

به نام خدا

دانشجویان گرامی پروژه ی پایانترم دو پروژه ی جدا از هم می باشند که هر دانشجو می بایست هر دو پروژه را پیاده سازی کند.

ارائه ی پروژه ها به صورت گروهی قابل قبول نیست .

آخرین مهلت برای آپلود پروژه شب قبل امتحان است و تحویل پروژه به صورت حضوری بعد از امتحان می باشد.

توضیحات مربوط به هر پروژه در صفحات بعد موجود است.

موفق و مؤید باشید

پروژه اول

بازی Minesweeper

بازی minesweeper یا مین روب یک بازی تک نفره است. این بازی به گونه ایست که در یک صفحه ی شطرنجی با ابعادی انتخابی تعدادی مین وجود دارد که کاربر فقط در صورتی برنده ی بازی می شود که مکان مین ها را تشخیص دهد و روی آن ها کلیک نکند تا زمانیکه تمام خانه های صفحه ی شطرنجی بجز خانه هایی که مین ها در آن قرار دارند رو شده باشند. اگر کاربر بر روی هر کدام از خانه هایی که مین ها قرار دارد کلیک کند، تمام مین ها رو می شوند و کاربر بازی را می بازد.

قواعد بازی:

- (۱) تعداد معینی مین به صورت تصادفی در صفحه ای $n \times n$ کاشته می شود. کاربر موظف است بدون گذر از خانه های دارای مین، تمامی خانه های فاقد مین را یافته و مسیر را باز کند.
- (۲) اگر کاربر احتمال دهد که خانه $i \times j$ مین است می تواند آنرا خنثی کند. اگر خانه ای را به اشتباه خنثی کرده باشد می توان آنرا بازگرداند.
- (۳) اگر کاربر روی خانه $i \times j$ که حاوی مین است کلیک کند می بازد.
- (۴) اگر جز خانه های حاوی مین هیچ خانه باز نشده ای وجود نداشته باشد کاربر برنده شده است.
- (۵) اعدادی که در طول بازی در خانه های مختلف ظاهر می شوند به معنی تعداد مینهایی هست که در 8 خانه همسایه این خانه وجود دارد. مثلاً ظاهر شدن 4 یعنی در همسایگی این خانه 4 خانه مین وجود دارد.
- (۶) اگر خانه ای حاوی مقدار صفر باز شود همه همسایه های آن نیز خود بخود خنثی می شوند (زیرا خالی بودن آنها از مین بدیهی است).

بارم بندی قسمت های مختلف برنامه:

- ۱- انجام دادن بازی و اعمال قوانین آن بر روی صفحه ۷۰/نمره
- ۲- نشان دادن یک ثانیه شمار برای بازی ۱۰/نمره
- ۳- گزینه **option** داخل منوی فایل که امکان تغییر اندازه به چهار صورت : ۲۰/نمره
راحت (**beginner**) <-- صفحه ۹*۹ به همراه ۱۰ امین
متوسط (**intermediate**) <-- صفحه ۱۶*۱۶ به همراه ۴۰ امین
سخت (**advanced**) <-- صفحه ۱۶*۳۰ به همراه ۹۹ امین
دلخواه (**custom**) <-- تنظیم تعداد سطر و ستون و تعداد مین ها به صورت دستی
توجه: تعداد مین ها نباید از تعداد کل خانه های صفحه بیشتر باشد.
- ۴- گزینه ی **change the color** داخل منوی **edit** برای تغییر رنگ های بازی با توجه به امکانات از ۵ تا ۱۵/نمره
- ۵- فعال کردن **short key** یا کلید میانبر برای کل گزینه های منوها ۱۵/نمره
- ۶- استفاده از **mousemotionlistener** برای تغییر رنگ خانه ها هنگام گذر ماوس از روی آن ها ۱۰/نمره
- ۷ - استفاده از آیکن به جای کاراکتر برای مین ها ۱۰/نمره
- ۸ - پیاده سازی با استفاده از **javafx** ۲۵/نمره
- ۹ - نشان دادن پیغام مناسب برنده یا بازنده شدن هنگام اتمام بازی ۱۰/نمره
- ۱۰- گزینه **new** داخل منوی فایل ۵/نمره
- ۱۱ - گزینه **save** داخل منوی فایل ۱۰/نمره
- ۱۲ - گزینه **save as** داخل منوی فایل ۵/نمره
- ۱۳ - گزینه **open** داخل منوی فایل ۱۰/نمره

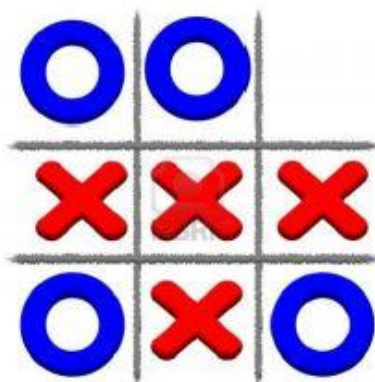
۱۴-گزینه **exit** داخل منوی فایل ۵/نمره

پروژه دوم

بازی دوز (TicTacToe)

این بازی یک بازی دو نفره است که در مستند جاری قالب کد برای حالت تک نفره (نفر دوم کامپیوتر) توضیح داده شده است.

این بازی بر روی یک صفحه سه در سه انجام می شود.



هر بازیکن یکی از نویسه های O یا X را انتخاب می کند. هر بازیکنی که اول بتواند یک سطر یا ستون یا قطر را با نویسه^۱ مربوط به خود پر کند، برنده خواهد بود.

بازی با استفاده از محیط خط فرمان^۲ اجرا خواهد شد. کاربر اول نویسه ای که باید با آن بازی کند را مشخص می کند. سپس در هر نوبت بازی خود مختصات نقطه ای که می خواد بازی کند را مشخص می کند. مختصات نقطه ها از صفر شروع شده و از سمت چپ بالا به راست و پایین افزایش می یابد.

0,0	0,1	0,2
1,0	1,1	1,2
2,0	2,1	2,2

^۱ Character
^۲ Command Prompt

نحوه پیاده‌سازی

مدل داده‌ای که برای این مسئله مناسب به نظر می‌رسد یک آرایه ۳ در ۳ برای نگهداری نویسه‌های فضای بازی است. برای انجام بازی بایستی شکست درستی از الگوریتم صورت پذیرد.

ردیف	نام متد	وظایف
<u>۱</u>	Main	شروع اجرای برنامه و اجرای دستورات مختلف بازی
<u>۲</u>	init	مقداردهی اولیه به متغیرهای مدل داده
<u>۳</u>	getOpponentSymbol	دریافت نویسه‌ای که کاربر می‌خواهد با آن بازی کند.
<u>۴</u>	isGameFinished	بررسی اینکه آیا بازی تمام شده است یا خیر
<u>۵</u>	wholsTheWinner	مشخص کردن برنده بازی پس از اتمام با بازگرداندن مقدار عددی: ۱= برنده شدن کامپیوتر ۲= برنامه شدن کاربر ۳= مساوی
<u>۶</u>	displayGameSpace	نمایش فضای بازی به کاربر
<u>۷</u>	getOpponentMove	دریافت حرکت حریف
<u>۸</u>	playComputerTurn	انجام نوبت بازی توسط کامپیوتر

توسعه بازی

مرحله اول توسعه بازی می‌تواند افزودن پیاده‌سازی برای متدهای خالی باشد. برای شروع فرض کنید که کامپیوتر اولین خانه خالی را که ببیند، بازی می‌کند.

در مراحل بعد می‌توان توسعه‌های زیر را برای بازی در نظر گرفت:

- تصادفی شدن بازیکن شروع کننده
- هوشمندسازی حداکثری نوبت بازی کامپیوتر به نحوی که مانند یک کاربر انسانی فکر کند و تصمیم بگیرد.
- بهبود پیغام‌های نشان‌داده شده به کاربر
- انجام اعتبارسنجی (validation) برای ورودی کاربر و نمایش پیغام‌های متناسب با ورودی اشتباه
- امکان انجام چندین باره بازی بعد از اتمام
- امکان نمایش نتایج کلی بعد از چند سری بازی با تعریف هر نوبت بازی به صورت یک شی (بعد از آموزش شی - گزایی)
- افزودن واسط گرافیکی به بازی (بعد آموزش واسط گرافیکی)