L0：amgame

徐臣 171180558

1. 实现

最终实现了一个简单的贪吃蛇游戏，玩家可以操作一只由上下左右控制方向的“蛇“，通过吃到屏幕上的白色方块达到增长自己的目的

2. 实现具体流程

```
struct time_flow{
    int last_FPS;
    int now_FPS;
    int upda;
}TI;
typedef struct _vec {
 int x;
 int y;
}vec;
typedef struct _body{
  int x;
  int y;
}body;
typedef struct _snake{
    body block;
    vec dir;
    struct _snake* next;
}snake;
```

实现中个结构体定义如下，由 TI 控制是否到达下一个更新的帧的对应时间（upda 表示已到达），vec 为蛇的方向向量，body 为蛇的位置，snake 本质为链表，保持从头到尾个部分的位置和运动。

```
now_fix=read_key();
if(now_fix!=NULL)
    puts(now_fix);
dir_change();
time_update();
if(update_enable()) {
    i++;
    if(i==10)   {
        new_blo();
        i=0;
    }
    inc_confirm();
    if(inc) {
        inc_snake(head);
    }
    redraw_snake();

}
```

程序主框架如下，在每一次循环中读取键盘信息，更新运动方向（dir_change），在帧时到达时，生成新的可吃方块（new_blo），并确认是否有方块被吃下（inc_confirm），并增长和更新蛇的位置（redraw_snake）。

3. 遇到的问题

1. 蛇的运动的更新问题：如何实现增长和运动？
   首先在无伸长的情况下，蛇的运动即为后面的格子运行至上一帧前方的格子处，并单独更新头尾（更新头，删除尾）。用链表容易实现。增长时，仅需不删除尾部即可（伸长的部位恰出现于上一帧的尾部）
2. 在何时选择更新图像
   最终我选择为在更新链表前删除尾，在链表更新后更新头，这样一来便不会出现时效性的问题
3. 方块的生成
   采用假的 random（乘法取模）生成对应的 x,y 坐标，在显示在屏幕上即可。
4. 边界处理
   统一采用取模，所以蛇会运动出屏幕，然后出现在对应的另一端。

5.在 qemu 实现后，在 native 中运行无图像：
使用 intel 的集成显卡即可，nivida 的独显无显示，估计为 ioe 的兼容问题。