با نام خدا

پروژه درس مبانی کامپیوتر **زمان تحویل: ۱۳۹۲/۱۰/۴**

پیاده سازی بازی Snake استاد درس: دکتر سعید ابریشمی

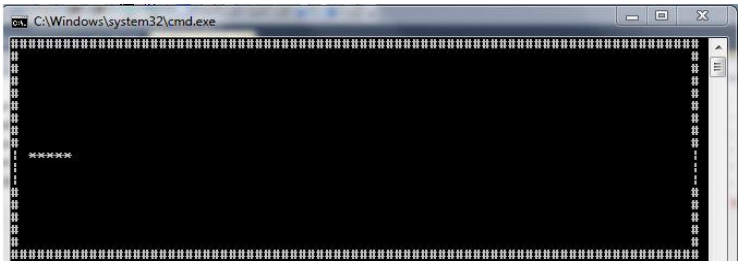
رسم توضیحی ۱نمای بازی در گوشی‌های نوکیا

بازی snake در اواخر ۱۹۷۰ ساخته شد از آن پس جزو بازی‌های محبوب به شمار می‌آید. در سال ۱۹۸۸ این بازی به یکی از بازی‌هایی تبدیل شد که به صورت پیش فرض در گوشی‌های نوکیا نصب می‌شدند. در طول زمان بر محبوبیت این بازی افزوده شد و مخاطبان بیشتری پیدا کرد.

هدف از این پروژه پیاده سازی بازی به صورت متنی بوده و نیازی به پیاده سازی گرافیکی نمی‌باشد.

این بازی شامل چندین نوع شی می‌باشد که در ادامه به بررسی هر یک از آن‌ها می‌پردازیم:

۱- مار: این شی توسط کاربر کنترل می‌شود. نمایش این شی با استفاده از مجموعه‌ای کاراکتری از "*" نمایش داده می‌شود. با فشردن هر یک از چهار جهت اصلی می‌توان جهت حرکت مار را مشخص کرد، این حرکت در جهت مشخص شده ادامه می‌یابد تا زمانی که کاربر جهت را تغییر دهد. طول مار بعد از خوردن هر طعمه یک واحد افزوده می‌شود.



رسم توضیحی ۲پیاده سازی snake در حالت متنی

خوردن هر طعمه ۵۰ امتیاز دارد. بعد از هر ۵۰۰ امتیاز یک جایزه برای مدت ۲۰ ثانیه ظاهر می‌شود که ۲۰۰ امتیاز دارد و طول مار را ۴ واحد کم می‌کند. سرعت مار بعد از هر ۱۰۰۰ امتیاز اضافه می‌شود.

۲- طعمه: این شی با کاراکتر "." نمایش داده می‌شود. طعمه ها باید دائما در صفحه بازی حضور داشته باشند. بعد از خورده شدن هر طعمه توسط مار باید طعمه دیگری به صورت تصادفی در نقشه ایجاد شود.

۳- جایزه: این شی با کاراکتر "@" نمایش داده می‌شود. با گرفتن هر ۵۰۰ امتیاز توسط مار یک جایزه به طور تصادفی در نقشه ظاهر می‌شود و به طور تصادفی شروع به حرکت در نقشه می‌کند این شی دارای ۲۰۰ امتیاز است و ۵ واحد از طول مار کم می‌کند. در صورت خورده نشدن جایزه بعد از ۱ دقیقه جایزه از نقشه حذف می‌شود. زمان باقی‌مانده برای از بین رفتن جایزه نیز در زیر نقشه نمایش داده شود.

۴- دیوار: این شی ۲ نوع است که نوع اول با کاراکتر "#" نمایش داده می‌شود. در صورتی که مار با این نوع دیوار یا خودش برخورد کند بازی به اتمام می‌رسد. نوع دوم با کاراکتر "|" نمایش داده می‌شود. که مار می‌تواند از آن عبور کرده و به طرف دیگر نقشه برود (به طور مثال اگر از راست وارد این نوع دیوار بشود از طرف چپ وارد نقشه می‌شود).

تذکر:

- پیاده سازی پروژه به صورت انفرادی است.
- کپی برداری یا عدم تسلط در ارائه پروژه نمره ۰ را در پی دارد.
- نقشه بازی به صورت آرایه‌ای ۲بعدی به همراه تعدادی تابع مفید برای پیاده سازی در پیوست آمده است.
- رعایت خوانایی کد از جمله دندانه گذاری و کامنت گذاری مهم است.
- پیاده سازی باید به صورت ساخت یافته باشد.
- تا حد امکان از تکرار کدها خودداری نمایید و از توابع استفاده کنید.
- تا حد امکان از تعریف متغییرهای سراسری خودداری نمایید.
- برای بررسی فشرده شدن کلیدی از روی صفحه کلید می‌توان از تابع ()kbhit_ استفاده کرد. این تابع در صورتی که کلیدی روی صفحه کلید فشرده نشده باشد عدد **صفر** و در صورت فشرده شدن کلیدی روی صفحه کلید **عددی غیر صفر** باز می‌گرداند. این تابع درون کتابخانه‌ی conio.h پیاده سازی شده.
- برای محاسبه زمان می‌توان از تابع ()clock که در کتابخانه ctime قرار گرفته استفاده کرد. این تابع زمان سپری شده از شروع برنامه را بر حسب میلی ثانیه به شما باز می‌گرداند.
- برای به دست آوردن اعداد تصادفی می‌توان از تابع ()rand استفاده کرد. این تابع با استفاده از الگوریتمی هر بار عددی بین ۰ تا RAND_MAX بر می‌گرداند این الگوریتم وابسته به زمان اجرای تابع نیست در نتیجه شما اگر ۲بار برنامه را اجرا کنید. هر بار یک سری از اعداد تصادفی را دریافت می‌کنید. برای رفع این مشکل در ابتدای برنامه تابع srand(time(NULL)) را فراخوانی می‌کنیم از این پس در هر بار فراخوانی ()rand ما اعدادی متفاوت دریافت می‌کنیم.
- پیکان‌ها (چپ، راست، بالا و پایین) دو کده هستند. به عنوان مثال، با فشردن کلید جهت بالا، فراخوانی اولین ()getch مقدار ۳۲- و فراخوانی دومین مقدار ۷۲ را باز می‌گرداند.

با آرزوی موفقیت

عرفانی - ابوالفضلی

پیوست ۱:

الف) پیاده‌سازی توابع مورد نیاز

از آنجایی که پیاده‌سازی پروژه در محیط Visual C++ انجام می‌شود، پیاده‌سازی برخی توابع جهت سهولت استفاده از محیط Console لازم است که در زیر به پیاده‌سازی و نحوه عملکرد هر کدام از آن‌ها می‌پردازیم:

۱- کتابخانه‌های مورد نیاز

```
#include <windows.h>
#include <ctime>
```

۲- تابعی جهت جابه جا کردن مکان نما. جهت چاپ یک کاراکتر در مکانی مشخص، قبل از فراخوانی printf، putch یا سایر توابع باید فراخوانی شود.

```
void gotoxy(int x, int y)
{
	HANDLE consoleHandle = GetStdHandle(STD_OUTPUT_HANDLE);
	COORD cursorCoord;
	cursorCoord.X=x;
	cursorCoord.Y=y;
	SetConsoleCursorPosition(consoleHandle, cursorCoord);
}
```

۳- تابعی جهت پاک کردن صفحه نمایش

```
void clrscr()
{
	system(“cls”);
}
```

۴- تابعی جهت تغییر رنگ متن قبل از فراخوانی توابع مورد نیاز جهت چاپ متن، با فراخوانی این تابع و دادن کد رنگ‌های مورد نظر (از ۰ تا ۱۵۰)، رنگ نوشته به دلخواه تغییر خواهد کرد.

```
void setTextColor(int textColor, int backColor=0)

{

	HANDLE consoleHandle = GetStdHandle(STD_OUTPUT_HANDLE);

	int colorAttribute = backColor << 4 | textColor;

	SetConsoleTextAttribute(consoleHandle, colorAttribute);

}
```

۵- با فراخوانی تابع و پاس دادن عددی به واحد میلی ثانیه، کار برنامه به اندازه‌ی دلخواه متوقف می‌شود.

```
void sleep(unsigned int mseconds)

{

	clock_t goal = mseconds + clock();

	while (goal > clock());

}
```

ب) نقشه ۲بعدی بازی

```
const int HEIGHT = 18;

const int WIDTH = 79;

char map[HEIGHT][WIDTH]={

{'#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#','#'},

{'#',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ','#'},

{'#',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ','#'},
```

{'#',' ','
',' ','#'},

{'#',' ','
',' ','#'},

{'#',' ','
',' ','#'},

{'#',' ','
',' ','#'},

{'#',' ','
',' ','#'},

{'#',' ','
',' ','#'},

{'|',' ','*','*','*','*','*',' ','
',' ','|'},

{'|',' ','
',' ','|'},

{'|',' ','
',' ','|'},

{'#',' ','
',' ','#'},

{'#',' ','
',' ','#'},

{'#',' ','
',' ','#'},

{'#',' ','
',' ','#'},

{'#',' ','
',' ','#'},

{'#','
#','#'
,'#','#','#','#','#','#','#','#','#','#','#','#','#','#'}

```
};
```