



Basics of Programming Final Project
Winter 2025

Course Instructor: Dr. Mohammad Navid Ghader



معرفی پروژه

سلام دوستان! امیدواریم حالتان خوب باشد و ترم خوبی را سپری کرده باشید. پروژه این ترم شما بازی منچ (Mensch) است که یکی از بازی‌های کلاسیک و محبوب تخته‌ای است. این بازی که ابتدا در آلمان معرفی شد، به عنوان یکی از اولین بازی‌های خانوادگی شناخته می‌شود و به دلیل سادگی و سرگرم‌کنندگی‌اش، همچنان مورد استقبال قرار می‌گیرد. در ادامه، بیشتر با نحوه بازی آشنا خواهیم شد. از اهداف انجام این پروژه دانستن تسلط کامل شما روی تمام مباحث مبانی برنامه سازی و مواجهه با یک پروژه بازی سازی هست و امیدواریم که از چیزهایی که یاد گرفتین به خوبی استفاده کنین

توضیحات بازی

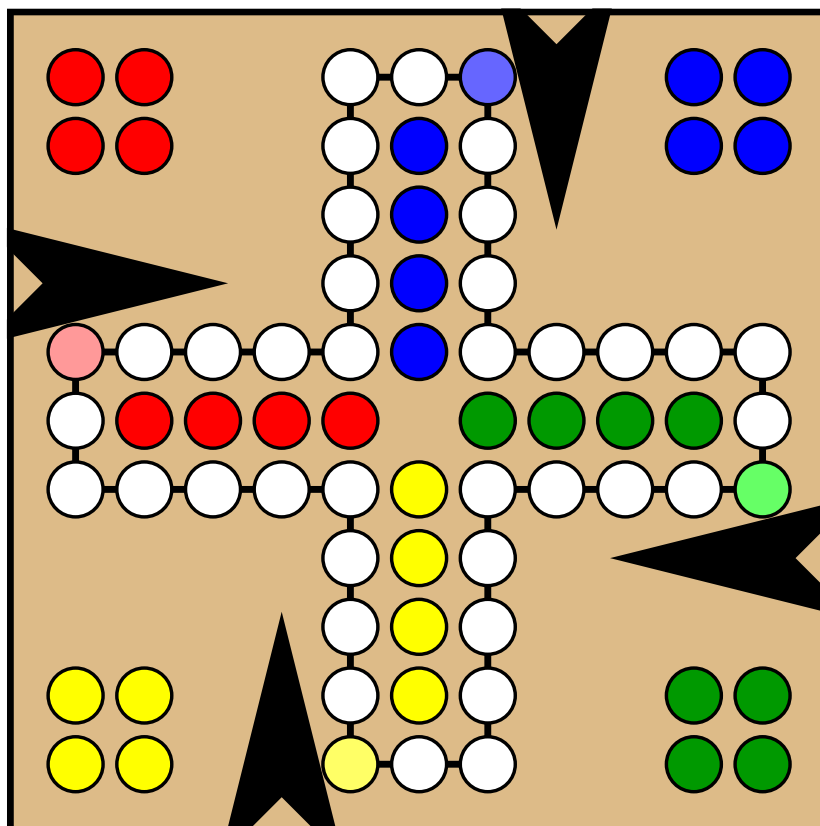
"Mensch" یک بازی رومیزی کلاسیک با قوانین ساده اما بسیار سرگرم‌کننده است. هدف بازی اینه که تمام مهره‌هاتون رو از خانه‌های مبدا به مقصد برسونید. در طول مسیر، مهره‌های حریف رو می‌تونید به خانه مبدا برگردونید تا مانع پیشرفت شون بشید.

شروع بازی

1. صفحه‌ی بازی شامل ۴ ناحیه رنگی (قرمز، آبی، سبز، زرد) و یک مسیر اصلی با ۴۰ خانه است.
2. هر بازیکن ۴ مهره‌ی هم‌رنگ داره که در خانه‌ی مبدا قرار می‌گیرند.
3. بازیکنان به نوبت تاس می‌ریزند. برای شروع حرکت هر مهره از خانه‌ی مبدا، باید عدد ۶ بیاورید.
4. وقتی مهره‌ای به مقصد رسید، نمی‌تونه برگرده.

پایان بازی

بازی زمانی تمام می‌شه که یکی از بازیکنان، تمام مهره‌های خودش رو به مقصد برسونه.



فازهای پیاده‌سازی

۱. منوی اصلی

ابتدا باید یک منوی اصلی طراحی کنید که گزینه‌های زیر رو شامل بشه:

- New Game
- How To Play
- Exit

برای هر گزینه توضیح کوتاهی بدید که کاربر متوجه بشه.

۲. محیط بازی

محیط بازی باید شامل موارد زیر باشه:

- نمایش موقعیت مهره‌ها روی صفحه.
- نمایش خانه‌های مبدا، مسیر اصلی و خانه‌های مقصد.
- استفاده از نمادهای ساده برای نشان دادن مهره‌ها و خانه‌ها.

۳. منطق بازی

1. تولید اعداد تصادفی برای تاس.
2. پیاده‌سازی حرکت مهره‌ها بر اساس قوانین بازی.
3. برخورد مهره‌ها: وقتی مهره‌ای روی خانه‌ای که مهره‌ی حریف قرار داره متوقف بشه، مهره‌ی حریف به خانه‌ی مبدا برمی‌گرده.
4. مدیریت نوبت‌ها و اختصاص نوبت اضافه در صورت آوردن عدد ۶.
5. تشخیص و اعلام برنده.

لینک ها و منابع برای آشنایی بیشتر با بازی:

- [ویدیو از گیم پلی بازی](#)
- [نمونه از بازی به صورت آنلاین](#)

لیست فیچر ها

در اینجا لیستی از تمام فیچرهای اجباری که باید پیاده سازی کنید مشاهده می کنید:

- زمانی که بازی رو اجرا می کنیم باید منوی اصلی (شامل بخش های زیر) به یوزر نمایش داده شود:
 - New Game
 - How to Play
 - Exit
- بازی با ریختن تاس شروع می شود و بازیکن با توجه به عدد تاس که از 1 تا 6 است می تواند مهره خود را حرکت دهد سپس نوبت به بازیکن بعدی می رسد. توجه کنید که شش جایزه دارد و دوباره بعد از حرکت باید تاس ریخته شود
- بازی می تواند به ازای 2 یا 4 بازیکن انجام شود .
- با شروع بازی تمام مهره ها در لانه 4 تایی قرار دارند و با آوردن 6 میتوانند وارد خانه شروع شوند
- خانه شروع هر بازیکن مختص خودش است و جزو مسیر بازی محسوب نمی شود و دیگر بازیکنان نمیتوانند وارد آن شوند (خانه رنگی کم رنگ در مسیر)
- اگر بازیکن باتوجه به تاس آمده مهره قابل حرکت نداشته باشد نوبت به نفر بعدی منتقل می شود در صورتی که بازیکن چند مهره قابل حرکت داشته باشد باید انتخاب انتخاب کند که کدام مهره حرکت کند برای این منظور می توانید برای مهره ها عدد تعیین کنید .
- بیش از یک مهره هم رنگ نمی توانند هم زمان روی یک خانه قرار بگیرند

- بازی با پر شدن 4 خانه مقصد برای یک بازیکن به اتمام می رسد .
- بازی در محیط ترمینال اجرا می شود . صفحه بازی باید نوبت بازیکن و عدد تاس بازیکن در هر لحظه را نشان دهد.

فازبندی

شما به دلخواه خودتان می توانید پروژه را از هر جایی که مناسب دیدید شروع کرده و کامل کنید. فازبندی زیر صرفاً یک پیشنهاد برای شماست که اگر سردرگم هستید از آن استفاده کنید. توجه کنید که استفاده از این فازبندی کاملاً اختیاری بوده و این قسمت فقط برای راهنمایی شماست.

فاز 1: پیاده سازی لوپ اصلی بازی

لوپ اصلی بازی یک لوپ `while` است که تا زمانی که بازی برنده نداشته ادامه پیدا می کند . تمام منطق بازی و آپدیت کردن المان های صفحه بازی و منطق بازی داخل این لوپ آپدیت می شود. ریختن تاس حرکت مهره ها و زدن بازیکن ها رو اینجا هندل کنید.

فاز 2: شروع بازی

در یک تابع جدا صفحه بازی را شروع کنید که مهره ها در جای خود قرار داشته باشند

فاز 3: نشان دادن صفحه بازی

صفحه بازی را با المان های موجود در آن با رعایت حداقل های زیبایی نشون بدین

فاز 4: پایان بازی

اینجا تصمیم بگیرید که وقتی بازی تموم میشه چه اتفاقی بی افته. شرط اینکه بازی کی تموم میشه هم باید برای اتمام `game loop` چک بشه و در یک متغیر نگهداری بشه.

نکات

شما اجازه دارید از تمامی مطالبی که در کلاس درس استاد و همین‌طور در کلاس‌های تمرین و خود تمرین‌ها بهتون آموزش داده شده استفاده کنید. شرطها، حلقه‌ها، آرایه‌ها، کار با فایل و کتابخانه ها از جمله ابزارهایی هستند که می‌توانید برای انجام پروژه از آنها استفاده کنید.

لیست کردن دقیق تمامی ابزارهایی که اجازه دارید از آنها استفاده کنید کار راحتی نیست، لذا اکیدا پیشنهاد می‌شود هرچیزی خارج از مطالب کلاس را نیاز داشتید، حتما با منتور پروژه‌تان در میان بگذارید تا بعدا مشکل‌ساز نشود.

نکته بسیار مهم: هرچیزی که مورد استفاده‌تان است را باید بلد باشید و بتوانید حین ارائه در موردش توضیح دهید و دلیل استفاده ازش را هم بیان کنید!

فیچر های امتیازی پیشنهادی

- مهم ترین فیچر امتیازی که میتونی اضافه کنی وجود بات هست یعنی علاوه بر بازیکن انسان یه ربات هم وجود داشته باشه که بتونه خودش بازی کنه (پنیک نکنید هوش مصنوعی و ماشین لرنینگ نیاز نیست) صرفا یه ربات ساده.
- افزودن توانایی چند بازیکن و با تعداد های متغیر و توانایی مشخص کردن تعداد ربات ها یعنی بازیکن بتونه یه ترکیب از بین 4 بازیکن و 3 تا ربات انتخاب کنه
- اضافه کردن موسیقی و آهنگ و افکت ها صوتی
- افزودن بخش تنظیمات
- میتونید تمام تنظیمات بازی از جمله وجود صدا رنگ ها تعداد بازیکن و همه اینها رو داخل یه منوی تنظیمات که توی منوی اصلی قابل دسترسیه تغییر بدید و داخل یه فایل ذخیره کنید که سری بعد که بازی دوباره اجرا شد با همون تنظیمات سری قبل اجرا شه
- در حالت عادی مهره با 6 وارد بازی میشه اما یه حالت هست که شاید بازیکن بخواد با آوردن عدد زوج شروع کنه یا با آوردن عدد یک پس این آپشن هم قرار بدین

- برای این پروژه Pygame مجازه (دقت کنید امتیازی نیست!) اگر یاد گرفته باشید میتونید با تلاش کمتر ظاهر بهتری بسازین فقط یه نکته رو دقت کنید اگر از این آپشن قراره استفاده کنید برنامه باید قابلیت اجرا بدون موس هم داشته باشه و مهم ترین نتیجه ش هم اینه که منو ها باید همه پویا (dynamic) باشن. منوی پویا یعنی یه گزینه فعال باشه و با زدن دکمه بالا و پایین گزینه فعال تغییر کنه و با زدن اینتر وارد اون گزینه بشه. کلا من ارادت خاصی به منوی پویا دارم پس سعی کنید حتما استفادهش کنین.
- یه فیچر امتیازی مهم که وجود داره استفاده از ورژن کنترل گیت هاب هست به این صورت که [این پروژه](#) رو fork می کنید اون رو clone می کنید رو سیستم خودتون و یه فولدر می سازید به اسم خودتون حالا پروژه رو داخل اون فولدر می سازین و روی پروژه کار میکنید البته برای این قسمت فقط روی گیت هاب آپلود کردن مهم نیست ها باید کامیت ها و برنج هایی که ساختین و مرج کردین مشخص باشه.

ارزیابی

موارد زیادی برای ارزیابی کدتان در نظر گرفته می شود، از جمله:

- رعایت نکات Clean Code، مانند خوانایی و سادگی کد
- رعایت اصول DRY و KISS

DRY: Don't repeat yourself

KISS: Keep it simple stupid

این دو اصل، از اصول مهم Clean Code هستند، که اولی به این معنی است که تکه های کدتان را تکرار نکنید، و اگر به یک کد بیشتر از یک بار نیاز دارید، آن را تبدیل به فانکشن کنید. دومی هم به این نکته اشاره می کند که تا جای ممکن بهتر است ساده کد بزنید، و از پیچیدگی بیش از حد و اضافه در کد جلوگیری کنید. برای مثال وقتی چند راه حل برای یک مسئله وجود دارد، ساده ترین راه را انتخاب کنید.

برای فهم بهتر این دو مفهوم به [این لینک](#) می توانید مراجعه کنید.

- معماری کد

بهتر است که بخش‌های مختلف پروژه را جدا کنید و مجزا پیاده‌سازی کنید، مثلا منطق بازی، بخش گرافیک، بخش دسترسی به فایل و غیره. در اینصورت برای دیباگ کردن ساده‌تر خواهد بود.

- کامنت گذاری (به خصوص برای توابع و سکشن‌های مختلف کد)

ارائه پروژه به صورت حضوری است و از ارائه دهنده انتظار می‌رود که به همه‌ی بخش‌های پروژه مسلط باشد و بداند هر فانکشن و هر خط کد چه نقشی دارد.

در کنار فیچرهای اصلی پروژه، موارد امتیازی و هرگونه ویژگی خلاقانه که پیاده‌سازی کنید در ارزیابی در نظر گرفته می‌شود و زیبایی و تمیزی کار قطعا تاثیر مثبت دارد.

ددلاین و تایم ارائه

برای تحویل پروژه تا پایان روز **جمعه 12 بهمن** باید پروژه را به صورت یک فایل ZIP داخل کوئرا آپلود کنید. گیت هم که جای خودش

نکته نهایی: ارائه پروژه به صورت **حضور**ی در روز **شنبه 13 بهمن** انجام خواهد شد.