

打砖块游戏要求

老师要求是用 c++ 写 但是我们除了学了用 cout 外 其他复杂的类啥的都没学。。。

麻烦尽量写的简单一些

运用库 conio2.h, 主要用到 int getch(); void gotoxy(); int kbhit();函数来读取用户的键盘输入 并输出。

游戏界面截屏

长宽分别为 80 和 40 个字节

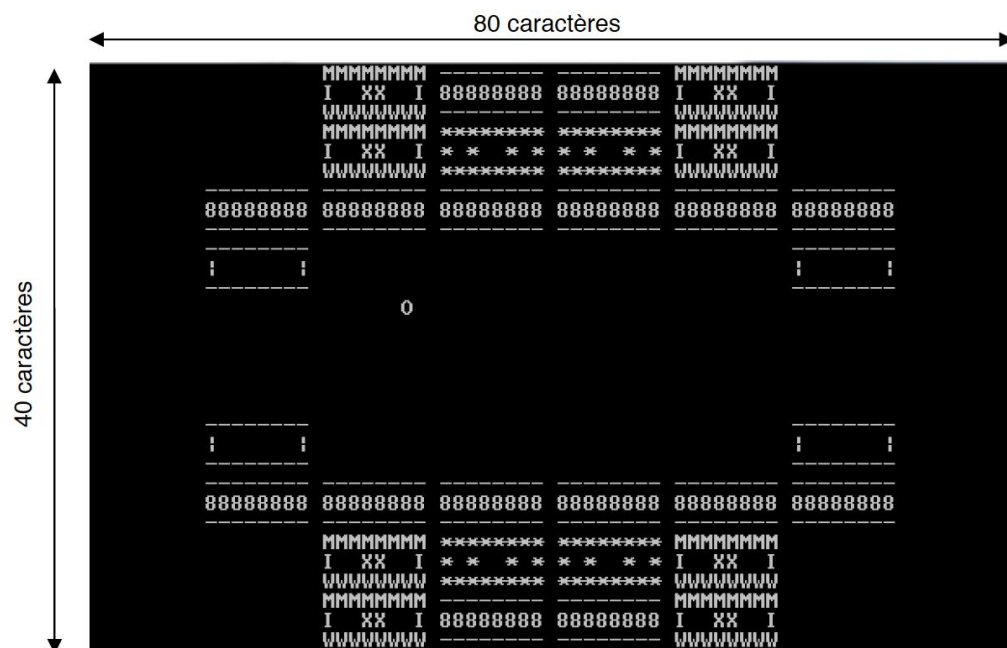


Figure 1 - Capture d'écran possible du jeu

游戏规则：

玩家用两个按键来控制小球让它左右移动（x 轴），小球在 y 轴以预先定义的速度自动运动。当小球碰到水平的上下墙面，小球被挡住；当小球碰到竖直的两面墙面或者是砖块，小球反弹。

砖块有几种不同的类型：

标准型：小球碰到就消失

加强型：碰两次才消失

永不消失型

毁灭型：小球碰到玩家损失一条生命

玩家有一定数量的生命，全部失去则游戏结束。当全部可以消的砖块都消掉后则可以进入下一等级。每次砖块消掉和升级时都会有得分，游戏结束计算总得分。

格式要求：定义的结构体及函数等要另用 .h 文件，不要全在 main 函数里

步骤：（库 conio2）

1. 写 `int keyboard()` 函数，确认用户有输入（控制小球左移右移的指令以及 `esc`），`kbhit()` 扫描输入的 ASCII 码
2. 写主函数，调用上面写的 `keyboard` 函数，检测 `return` 的值是否正确
3. 定义两个结构体 小球的 `Position` 和 `Speed`（用 `typedef` 定义）
4. 定义一个结构体 `Ball`，其中包含了位置和速度信息
5. 写两个函数 `void moveBallX(Ball *aBall)` 和 `void moveBallY(Ball *aBall)`，用来控制小球的横纵坐标（根据速度），如果小球超出了限定的边界，碰到 `x` 轴被挡住，碰到 `y` 轴就反弹
6. 写两个函数 `void displayBallX(Ball *aBall)` 和 `void eraseBallX(Ball *aBall)`，用来输出和清除小球当前的位置。再写一个无限循环用来检测游戏的时间进程。在每轮循环中：
 - 计数器是一个增量，他要用来测量游戏进行的时间，以便于规范球的运动。
 - 引用 `keyboard` 函数。当玩家发出指令时，用 `switch-caes` 来改变游戏中的一些变量。当玩家按 `esc` 键时，跳出循环，游戏结束。
 - 我们需要检测计数器是否会超过一个已知常数（由计算机内部决定）。如果超过该常数，我们重新将计数器置 0，并输出小球的新坐标（别忘记擦除小球的原坐标）
7. 在 `main` 函数中，我们要实现以下功能。
8. 写一个名为 `brique` 的结构体来实现不同等级的砖块。我们需要定义砖块的类型（包括 `VIDE` 型）。我们用砖块的位置和一个二维数组来画出砖块的图像。我们定义两个常量 `BRIQUE-HEIGHT` 和 `BRIQUE-WIDTH` 来定义砖块的长和宽。并用一个 `char` 类型的静态数组来储存砖块的图像。
9. 写一个名为 `bricksTab` 的数组用来存储不同等级的砖块。写一个函数用来初始化这个数组，再写一个函数用来输出砖块的图像。

10. 写一个名为 `bool is colliding` 的函数（`prototype` 型），`bool is colliding (Ball *aBall, Brick *aBrick)` 如果球和砖块发生了碰撞 `return 1`；如果没有 `return 0`.
11. 完成前面的无限循环。当小球碰到砖块的时候，小球会反弹。
12. 有一种砖块类型是是 小球一碰到，游戏就会结束。
13. 7 中计数器的原理是 我们要精确的测出小球两次移动的时间差，这个函数的内容在 `time.h` 中可以找到。