程序设计

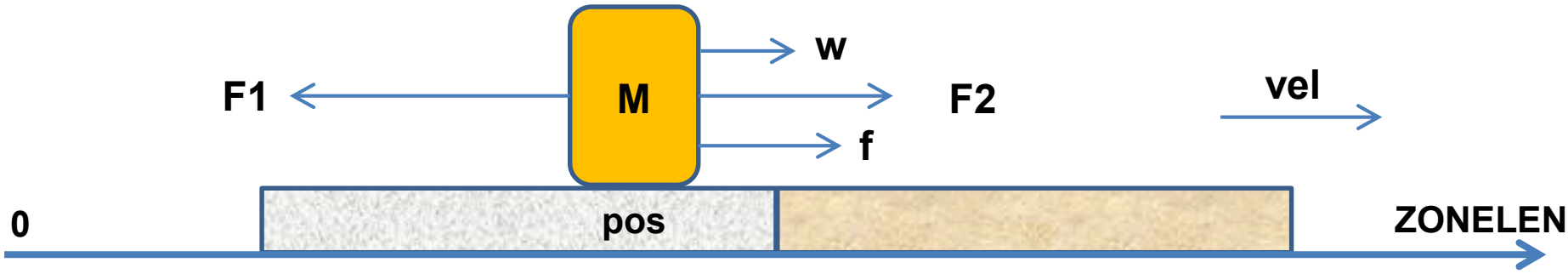
拔河比赛说明

林靖宇
广东工业大学自动化学院

内容提要

1. 物理模型
2. 比赛规则
3. 仿真程序说明

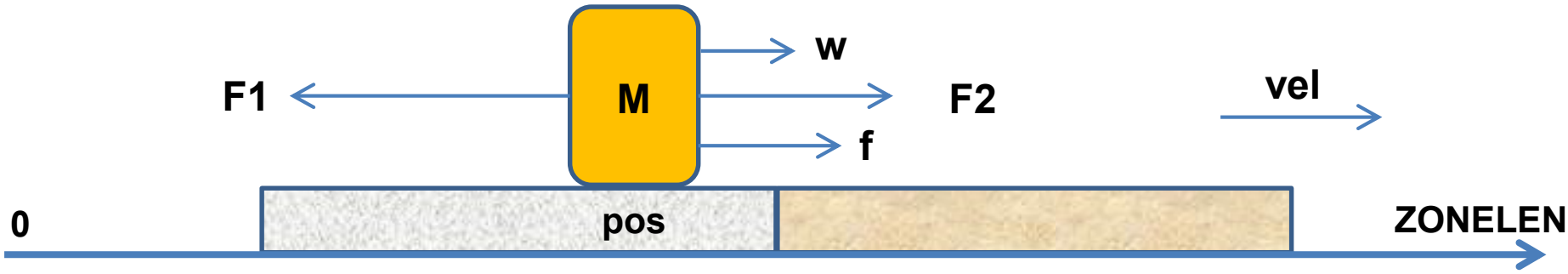
1. 物理模型



物理量说明：

- 比赛区域左端为坐标原点，向右为正向，长度为**ZONELEN**。
- 比赛区域中有一个滑块，质量**M = 1**，初始位置**pos=ZONELEN/2**，初始速度**vel=0**。
- 滑块受力包括左右玩家的拉力**F1**和**F2**、风力**w**、摩擦力**f**。
- 拉力**F1**、**F2**分别由左右玩家控制，方向分别向左和向右，幅值不限。
- 风力幅值**|w| ≤ WIND**，大小和方向随机变化。
- 摩擦力依赖运动状态。速度**vel**不为**0**时，动摩擦力幅值**|f| = 1**，方向与**vel**相反；**vel**为**0**时，静摩擦力幅值**|f| ≤ 1**，方向与合力相反，如果合力为**0**则摩擦力为**0**。

1. 物理模型



离散运动模型：

- 位置
- 速度
- 合力

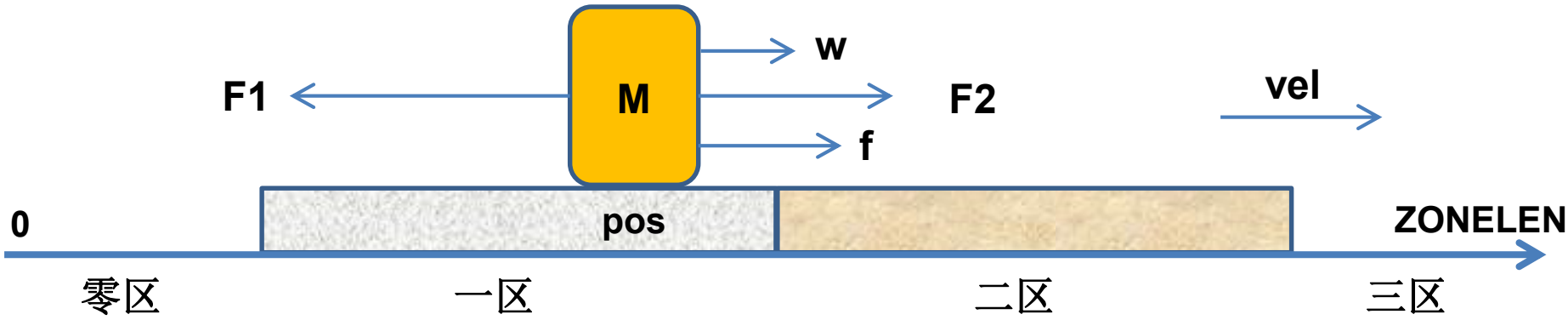
$$\begin{cases} \mathrm{pos}(t_s) = \mathrm{pos}(t_s - \Delta t) + \mathrm{vel}(t_s - \Delta t) \times \Delta t \\ \mathrm{vel}(t_s) = \mathrm{vel}(t_s - \Delta t) + \left[F(t_s) + f(t_s)\right] / M \times \Delta t \\ F(t_s) = \left[F_2(t_s) - F_1(t_s) + w(t_s)\right] / M \end{cases}$$

- 离散时间间隔$\Delta t = 1$

内容提要

1. 物理模型
2. 比赛规则
3. 仿真程序说明

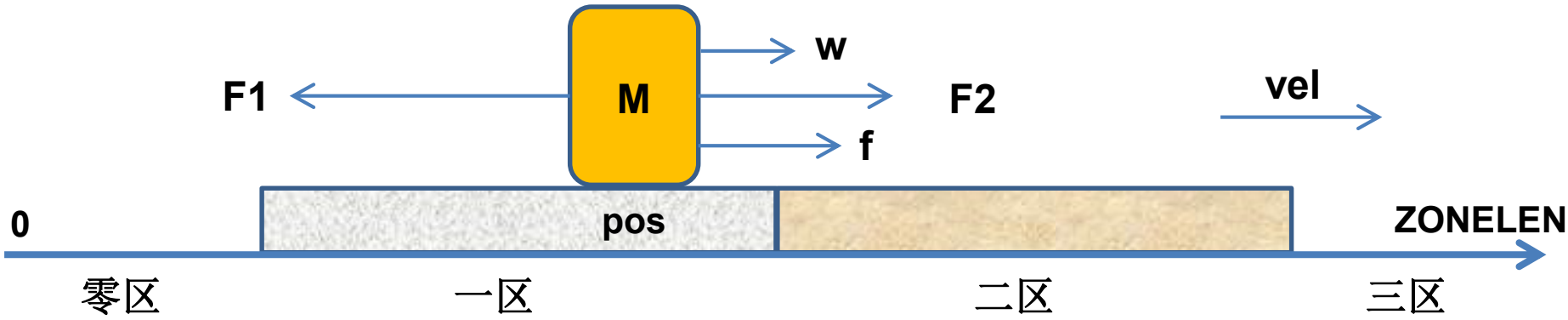
2. 比赛规则



规则说明：

- 比赛有两个玩家，比赛持续时间为**PLAYTIME**。
- 比赛区域分为四部分，从左到右为零区、一区、二区、三区，总长度为**ZONELEN** 。一区和二区长度均为**WIN_ZONE**，零区和三区长度相同。
- 比赛过程中，左右玩家获得滑块当前位置**pos**和速度**vel**，输出拉力**F1**、**F2**。
- 胜负判定如下：
  - 如果比赛过程中，滑块进入零区则右玩家胜，滑块进入三区则左玩家胜，判定结束。
  - 如果比赛结束时，滑块在一区则左玩家胜，滑块在二区则右玩家胜。

2. 比赛规则



玩家函数说明：

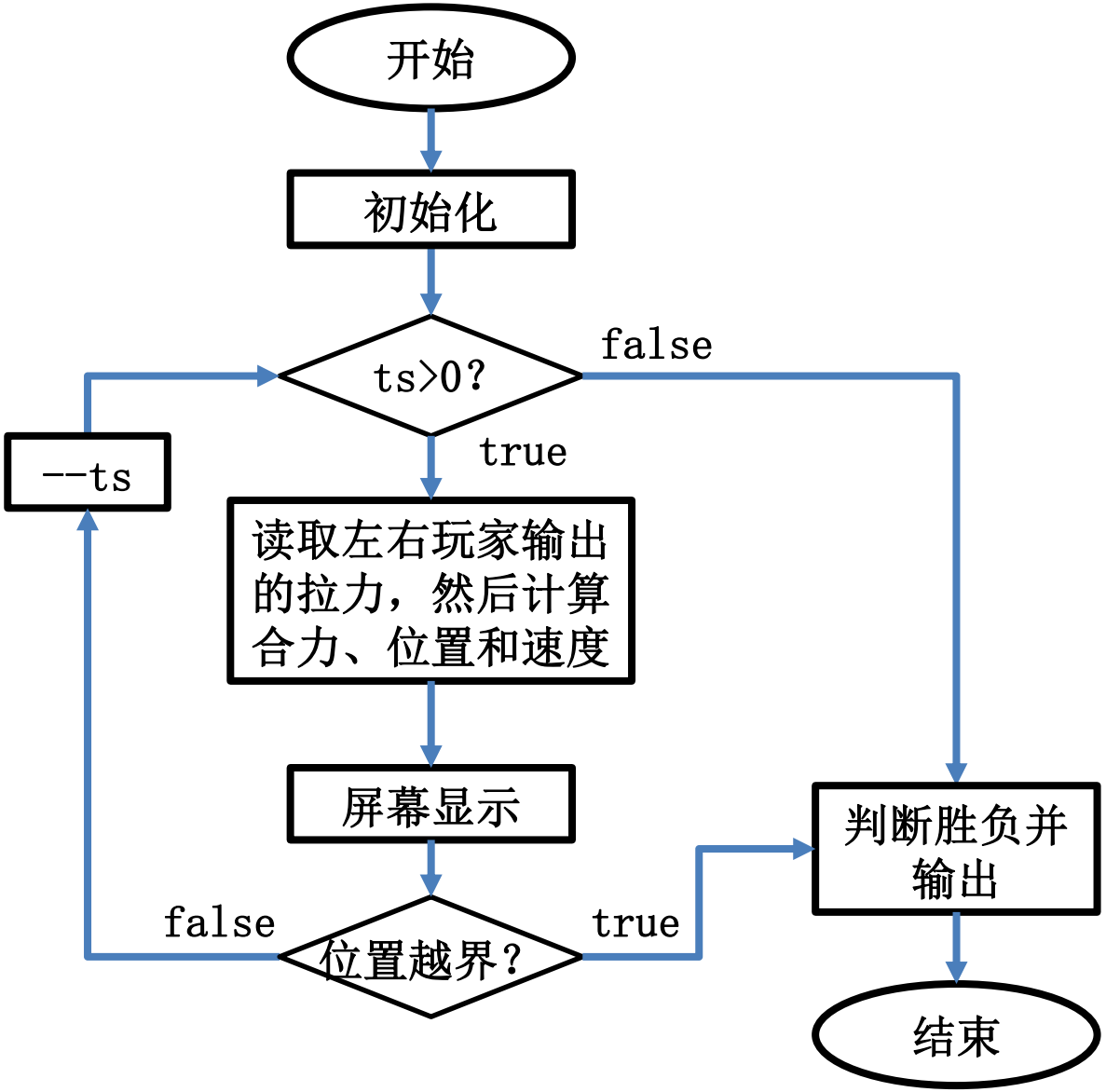
- 函数原型：int player1(struct war_info info)
- 玩家函数均为右玩家视角，可以随意交换位置。
- 输入参数info是结构war_info，包含四个成员：
  - pos是滑块坐标， vel是滑块速度， wind是实时风力，ts_left是比赛剩余时间
  - zone是比赛区域设置。zone[0]是一区左端坐标，zone[1]是二区左端坐标，zone[2]是三区左端坐标，zone[3]是比赛区域长度（即 ZONELEN ）。
- 玩家函数应根据参数信息计算拉力F2作为返回值。

内容提要

1. 物理模型
2. 比赛规则
3. 仿真程序说明

3. 仿真程序说明

主程序流程图



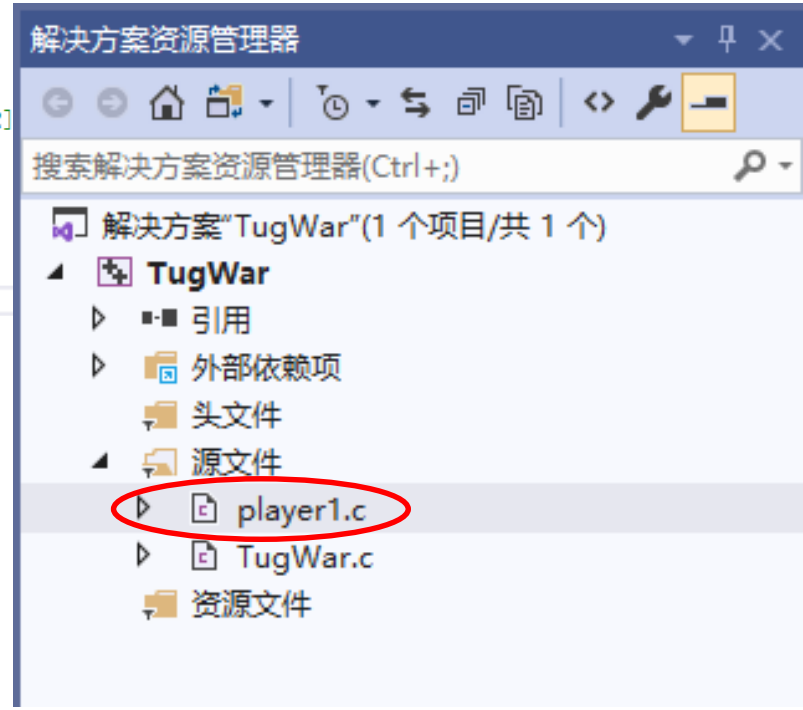
3. 仿真程序说明

使用说明：

①编写算法：在项目中添加一个c文件（例如player1.c），编写玩家函数。

player1_demo.c | TugWar.c

```
// player1.c

#include <stdlib.h>
#include <math.h>
struct war_info
{
    // 比赛信息
    double pos;      // 滑块当前位置
    double vel;      // 滑块当前速度
    double wind;     // 实时风力
    double ts_left; // 剩余时间（转换为整数）
    double zones[4];    // 比赛区域大小：[0] - 零区长度；[1] - 零区+一区长度；[2]

    // 玩家输出信息
    double player_acc;  // 玩家输出
    char* player_name;  // 玩家团队名称和成员姓名
};

/// <summary>
/// 在此实现进攻算法
/// </summary>
/// <param name="info">比赛信息，参见war_info结构的说明</param>
void player1(struct war_info* info)
{
    // 玩家团队名称和成员姓名
    info->player_name = "匿名";

    // 输出加速度
    info->player_acc = 0;

}
```



3. 仿真程序说明

使用说明：

①编写算法：在项目中添加一个**c**文件（例如**player1.c**），编写玩家函数。

②测试：在**TugWar.c**中，按照的注释说明声明玩家函数并设置左侧和右侧的玩家函数。例程**demo_player**可以作为其中一侧玩家。生成应用程序后运行。

```
/************************ 玩家函数 ************************/
// 在这里列出所有参赛玩家函数声明
void demo_player(struct war_info* info);    // 示例玩家
void random_player(struct war_info* info);  // 示例玩家
void player1(struct war_info* info);        // 示例玩家


// 在这里列出所有参赛玩家函数和名称
void (*playeracc[])(struct war_info* info) =
{
    demo_player, player1, random_player
};

// 正式比赛时，在命令行输入“TugWar <p1> <p2>”，其中<p1>、<p2>是参赛玩家编号
// 测试时，在这里选择本次参赛的玩家函数编号
int LEFT = 0;   // 左侧玩家
int RIGHT = 1;  // 右侧玩家
```

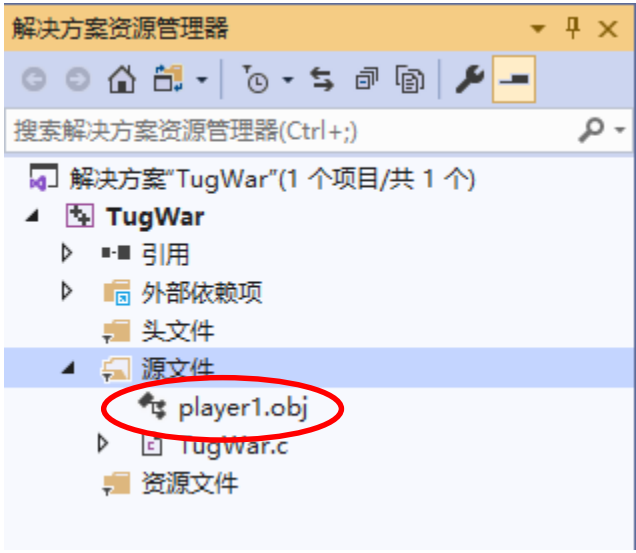
3. 仿真程序说明

使用说明：

③正式比赛

- 参赛者在项目目录中找到玩家函数的c文件，将c文件（例如player1.c）或相应的obj文件（例如player1.obj）发给裁判。
- 裁判在项目中添加所有参赛者的c文件或obj文件，在TugWar.c中按照注释说明在玩家函数列表中声明玩家函数，设置参赛玩家编号。

```
/************************ 玩家函数 ************************/
// 在这里列出所有参赛玩家函数声明
void demo_player(struct war_info* info);     // 示例玩家
void random_player(struct war_info* info);   // 示例玩家
void player1(struct war_info* info);         // 示例玩家


// 在这里列出所有参赛玩家函数和名称
void (*playeracc[])(struct war_info* info) =
{
    demo_player, player1, random_player
};

// 正式比赛时，在命令行输入“TugWar <p1> <p2>”，其中<p1>、<p2>是参赛玩家编号
```



3. 仿真程序说明

使用说明：

③正式比赛

- 参赛者在项目目录中找到玩家函数的**c**文件，将**c**文件（例如**player1.c**）或相应的**obj**文件（例如**player1.obj**）发给裁判。
- 裁判在项目中添加所有参赛者的**c**文件或**obj**文件，在**TugWar.c**中按照注释说明在玩家函数列表中声明玩家函数，设置参赛玩家函数。
- 裁判在**TugWar.c**中按照注释说明设置比赛参数（比赛过程中不能改动）。

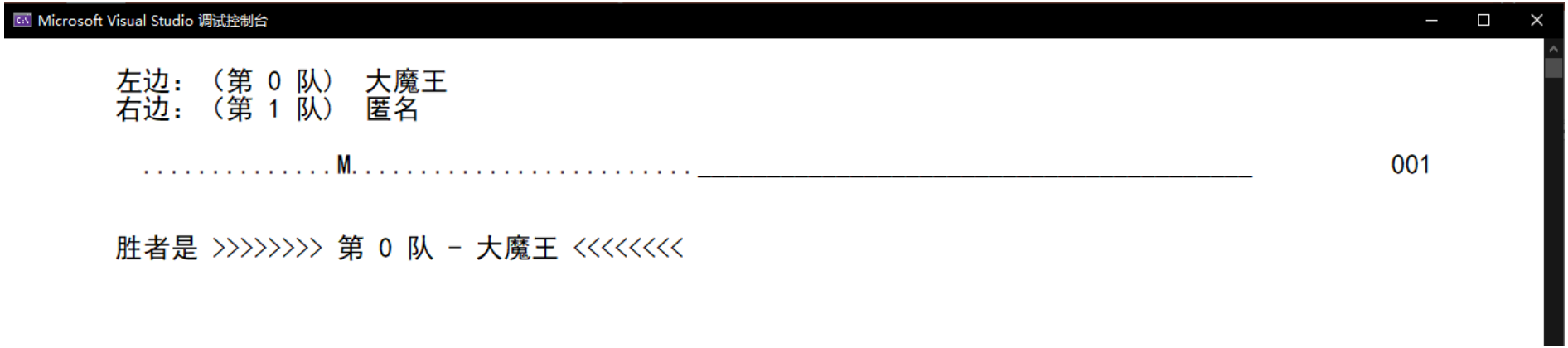
```
/************************* 在这里设置比赛参数 *************************/
#define PLAYTIME 200 /*比赛时间*/
#define SHOWRATE 100 /*显示间隔*/
#define WIND 2 /*最大风力*/
#define ZONELEN 80   /*比赛区长度*/
#define WIN_ZONE 30  /*胜负区长度*/
#define INIT_POS (ZONELEN / 2)  /*旗子初始位置*/
```

3. 仿真程序说明

使用说明：

③正式比赛

- 参赛者在项目目录中找到玩家函数的c文件，将c文件（例如player1.c）或相应的obj文件（例如player1.obj）发给裁判。
- 裁判在项目中添加所有参赛者的c文件或obj文件，在TugWar.c中按照注释说明在玩家函数列表中声明玩家函数，设置参赛玩家函数。
- 裁判在TugWar.c中按照注释说明设置比赛参数（比赛过程中不能改动）。
- 裁判生成和运行应用程序，公布比赛结果，结束本次比赛。



3. 仿真程序说明

默认比赛参数：

- 比赛时间**PLAYTIME**：**100**（节拍）
- 显示间隔**SHOWRATE**：**80**（**ms**）
- 最大风力**WIND**：**2**
- 比赛区长度**ZONELEN**：**100**（字符）
- 胜负区长度**WIN_ZONE**：**40**（字符）