

۱. مقدمه

۲. کتاب خانه های استفاده شده

۳. الگوریتم پیاده سازی شده

۴. توابع تعریف شده

۱. ۴. INIT()

۲. ۴. DRAWER()

۳. ۴. USER_INPUT(INT USER)

۴. ۴. GAMER(INT USER_VALUE , INT USER)

۵. ۴. INLINE(INT PX , INT PY)

۶. ۴. END(INT C)

سید علی میرفردوس

پروژه میان ترم مبانی کامپیوتر: پیاده سازی بازی کانکت فور با C

استاد نصیحت کن

پاییز ۱۳۹۵

۱. مقدمه

بازی کانکت فور از بازی های استراتژیک معروفی است که در سال ۱۹۷۴ میلادی اختراع شده و امروزه طرفداران زیادی دارد. این بازی دو نفره بوده و حدود ۱-۵ دقیقه طول می کشد. بازی از تعدادی دیسک و یک برد تشکیل شده که معروف ترین آن شامل ۶ ردیف و ۷ ستون می باشد.

هر بازیکن در نوبت خود می تواند دیسک خود را در یکی از ستون ها بیندازد و دیسک در پایین ترین حفره خالی جای می گیرد. اولین بازیکنی که بتواند ۴ دیسک متوالی هم رنگ خود به صورت افقی، عمودی یا قطری ایجاد کند برنده بازی است و چنانچه تمام حفره ها پر شوند و بازیکنی نبرده باشد، بازی مساوی اعلام می شود.

در این پروژه قصد داریم این بازی را با زبان C و به صورت برنامه ی کنسولی (Console Application) پیاده سازی کنیم.

۲. کتاب خانه های استفاده شده

در این پروژه تنها از دو کتابخانه ی زیر استفاده شده است:

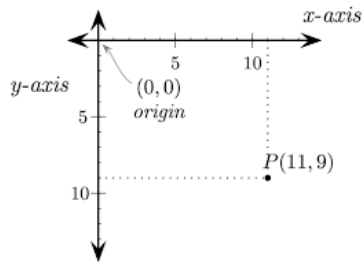
۱. stdio.h

۲. windows.h

۳. الگوریتم پیاده سازی شده

آرایه ای دو بعدی ۷ در ۷ تعریف شده که در هر لحظه وضعیت فعلی صفحه بازی را ذخیره می کند. از آنجایی که صفحه ی بازی ۶ ردیف دارد، ردیف اول اضافی فرض شده که به آن خواهیم پرداخت.

در ابتدای بازی تمام خانه های آرایه به غیر از ردیف اول که ردیف فرضی است مقدار صفر می گیرند. ردیف اول نیز مقدار ۶ را می گیرد.



هر حفره روی صفحه بازی با مختصات یکی از خانه های آرایه مشخص می شود. باید توجه کرد که صفحه مختصاتی که فرض شده، به صورت صفحه مختصات گرافیک کامپیوتری است که نقطه بالا چپ مبدا است؛ شکل رو به رو این موضوع را به خوبی تشریح می کند.

کاربر اول با عدد ۱ و کاربر دوم با عدد ۲ مشخص می شوند. در هر نوبت، برنامه با گرفتن شماره ستون مورد نظر کاربر، پایین ترین حفره ی خالی یعنی پایین ترین خانه آرایه در آن ستون که مقدار آن صفر است را برابر با عدد کاربر قرار می دهد.

با قرار گرفتن عدد کاربر در محل مربوطه، تابع چک کننده وضعیت برد صدا زده می شود. این تابع، ابتدا امکان سمت های چپ، راست و پایین را چک می کند. از آنجایی که برای برنده شدن باید چهار مقدار برابر، کنار هم باشند کافی است سه مقدار برابر با مقدار مورد نظر موجود باشند پس:

- اگر شماره ستون + ۳ کوچک تر مساوی تعداد ستون ها باشد سمت راست ممکن است.
- اگر شماره ستون + ۳ بزرگ تر مساوی صفر باشد سمت چپ ممکن است.
- اگر شماره سطر + ۳ + ۱ کوچک تر مساوی تعداد سطرها باشد سمت پایین ممکن است.

اولین سمتی که امکان پذیر باشد، ابتدا چک می شود.

برای سمت چپ و راست اگر ممکن بود، راست پایین قطری (چک می شود که پایین ممکن است یا نه) و راست بالا قطری (چک می شود که بالا ممکن است یا نه) و همچنین برای چپ، بدین ترتیب که آیا مقدار هر سه خانه از هر کدام حالت ها برابر عدد مورد نظر هست یا نه، چک می شوند. برای حالت پایین نیز این اتفاق می افتد.

نحوه بدست آوردن مختصات خانه ها بسیار ساده است. برای مثال حالت قطری توضیح داده می شود:



همان طور که در شکل رو به رو مشخص است، اگر در خانه مورد نظر ۱ قرار گرفته باشد. ۳ خانه قطری راست بالا با مختصات های زیر مشخص می شوند:

$$(x+1, y+1) \quad (a)$$

$$(x+2, y+2) \quad (b)$$

$$(x+3, y+3) \quad (c)$$

برای سایر حالت ها هم به راحتی از روی جدول محاسبه می شود.

ردیف اول، شماره پایین ترین حفره خالی آن ستون (پایین ترین صفر) را ذخیره می کند. در صورتی که چنین خانه ای وجود نداشته باشد یعنی آن ستون پر شده باشد، مقدار ۱- را خواهد گرفت. متغیری تعریف شده که ابتدا برابر با تعداد ستون ها است و با پر شدن هر ستون، یک واحد کم می شود و وقتی به صفر برسد یعنی دیگر خانه ای خالی نمانده و بازیکنی برنده نشده است؛ در نتیجه بازی را مساوی اعلام می کند.

۴. توابع تعریف شده

۱.۴ init()

این تابع در ابتدای شروع برنامه صدا زده می شود و مسئولیت صفر کردن درایه های آرایه board که به صورت static و global تعریف شده را دارد.

در ضمن، درایه های ردیف اول این آرایه، مقدار ۶ را می گیرند.

این تابع هیچ ورودی و خروجی ندارد.

۲.۴ drawer()

این تابع، مسئولیت چاپ کردن آرایه board در صفحه کنسول را دارد. در ابتدا صفحه کنسول را پاک می کند و بعد از چاپ کردن مرتب آرایه، خطی نقطه چین می کشد.

این تابع نیز هیچ ورودی و خروجی ندارد و در جاهای مختلفی از برنامه صدا زده شده است.

۳.۴ user_input(int user)

این تابع مسئولیت گرفتن ورودی کاربر و چاپ متن مناسب برای ورودی را دارد. در ابتدا متنی را چاپ می کند و نوبت کاربر را اعلام می کند. متغیر ورودی user فقط در چاپ کردن این متن به کار می رود.

چنانچه کاربر عددی خارج از محدوده شماره ستون ها اعلام کند، ابتدا تابع drawer صدا زده می شود تا متن نوبت قبلی پاک شود؛ سپس متن مناسب نمایش داده شده و مجدد همین تابع صدا زده می شود تا کاربر عدد مناسب را وارد کند.

با توجه به این که آرایه از صفر شروع می شود، از مقدار وارد شده یک واحد کم شده و سپس بازگردانده می شود.

۴.۴ gamer(int user_value , int user)

این تابع مسئولیت تغییر وضعیت آرایه board بر اساس مقدار وارد شده جدید را دارد.

مقدار آرایه در خانه ای با ستون user_value و ردیف مقدار خانه اول آن ستون - که آخرین حفره خالی را نشان می دهد - برابر با user می شود که ۱ یا ۲ است. سپس مقدار خانه اول یک واحد کاهش می یابد و چنانچه به صفر برسد آن را ۱- می کند یعنی که آن ستون پر شده و از تعداد ستون های مجاز که هنوز پر نشده اند یک واحد کم می کند.

چنانچه خانه اول آن ستون ۱- باشد مجدداً از کاربر ورودی می خواهد و تابع را صدا می زند. این تابع مقداری برنمی گرداند.

۵.۴ InLine(int px , int py)

این تابع مسئولیت چک کردن شرط برنده شدن بازی را دارد.

با فرض این که خانه ای که کاربر در آن دیسک جدید انداخته (px,py) است و با الگوریتم توضیح داده شده در بالا شرط را چک می کند.

چنانچه شرط برقرار باشد مقدار ۱ و در غیر این صورت ۰ بر می گرداند.

در ابتدای تابع نیز چک شده که خانه با مختصات داده شده حتما دارای دیسک باشد.

۶.۴ end(int c)

این تابع اعلام نتیجه نهایی بازی را بر عهده دارد.

با توجه به متغیر ورودی، جمله مورد نظر را چاپ می کند. این متغیر طی اجرای تابع main و صدا زده شدن تابع های مختلف مقدار دهی می شود.

این تابع مقداری برنمی گرداند.