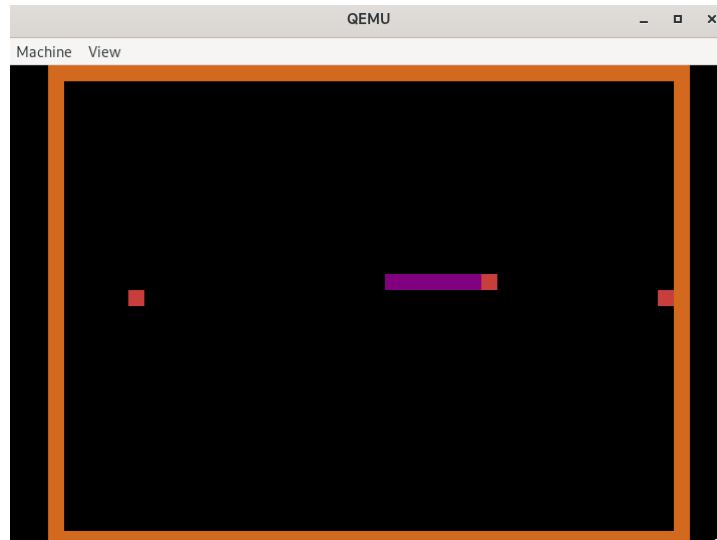


L0_REPORT

我实现的小游戏是贪吃蛇🐍



- 按键操作：

- 控制方向：W, S, A, D
- 暂停/继续：P/C
- 重启/结束：R/Esc
- 低速/中速/高速：1/2/3

- 部分实现细节：

- 代码架构：

```
1  init();
2  while(true){
3      while(true){
4          while(uptime()<last_time+speed){};
5          last_time=uptime();
6          mov=read_key();
7          snake_update(mov);
8          if(dead || succ) break;
9          render_screen();
10     }
11     restart();
12 }
```

- 使用结构体数组来定义一条蛇，这样，蛇增长时就只需要在数组末尾加入一个Node，当蛇长度为MAX_LEN时就赢得了游戏。

```
1  struct Snake{
2      int dire,len;
3      struct Node node[MAX_LEN];
4  };
```

- 设置了两个食物，当蛇吃了一个食物后就更新那个食物的位置。

- 使用 `speed` 来控制画面更新的速度从而达到蛇移动速度的改变，设置了三档速度。
 - 实现的难点：食物生成与边界控制导致了一些bug，因为要避免许多位置碰撞的问题，如食物出现在蛇身等。最后我统一使用了一个函数 `collision` 来判断位置是否合法。
- 受群友提示，加入了PA中的Log黑科技来输出信息：

```
1  #define Log(format,...)\
2      printf("\33[0m[\33[1;35mLog\33[0m]\33[1;34[%s,%d,%s] " format
3      "\33[0m\n", \
        __FILE__, __LINE__, __func__, ##__VA_ARGS__)
```