

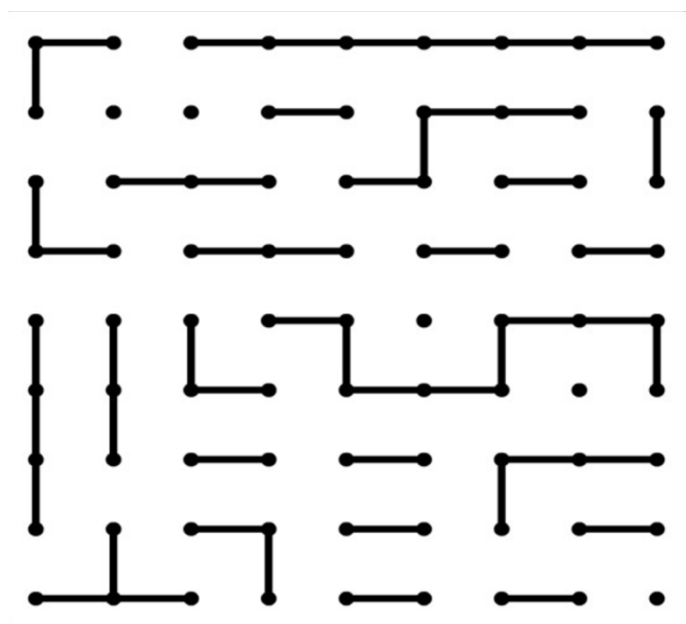


نقطه بازی!

توضیح بازی

شاید خیلی از ما زمانی که کوچکتر بودیم نام این بازی به گوشمان خورده باشد و با دوستان و خانواده‌ی خود این بازی ساده و در عین حال جذاب را بازی کرده باشیم. شما قرار است این بازی را در دو بخش پیاده‌سازی کنید.

بهتر است اول شرایط و قوانین نقطه بازی را مرور کنیم.



در این بازی دو بازیکن روی یک شبکه‌ی مستطیلی از نقاط بازی می‌کنند. هر بازیکن در نوبت خود دو نقطه‌ی مجاور را با خطی عمودی یا افقی به هم متصل می‌کند. بازیکنی که چهارمین ضلع مربعی را رسم می‌کند، صاحب این مربع خواهد بود و باید خط دیگری به عنوان جایزه‌ی اجباری رسم کند. بازی با پر شدن صفحه تمام می‌شود و برنده کسی است که صاحب خانه‌های بیش‌تری باشد. بازیکن‌ها مجبور به کامل کردن یک خانه - وقتی که امکان این کار را داشته باشند - نیستند، مگر آنکه حرکت دیگری نداشته باشند.

توضیح پروژه

بخش اول :

شما باید در بخش اول فلوچارت مربوط به روند و مراحل بازی را رسم کنید. (برای رسم فلوچارت می توانید از توضیحات بخش بعدی پروژه کمک بگیرید.)

بخش دوم :

طراحی بالا به پایین : طراحی بالا به پایین یک روند است که از یک مفهوم سطح بالا به سمت سطح پایین تر حرکت می کند. مفهوم طرح سطح بالا، بزرگتر، گسترده تر و عمومی تر است. طرح سطح پایین کوچکتر، خاص منظره تر و دقیق تر است.

طراحی بالا به پایین در هر رفتار انسانی رخ می دهد و مربوط به سیستم درک و حافظه در مغز انسان می باشد. برای مثال یک نقاش را در نظر بگیرید که برای به وجود آوردن آنچه که در ذهنش دارد ابتدا یک طرح کلی می کشد (سطح بالا) و در نهایت با اضافه کردن نکات و ریزه کاری ها در بخش های مختلف طرح آن را کامل می کند. طراحی بالا به پایین در برنامه نویسی کامپیوتر نیز همین گونه می باشد. با یک تعریف کلی از صورت مسئله شروع می شود و به نیازهایی که باید توسط آن برنامه رفع شود می پردازد. این طرح سطح بالا به صورت مکرر شکسته می شود و به نیازها و طرح های سطح پایین تر تبدیل می شود تا به اندازه کافی ساده برای پیاده سازی در یک زبان برنامه نویسی باشد.

شما باید در این بخش طراحی بالا به پایین خود و در واقع شکستن مسئله به مسائل کوچکتر را تحویل دهید. قالب این طراحی بر عهده خودتان می باشد و باید به گونه ای باشد که شیوهی تقسیم بندی مسئله تان واضح باشد.

در رابطه با طراحی بالا به پایین در جلسه ی توجیهی پروژه توضیحات بیشتر داده خواهد می شود.

بخش سوم :

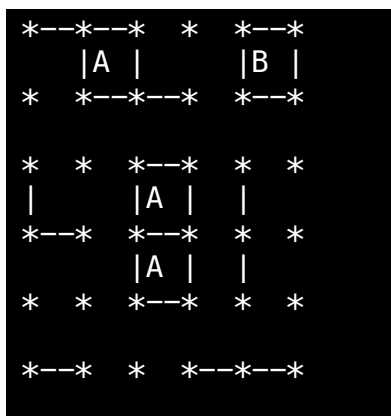
در این بخش شما باید نقطه بازی را بدون نمایش گرافیکی پیاده سازی کنید.

بازی با نوبت بازیکن اول که A نامیده می شود، شروع می شود. بازیکن با دادن مختصات یک خط محل قرارگیری خط را مشخص می کند و در صورتی که قرار دادن خط منجر به کسب امتیاز شد دوباره همان بازیکن مختصات دیگری را انتخاب می کند. در غیر این صورت نوبت بازی عوض شده و به همین منوال بازی ادامه پیدا میکند تا تمام محل های قرارگیری خطها پر شوند. در پایان برنده ی بازی با نمایش پیامی مناسب مشخص می شود.

در هر مرحله از بازی روند بازی به شکل زیر می باشد.

۱- نمایش نقشه ی بازی : در هر مرحله نقشه ی بازی که در ابتدای بازی از ۶ ردیف ۶ نقطه ای تشکیل شده است باید نمایش داده شود. برای نمایش نقشه شما باید به جای نقطه از *، به جای خط های افقی از -- و به جای خط های عمودی از | استفاده کنید.

همینطور در صورتی که مربعی در بازی ایجاد شود شما باید صاحب مربع را با حرف A یا B مشخص کنید. نمونه‌ای از نقشه‌ای که باید نشان داده شود در زیر آمده است.



ورودی خروجی ۱ : نمایش نقشه بازی

۲- گرفتن مختصات : در هر مرحله از بازی با توجه به نوبت بازی شما باید مختصات خطی که بازیکن قصد انتخاب آن را دارد به عنوان ورودی بگیرید. این با مختصات را با سه عدد نشان می‌دهیم.

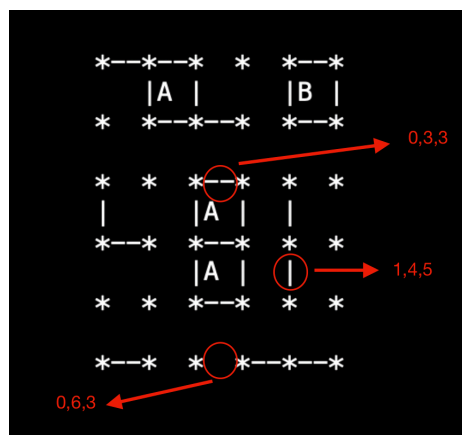
turn : Player B. Please enter coordinates :
1 1 1

عدد اول : عدد اول ورودی، افقی یا عمودی بودن خط را نشان می‌دهد. ۱ به معنای عمودی بودن خط و ۰ به معنای افقی بودن خط می‌باشد.

عدد دوم : عدد دوم ورودی، شماره‌ی سطر آن خط از بین خطوط افقی یا عمودی را مشخص می‌کند. بنابراین واضح است که این عدد برای خطوط افقی می‌تواند از ۱ تا ۶ و برای خطوط عمودی می‌تواند بین ۱ تا ۵ باشد.

عدد سوم : عدد سوم ورودی، شماره‌ی ستون آن خط از بین خطوط افقی یا عمودی را مشخص می‌کند. بنابراین این عدد برای خطوط افقی می‌تواند از ۱ تا ۵ و برای خطوط عمودی می‌تواند بین ۱ تا ۶ باشد.

برای نمونه مختصات چند تا از خطوط نقشه‌ی بالا مشخص شده است.



در قسمت ورودی برنامه شما در صورتی که مختصات صحیحی وارد نشد باید این موضوع با نمایش پیامی مناسب اطلاع داده شود و دوباره از بازیکن مختصات را به عنوان ورودی بگیرد. همچنین اگر بازیکن مختصاتی که قبلاً در آن خطی قرار داده شده را وارد کرد، باید با نمایش پیامی مناسب اطلاع داده شود و دوباره از او مختصات را به عنوان ورودی بگیرد.

بعد از گرفتن مختصات باید خطی در این مختصات قرار بگیرد.

- در صورتی که مربعی ایجاد نشد نوبت به بازیکن بعدی داده می شود و دوباره همین مراحل از مرحله ۱ اجرا شوند.
- در صورتی که مربعی ایجاد شد مربع ایجاد شده به اسم آن بازیکن علامت می خورد و دوباره همین مراحل بدون تغییر نوبت از مرحله ۱ اجرا می شوند.
- در صورتی که تمام جاهای خالی پر شده بود بازی تمام شده و برنامه به مرحله ۳ که مرحله نمایش نتایج می باشد می رود.

۳- نمایش نتایج : در پایان بازی برنامه شما باید امتیاز هر بازیکن و برنده ی بازی را نمایش دهد.

Player A : 15

Player B : 10

Player A won the game!

ورودی خروجی ۲ : خروجی نمونه برای اعلام نتایج نهایی

چند نکته و راهنمایی

- بهتر است هر کدام از مراحل مختلف بازی را در تابع‌های جداگانه بنویسید و از آن‌ها استفاده کنید. (برای مثال می‌توانید یک تابع برای نمایش نقشه بنویسید و در مرحله‌ی نمایش نقشه تنها آن را صدا بزنید. و یا اینکه می‌توانید یک تابع برای چک کردن تمام شدن بازی بنویسید و در زمان لازم از آن استفاده کنید).
 - بهتر است که از یک حلقه‌ی کنترلی کلی برای تکرار مراحل کلی بازی استفاده کنید.
 - بهتر است برای ذخیره‌ی وضعیت خطوط افقی و خطوط عمودی از دو آرایه‌ی دوبعدی کمک بگیرید.
 - می‌توانید برای وضعیت هر کدام از مربع‌ها از یک آرایه‌ی دوبعدی کمک بگیرید.
 - برای رسم فلوچارت می‌توانید از نرم‌افزار Microsoft Visio و سایت‌های مربوطه استفاده کنید.
- توجه داشته باشید که شما باید در این پروژه توابع خود را در فایل‌های جداگانه با هدر فایل‌های مخصوص خود بنویسید. شیوه‌ی تقسیم‌بندی توابع به خودتان مربوط است و هر روش قابل قبولی پذیرفتنی است.

نحوه‌ی نمره‌دهی :

عنوان	نمره
صحیح بودن فلوچارت	۱۰
تقسیم درست مسئله به مسائل کوچکتر	۱۰
نام‌گذاری مناسب و اصولی متغیرها	۵
استفاده از تمام ورودی‌های تابع در آن	۵
نام‌گذاری مناسب توابع	۵
عدم وجود قطعه کد تکراری	۱۰
استفاده‌ی درست از هدر فایل‌ها برای تقسیم بندی توابع	۵
دریافت صحیح ورودی	۱۰
استفاده از آرایه‌ی دوبعدی	۵
رسیدگی به خطاها	۵
چاپ صحیح نقشه‌ی بازی	۵
تغییر درست نوبت‌ها با توجه به شرایط بازی و خطاها	۵
آپدیت درست بازی	۵
کامنت‌گذاری مناسب در هر جایی که نیاز به مستندسازی دارد	۵
تست و اجرای برنامه	۱۰

- در رابطه با اینکه منظور از نام‌گذاری درست و صحیح چه می‌باشد در فروم درس نکاتی ذکر خواهد شد.

نحوه‌ی تحویل پروژه :

شما باید فایل‌های .h ، .c و همچنین فلوچارت و طراحی بالا به پایین خود را در قالب pdf در قسمت در نظر گرفته شده با عنوان CA2[UploadHere] در صفحه‌ی درس در سایت CECM آپلود نمایید. مهلت آپلود تا ساعت ۲۳:۵۵ جمعه ۱۸ ام آبان است . تأخیر در سه روز نخست به ازای هر روز ۱۰ درصد و در سه روز دوم به ازای هر روز ۱۵ درصد جریمه خواهد داشت. پس از این شش روز، به هیچ وجه نمره‌ای در نظر گرفته نخواهد شد. همچنین توجه کنید که با توجه به تأخیر در نظر گرفته شده به هیچ وجه امکان تمدید پروژه‌ها وجود ندارد.

فایل‌های خود را در یک فایل زیپ با فرمت “zip” و با نام CA2-[SID].zip قرار دهید که SID همان شماره‌ی دانشجویی شماست. برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۷۰۰۰ باشد باید نام فایل خود را CA2-810197000.zip قرار دهید .

پروژه‌ها برای یادگیری برنامه نویسی و مباحث مطرح شده در کلاس طراحی میشوند و انجام آنها به صورت انفرادی خواهد بود . همچنین، در صورت شباهت میان دو پروژه (که به وسیله ی نرم افزارهای مربوطه چک میشود) برای هر دو نفر نمره ی صفر در نظر گرفته خواهد شد .

در صورت وجود هرگونه سوال میتوانید پرسشهای خود را در فروم درس (در بخش مربوط به این پروژه) مطرح نمایید یا به icsp97.ca@gmail.com ایمیل بزنید.

موفق و سربلند باشید