

L0:amgame

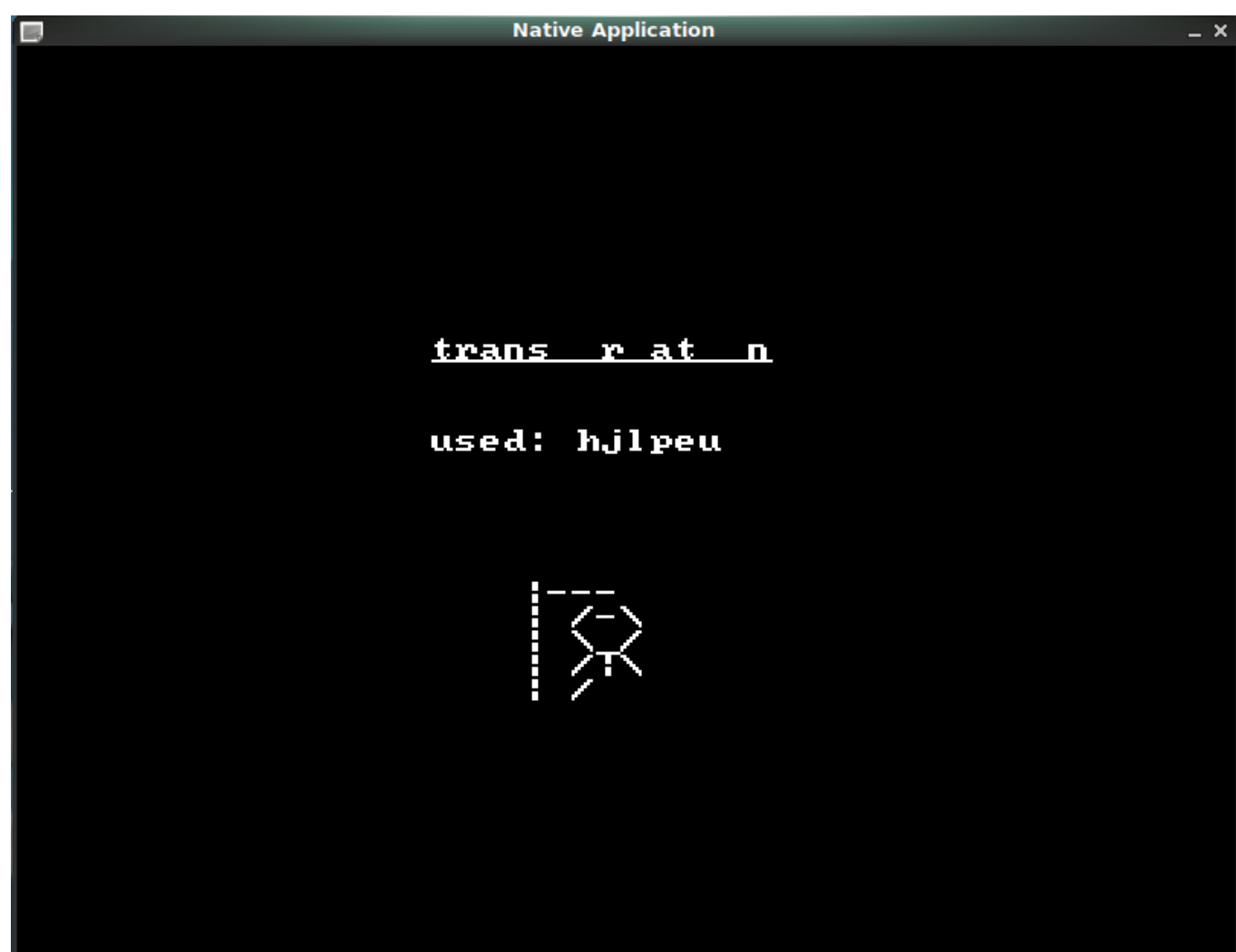
171860575 毛一鸣

实现游戏简介

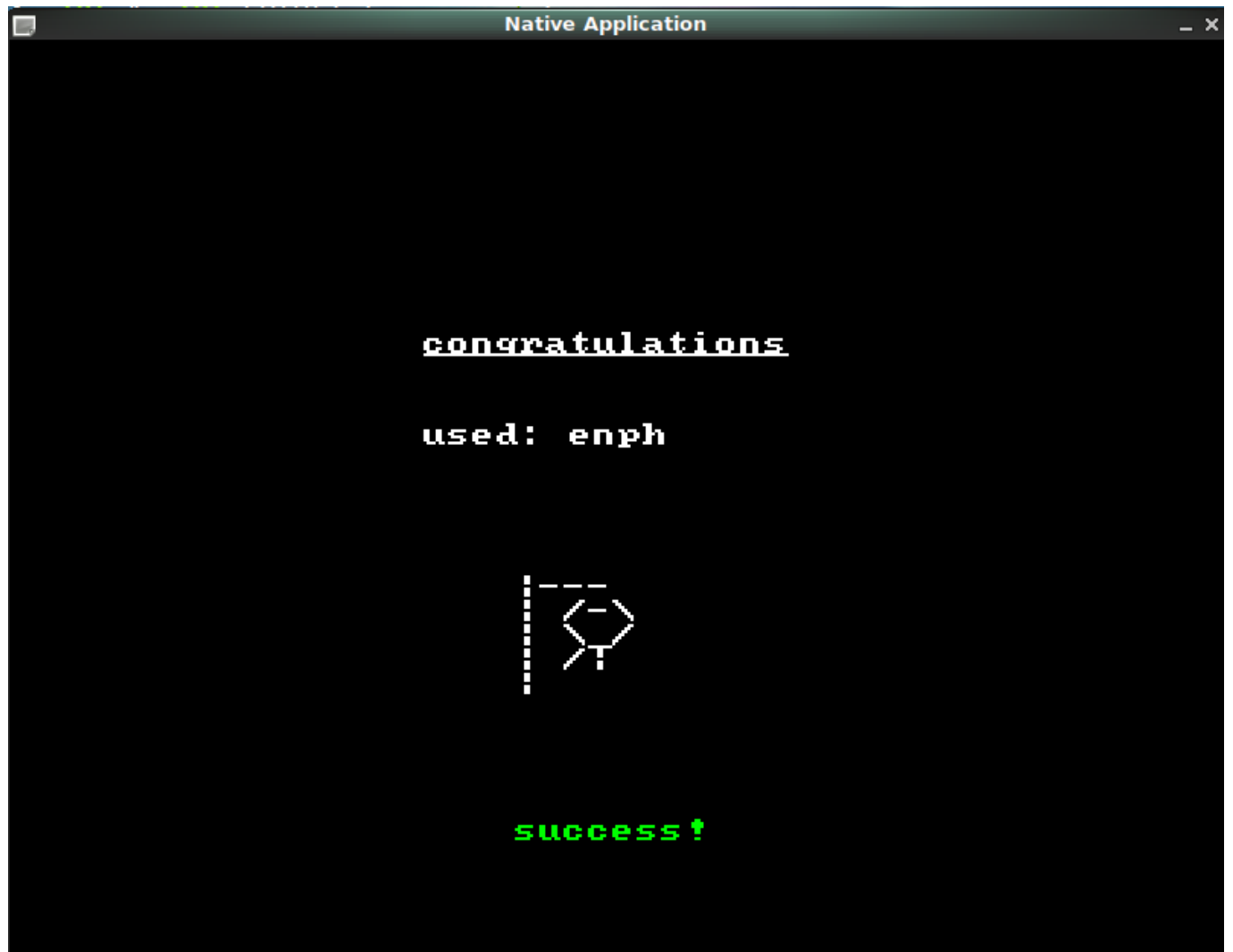
hangman:猜词小游戏

规则：输入字母猜单词，如果当前字母是单词的一部分则该字母会在单词的对应位置显示；如果输入的字母不属于当前单词，则下方的“上吊小人”图像被更新，只有7次猜错的机会，当“上吊小人”图像完成时游戏失败，在图像完成之前猜出当前单词的所有字母则游戏成功。成功或失败之后均会自动进入下一轮。

游戏界面：



成功猜词：



未猜出:



思路

借鉴了am中打字小游戏的c语言实现，上学期数电大实验实现打字小游戏的经验也提供了有益的帮助。

游戏主界面被分为4个部分：gameboard(填词的区域)，usedboard(显示已经用过的不正确的字母)，hangman(“上吊小人”图像区域)，finishboard(游戏结束后显示“success”或“GG”的区域)。游戏初始化时，根据屏幕的宽度与长度，在init_screen()中设置上述4个部分的左上角顶点的坐标。接下来通过reset_game()初始化一局游戏，包括重置相关变量，绘制初始图像等。其中，绘制图像都是通过函数draw_character()和redraw()实现的，而redraw()函数内部调用am提供的draw_rect()和draw_sync()，图像的绘制在此处与底层硬件层相接。

游戏的主循环中，每隔1000 / FPS毫秒检测一次键盘输入，如果检测到键盘按键抬起，则对该按键对应字母进行检查，若与单词中的字母匹配，则更新gameboard，并将剩余字母数left减小；若不匹配，则更新hangman区域内的图像，并将剩余机会数chance减1。若chance或left为0，则游戏结束，在finishboard区域绘制“success!”或“GG”。等待1秒后，调用reset_game()，重新开始游戏。

遇到的问题 and 解决方案

调用klib时提示冲突

原因在于game.c中默认提供的函数与klib中定义的函数原型冲突了。经过权衡后，决定删去提供的默认函数，因为klib中除了默认函数，还有包括stdio、stdlib等其他库的内容，兼容klib无疑是更好的选择。

字符画数组无法初始化

编译时报错，报得五花八门，最后发现是用字符\初始化数组时，其默认被识别为转义符而非原字符造成的，将其改为'\\'解决了问题。如果是一个单独的字符，也许这个问题很容易被发现，然而当它混在字符画中时，事情就变得没那么简单了orz

开始新的游戏之后，屏幕上会残留之前的字符

显然是由于屏幕没有被擦除所致，因此每次开始新的一局游戏之前，首先将gameboard、usedboard、hangman、finishboard四个区域对应的数组清零，并调用redraw()进行一次重绘，这样就会在清屏之后进行新的绘制了。

游戏中多次输入正确字符，会被多次判定有效

逻辑上，当单词中的一个字母被正确输入后，再次输入该字母就不再有任何效果了。然而在代码实现时，每次检测到正确的字母时剩余字母数都会减少，导致最后剩余字母数变为负数。为此，在检测到正确的字母后，将该位置的字符改为'*'，就避免了这一问题。