

```

int main(int argc, char** argv)
{
    char c = 0;
    char* commands = "ads pq"; // key commands: "left,right,rotate,confirm,pause,quit"
    int speed = 2; // sets max moves per row
    int moves_to_go = 2;
    int full = 0; // whether board is full
    init(); // initialize board an tetrominoes

```

MAC122 - PRINCÍPIOS DE DESENVOLVIMENTO DE ALGORITMOS

Apresentação do curso

```

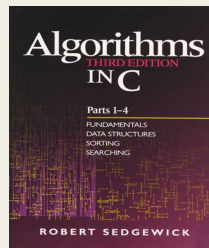
// process user action
c = getchar(); // get new action
if (c == commands[0] && !intersect(cur, state[0]-1, state[1])) state[0]--; // move left
if (c == commands[1] && !intersect(cur, state[0]+1, state[1])) state[0]++; // move right
if (c == commands[2] && !intersect(cur->rotated, state[0], state[1])) cur = cur->rotated;
if (c == commands[3]) moves_to_go=0;

// scroll down
if (!moves_to_go--)
{
    if (intersect(cur,state[0],state[1]+1)) // if tetromino intersected with sth
    {
        cramp_tetromino();
        remove_complete_lines();
        cur = &tetrominoes[rand() % NUM_POSES];
        state[0] = (WIDTH - cur->width)/2;

```

- ▶ Curso introdutório ao desenvolvimento de algoritmos
- ▶ **Pré-requisitos:** familiaridade com alguma linguagem de programação procedural (Python, C, Java, etc)
- ▶ Página: e-disciplinas
- ▶ Comunicação: mensagens no e-disciplinas ou por email `ddm@ime.usp.br`
- ▶ Submissão de exercícios e projeto **exclusivamente** pela página (envie com antecedência para se prevenir de falhas de sistema etc).

- ▶ Livro-texto: Paulo Feofiloff, Algoritmos em C, 2009. Ed. Campus/Elsevier (cópias na Biblioteca)
- ▶ Site do livro: <http://www.ime.usp.br/~pf/algoritmos/>
- ▶ Recomendado: R. Sedgewick, Algorithms in C, 3a. ed, vol. 1, Addison-Wesley/Longman, 1998.



O que eu **espero** que vocês aprendam

- ▶ **Princípios básicos** de desenvolvimento de software: corretude, eficiência, elegância, documentação
- ▶ **Estruturas de dados** elementares: pilhas, filas, listas encadeadas, árvores binárias
- ▶ **Algoritmos** para tarefas **elementares**: ordenação, enumeração, busca

Por que C? (ou por que não Python/Matlab/Go/Lisp/...?)

- ▶ C possui uma **sintaxe robusta** (p.ex., Python 3 não é compatível com Python 2.7)
- ▶ C possui **compiladores confiáveis**
- ▶ C permite **programação de baixo nível** a médio nível, o que é interessante para a aprendizagem
- ▶ C é **rápido e popular** (é fácil encontrar resposta para sua dúvida na internet)
- ▶ Boa parte das linguagens imperativas tem sintaxe parecida

Por que C? (ou por que não Python/Matlab/Go/Lisp/...?)

C lhe permite não apenas dizer o que fazer mas exatamente como fazer

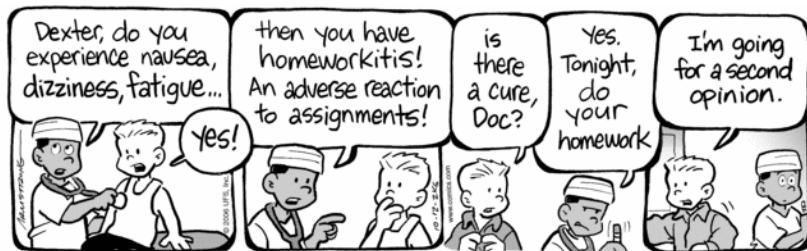
- ▶ Isso lhe dá mais flexibilidade ao programar
- ▶ Mas também torna mais difícil a depuração de erros
- ▶ Escrever programas elegantes é ainda mais crucial em C
- ▶ Muita atenção precisa ser dedicada ao uso de ponteiros

- ▶ Aulas às 3as (19:20 – 21:00) e 5as (21:10 – 22:50)
pontualmente!
- ▶ **Monitoria:** Igor (igorcs@ime.usp.br)
- ▶ Avaliação: 3 provas (SUB aberta), exercícios de programação (EPs) semanais, **projeto final** (editor de texto)

$$\text{Média Final} = \frac{1}{2} * \text{Provas} + \frac{1}{4} * \text{EPs} + \frac{1}{4} * \text{Projeto}$$

se cada critério for ≥ 5 , então

$$\text{Média Final} = \min\{\text{Provas}, \text{EPs}, \text{Projeto}\}$$



Para passar: Média dos EPs $\geq 70\%$

DENIS DERATANI MAUÁ

<http://ime.usp.br/~ddm>

ddm@ime.usp.br

Sala CCSL-218

Tel: 3091-5036