos de controle da nave
- Ao simplificar, aumenta a segurança e o entendimento sobre a nave, diminuindo o desafio do controle
- Possíveis iterações da linguagem para contribuir para descobertas científicas, como por exemplo: Criar uma palavra em que a nave colete amostras de um planeta.



Estrutura da Void Language, no padrão EBNF



- PROGRAM = STATEMENT
- BLOCK = "{" STATEMENT "}“
- STATEMENT = (λ | ASSIGNMENT | PRINT | VARIABLE | FLIGHT_CONTROLLER | SPACIAL_COMANDS), "\n"
- FLIGHT_CONTROLLER = "pousar" | "decolar" | "ajustar_ângulo"
- SPACIAL_COMANDS = "ativar_foguete", "(", INT, ")" | "alinhamento-orbita", "(", INT, ")" | "ajustar_posição", "(", INT, ",", INT, ")"
- VARIABLE = "var", IDENTIFIER, TYPE
- ASSIGNMENT = IDENTIFIER, "=", EXPRESSION
- PRINT = "Print", "(", EXPRESSION, ")"
- IF = "if", "(", BOOLEXP, ")", BLOCK, ("else", BLOCK)?
- LOOP = "loop", "(", BOOLEXP, ")", BLOCK
- BOOLEXP = EXPRESSION, { ("<" | ">" | "=="), EXPRESSION }
- EXPRESSION = TERM, { ("+" | "-"), TERM }
- TERM = FACTOR, { ("*" | "/"), FACTOR }
- FACTOR = (("+" | "-"), FACTOR) | INT | "(", EXPRESSION, ")" | IDENTIFIER
- TYPE = INT | STR
- IDENTIFIER = LETTER, { LETTER | DIGIT | "_" }
- INT = DIGIT, { DIGIT }
- STR = LETTER, { LETTER | DIGIT};
- LETTER = (a | ... | z | A | ... | Z);
- DIGIT = (1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0);

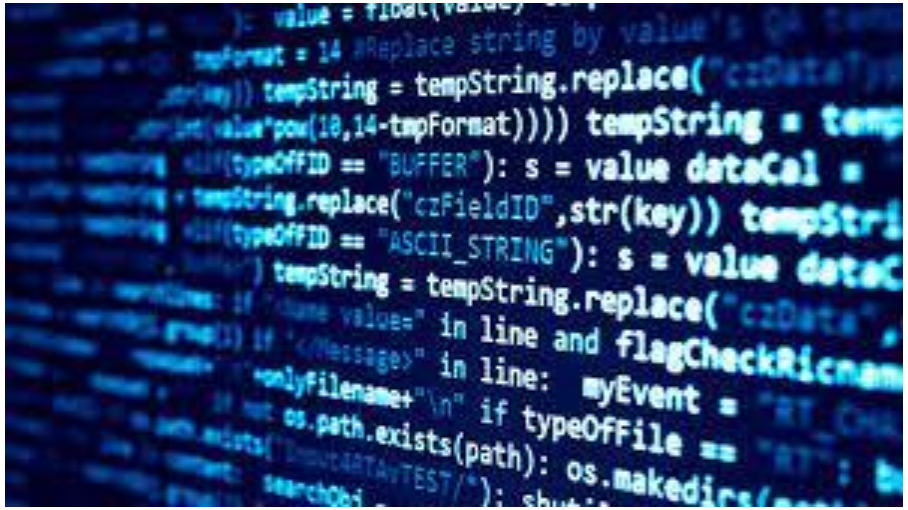
Características da Linguagem

Esta linguagem tem a capacidade de integração com o sistema de uma nave espacial

Fácil entendimento dos acontecimentos, visto que praticamente tudo que ocorre, a nave tem um output de resposta

A linguagem suporta a criação de variáveis, condicionais, e loops. Com isso, a linguagem oferece diferentes formas de codificar

Fácil iteração da linguagem, visto que caso seja necessário a implementação de novas palavras de voo, precisaria mudar apenas na parte da EBNF, a qual são as FLIGHT_CONTROLLER e SPACE_COMANDS, e com isso mudar o tokenizer, parser e o node com suas funcionalidades



Exemplo

Um exemplo de código com as novas implementações feitas, e como seria o output:

```
C:\Users\diogo\OneDrive\Documentos\Insper\setimoSemestre\LogComp\LogComp-APS\Compilador>python main.py test.void
A nave será ligada em 30 segundos
A nave está decolando
90
91
92
93
94
95
96
97
98
99
A nave está ajustando a sua angulação para o refencial predefinido
A nave está ajustando a sua posição para a coordenada (10, 20)
A nave está pousando
A angulação final foi:
90
A coordenada x final foi:
10
A coordenada y final foi:
20
```

```
var ang int
var x int
var y int

var a string
var b string
var c string

a = "A angulação final foi: "
b = "A coordenada x final foi: "
c = "A coordenada y final foi: "

x = 6
y = 3

ativar_foguete(30)
decolar


for ang = 90; ang < 100; ang = ang + 1 {
    Println(ang)
}

ajustar_angulo
ang = 90

if x == 6 || y == 3{
    x = 10
    y = 20
}

ajustar_posicao(x,y)
pousar

// Saida final
Println(a)
Println(ang)
Println(b)
Println(x)
Println(c)
Println(y)
```