

!重要：请务必将windows terminal窗口调至足够大！！

基本信息

- 操作系统：Windows 11
- C语言环境：TDM-gcc\x86_64-w64-mingw32
- 编译选项：vscode编写后直接由gcc编译可执行文件，无cMakeList
- 使用的第三方库： windows.h
- ！！ 如查看代码建议使用GBK编码标准打开

游戏方式

游戏规则

- 小恐龙向前冲,可以跳跃或下蹲
- 遇见障碍物即游戏结束
- 游戏分数随时间增长(屏幕下方显示)
- '#' 代表障碍物，分为空中的鸟和地上的仙人掌
- '\$' 代表金币，吃到可以加50分
- 'p' (extended ascii: 237) 代表无敌，吃到可以无敌一段时间(屏幕下方进度条显示无敌时间)

操作方式

- 键盘上键'↑'或空格键：跳跃
- 键盘下键'↓'：下蹲
- 下蹲过程中按跳马上能跳起，空中按下蹲马上能瞬间到蹲的状态
- 游戏过程中随时按'q'暂停，按'esc'结束
- 游戏结束后可以按'a'再来一把,按'q'推出程序

关于项目

基本思路

1. 使用字符界面, 利用Sleep () 以及重复清屏加输出字符数组达到动态效果, 以画面左上角为[0,0]原点
2. 通过互联网获取windows下的改变光标位置和隐藏光标函数，做到不需要每次打印整个画面。
3. 所有的实体（恐龙、障碍）都是结构体(障碍物是链表)，通过判断结构体中的xy是否重叠来判断碰撞
4. 游戏结束后使用goto语句实现是否再来一把
5. 实现了游戏速度随着分数加快，使用 $\text{Sleep}((100 - (\text{score} / 50)) \geq 10 ? (100 - (\text{score} / 50)) : 0)$;从而在加快的同时保证了不会出现Sleep(ms<10)这样过快的游戏

有价值的构思

1. 为了模拟更加真实的物理引擎,在跳跃时,距离地面较近的两帧每帧跨越两格,距离最高处较近的两帧每帧跨越一格(使用结构体变量dino.jump_time控制)
2. 因为要在场上同时出现多个同种障碍物,势必要用数组,但是障碍物消失后,由于下标小的障碍物是先出现先消失,单纯数组无法做到有效的修改操作,因此在思考和一些尝试过后选择使用链表来实现
3. 为了防止多个障碍物出现的过近导致必杀,使用 $\text{if}(p \rightarrow x \leq \text{scrX} - 20)$,其中p是障碍物链表指针,从而实现不会出现过近的障碍物
4. 为了实现在随机的同时不会出现空屏幕,或者过长时间不生成道具,在生成时除了使用随机数以外,还加了一条判断:对于障碍物,是若场上没有障碍物就生成障碍物;对于道具,是每达到固定分数必定会生成一个道具

小疑惑

1. 在他人提醒下得知方向键第一次读取时返回的竟然是224,第二次才会返回实际键值,可是奇怪的是在我没有写第二次读取时,游戏也可以正常运行,是什么缘故?