# مینی پروژه درس برنامه نویسی پیشرفته

**استاد درس:** دکتر منصوری

بهار ۱۴۰۰

---

## شرایط انجام مینی پروژه

پروژه در گروه‌های دو نفره انجام می‌پذیرد. با توجه به محدودیت زمانی پروژه، دانشجویان می‌توانند برای پیدا کردن هم‌تیمی، به دستیاران آموزشی پیام دهند و آن‌ها را مطلع کنند.

دانشجویان باید برای انجام پروژه از گیت استفاده کنند و برای مشخص بودن میزان مشارکت هر فرد، کامنت‌ها و لاگ‌های گیت مورد بازبینی قرار می‌گیرند. برای انجام این کار، دانشجویان باید یک ریپوزیتوری به صورت Private در سایت گیت‌هاب ایجاد کرده و این ایمیل را بعنوان یکی از اعضا در ریپوزیتوری ساخته شده اضافه کنید.

در صورت نیاز می‌توانید در طول انجام پروژه از Branchهای متفاوت استفاده کنید و در قسمت‌های مختلف عملیات Merge انجام دهید.

نوشتن Readme مناسب و کامل، می‌تواند نمره‌ی اضافی داشته باشد.

---

## کلیات پروژه

در این بخش به معرفی بازی می‌پردازیم .

در این مینی پروژه قصد داریم یک سیستم مدیریت بازی‌های دوز راه‌اندازی کنیم. کاربران می‌توانند در سیستم از بین کاربران آنلاین (لیستی که سرور هنگام درخواست بازی برای آن ها میفرستد)، حریفی انتخاب کرده و با توافق طرف مقابل، بازی دوزی با آن شخص انجام دهند. در این بازی، هر بازیکن یک مهره و یا یک شکل مخصوص به خود(برای مثال کارکتر"*") دارد که می‌تواند در نوبت خود، آن را در هر یک از خانه‌های خالی قرار دهد. در این بازی هدف این است که به نسبت ابعاد زمین، تعداد مشخصی مهره هم‌شکل در یک ردیف قرار داده شوند. ردیف‌ها در این بازی می‌توانند سطری، ستونی و قطری باشند.

---

## محیط و گرافیک بازی (رابط کاربری)

با توجه به ماهیت مینی پروژه، هدف پیاده‌سازی محیط گرافیکی نیست و می‌توانید منو و محیط اولیه بازی و همچنین محیط مسابقه‌های دوز را درون ترمینال و به صورت کارکتری و بصورت یک ماتریس دو بعدی پیاده‌سازی نمایید.

همچنین میتوانید از کتابخانه های موجود در CPP برای چاپ رنگی در ترمینال استفاده نمایید.

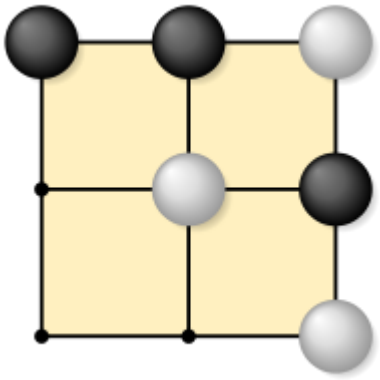
---

## زمین بازی

زمین بازی شامل:

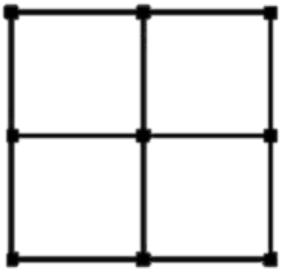
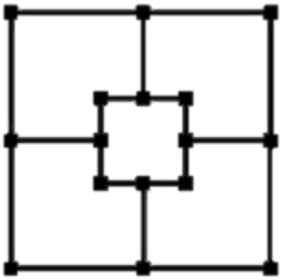
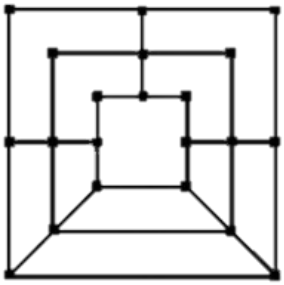
- نام زمین
- اندازه زمین
- تعداد خانه های پشت سرهم لازم برای برد
- فرمت کلی زمین در قالب یک ماتریس دو بعدی

همانگونه که احتمالا آشنایی دارید، ساده‌ترین بازی دوز، دوز ۳تایی نام دارد که داخل یک زمین مربعی ۳ در ۳ انجام می‌پزیرد و در شکل زیر نمونه‌ای از آن آورده شده است و همچنین در این لینک می‌توانید این بازی را تجربه کنید:



اما در این سیستم ابعاد زمین متغیر است و هر کاربر در هنگام ساخت بازی می‌تواند از بین زمین‌هایی که بطور پیش فرض در بازی طراحی شده‌اند یک زمین را انتخاب نماید. هر بازی از دو بازیکن و یک زمین بازی تشکیل شده است.

دانشجویان باید حداقل ۳ زمین که در زیر نشان داده شده‌اند را در لیست زمین‌های انتخابی قرار دهند: (قرار دادن قابلیت ساخت زمین دلخواه برای کاربر، می‌تواند امتیاز اضافی داشته باشد)

---

## بازیکن‌ها

هر بازیکن دارای موارد زیر است:

- آی‌دی مخصوص بازیکن
- آنلاین یا آفلاین بودن بازیکن

---

## روند اجرای برنامه

پس از راه‌اندازی سرور، کاربران می‌توانند در ترمینال‌های جداگانه به سرور متصل شده و از بین افراد آنلاین در سرور، بازیکنی را انتخاب کنند. پس از انتخاب بازیکن مورد نظر، کاربر باید نام زمین مورد نظر را انتخاب کند تا سرور بتواند مسابقه مورد نظر کاربر را بسازد. سپس سرور وظیفه دارد ماتریس مربوط به زمین بازی را برای هر دو کاربر درون بازی ارسال کند. در این حین سرور باید منتظر درخواست‌های بقیه کاربران باشد و در صورت درخواست برای بازی‌های دیگر، سرور بتواند به آن درخواست‌ها پاسخ دهد.

در هر مرحله از بازی، کاربر خانه مورد نظر خود را به سرور اعلام می‌کند(برای مثال درصورت شماره بندی بودن خانه ها شماره مورد نظر را اعلام کند) و در صورتی که بازیکن مجاز به حرکت باشد (بازیکن در نوبت خود اقدام به فرستادن داده کرده باشد) و همچنین خانه انتخاب شده معتبر باشد، سرور حرکت بازیکن را به رقیبش ارسال کند تا هر دو بازیکن صفحه ی بازی خود را بتوانند به روز رسانی کنند. سپس نوبت را عوض کرده و بازیکن حریف را از نوبتش مطلع می‌کند.

اگر کاربری به صورت غیر مجاز داده‌ای بفرستد، سرور باید با پیغام مناسب، بازیکن را از وضعیت پیش آمده مطلع کند و همچنین تغییرات غیر مجاز را اعمال نکند.

سرور باید بتواند تا حداقل ۱۰ کاربر و ۵ مسابقه همزمان را مدیریت کند.

---

## مدیریت خطاها

در قسمت‌های مختلف پروژه در صورت پیش آمدن مشکل و یا انجام حرکات غیر مجاز، کاربران باید توسط سرور از مشکل و یا وضعیت پیش آمده با پیغامی مناسب مطلع شوند.

---

## شبکه

معماری فرض شده برای ارتباط کاربران به صورت کلاینت–سرور است و دانشجویان نیز باید همین معماری را برای ارتباطات لازم، پیاده‌سازی کنند. در این معماری یک کامپیوتر نقش کارگزار که همان سرور است را برعهده می‌گیرد و سایر کامپیوترها با استفاده از آی‌پی و پورت سرور به آن متصل می‌شوند و سرور یک بازیکن برای هر کدام از آن‌ها می‌سازد. در نتیجه شما در این فاز باید علاوه‌بر بخش مربوط به کلاینت‌ها، بخش سرور را نیز آماده کرده و برخی وظایف بازی را به سرور منتقل کنید.

همانطور که در کلاس خوانده‌اید، ارتباط کلاینت‌ها و سرور از طریق سوکت برقرار خواهد شد که این ارتباط از نوع TCP است. برای اتصال کافیست سرور روی یک پورت خاص گوش کند تا کلاینت‌ها بتوانند با اتصال به آن، به سرور متصل شوند.

**لطفا به موارد زیر دقت فرمایید:**

- در طراحی توابع هر کلاس به حفظ ویژگی encapsulation و data hiding در آن کلاس توجه داشته باشید.
- طراحی شما باید تا حد امکان بهینه باشد. (تلاش شود در طراحی کلاس‌ها حداقل کدنویسی لازم انجام شود و اطلاعات کلاس‌ها باهم همپوشانی نداشته باشند.)
- طراحی اصولی و درست کلاس‌ها، خوانایی و reusable بودن کد شما اهمیت بسیاری دارد.

**نکات مهم:**

- این مینی‌پروژه بصورت گروه‌های دونفره انجام می‌شود و تیم‌های خود را از طریق این ایمیل و با عنوان "گروه‌بندی درس AP" اطلاع رسانی کنید.
- استفاده از Git بصورت مستمر در طول انجام مینی‌پروژه الزامی است و توجه کنید که برای commitهای خود پیام‌های معنا داری بنویسید. (برای نحوه‌ی نوشتن یک commit message خوب به این لینک می‌توانید مراجعه کنید.)
- در صورت وجود هرگونه سوال می‌تواند با دستیاران درس ارتباط برقرار کنید و یا از طریق این ایمیل سوال خود را مطرح کنید.
- موعد تحویل پروژه، **۱۹ فروردین ماه** می‌باشد. زمان ارائه آنلاین هر تیم متعاقبا اعلام خواهد شد.

با آرزوی سلامتی و موفقیت :)