

به نام خداوند علم و دانش



دانشکده مهندسی کامپیوتر

مبانی کامپیوتر و برنامه نویسی (سی پلاس پلاس)

پروژه اول (Game Of Life)

دکتر مرضیه داودآبادی

پاییز 1403

طراح: احمد رضا ذبیحی

در این مینی پروژه قصد داریم یک بازی ساده به همراه یک الگوریتم در محیط CLI پیاده سازی کنیم. بازی **Game of Life** که توسط ریاضی دان Conway ساخته شد یکی از بازی های جالب است که کاربرد های جالبی در دنیای هوش مصنوعی و الگوریتم ها دارد. این بازی شامل یک صفحه ماتریسی $N \times M$ می باشد. هر خانه از ماتریس می تواند دو حالت زنده یا مرده را داشته باشد. بازی از یک اسنپ شات رندوم از خانه های بازی شروع می شود. در اسنپ شات اولیه بعضی از خانه ها زنده و بعضی دیگر مرده هستند. تمامی ۸ خانه اطراف یک خانه همسایه های او حساب می شوند. (خانه های موجود در چهار جهت اصلی و چهار جهت فرعی) برای محاسبه اسنپ شات بعدی نیاز داریم تا مراحل زیر را انجام دهیم:

1. هر خانه که ۲، ۳، یا ۴ همسایه آن زنده باشند در اسنپ شات بعدی زنده خواهد بود.
 2. هر خانه ای که تعداد همسایه های زنده آن در محدوده بالا نباشند در اسنپ شات بعدی می میرد.
- در زمان بررسی هر خانه، فقط مقادیر همسایه های آن مهم می باشد؛ برای مثال اگر خود خانه زنده باشد و ۲ همسایه زنده داشته باشد در اسنپ شات بعدی تغییر نخواهد کرد و زنده خواهد بود. در نتیجه حالت فعلی هر خانه در اسنپ شات بعدی هیچ تاثیری در زنده یا مرده بودن آن خانه نمی گذارد.

بدین صورت بازی ادامه می‌یاد و اسنپ‌شات‌های بعدی ایجاد می‌شود. برای مطالعه بیشتر بازی Game of Life به این لینک مراجعه کنید. جهت مشاهده و انجام بازی نیز می‌توانید به این لینک مراجعه کنید.

وظیفه شما ایجاد یک زمینه تصادفی که تعداد خانه‌های زنده آن از کاربر ورودی گرفته می‌شود می‌باشد. اندازه این صفحه باید حداقل 10×15 باشد. برای نمایش خانه‌ها می‌توانید از دو کاراکتر متفاوت استفاده کنید. مثلاً خانه‌های زنده @ و خانه‌های مرده # باشند. (از هر ترکیب دلخواه که باعث بهتر شدن خروجی و خراب نشدن نظم برنامه باشد استفاده کنید.) در زیر یک نمونه از این برنامه را می‌توانید ببینید:



پس از ایجاد زمین بازی، بازی شروع به نمایش ۳ ایتريشن با فاصله زمانی ۲ ثانیه انجام می دهد. پس از نمایش این ۳ اسنپشات بازی از کاربر مختصات دو خانه مرده را به عنوان نقطه شروع و نقطه پایان می گیرد. سپس برنامه باید کوتاه ترین مسیر ممکن که تنها از خانه مرده عبور می کند

را به کاربر نشان دهد. (در صورت عدم وجود مسیر بین دو خانه پیمایی شامل عدم وجود مسیر و در صورت وجود چند کوتاه ترین مسیر با توجه به پیاده سازی تفاوتی بین نمایش دادن آن ها وجود ندارد.) برای پیاده سازی این قسمت می توانید از الگوریتم ها و روش های مختلفی استفاده کنید. راه حل پیشنهادی ما استفاده از الگوریتم **BFS** می باشد. برای مطالعه این الگوریتم می توانید به [این لینک](#) مراجعه کنید.

نکات مورد توجه:

- در طول پیاده سازی قسمت های مختلف به خطاهای احتمالی کاربر توجه داشته باشید و اقدام مناسب را انجام دهید. مثلاً در صورت دادن آدرس خانه ورودی که در ماتریس وجود ندارد باید پیام درخواست ورودی مجدد از کاربر صورت پذیرد. (پس از تمام کردن پروژه لیست اکسپشن های هندل شده خود را بنویسید.)
- برنامه باید جدول نمایش داده شده در صفحه را آپدیت کند. برای اینکار می توانید از روش های مختلف مانند رفرش کردن صفحه یا ادیت کردن یک کاراکتر در صفحه استفاده کنید. (نمایش پشت سر هم اسنپ شات ها در صفحه قابل پذیرش نیست.)
- مراحل پروژه را به ترجیحا به ترتیب انجام دهید تا در زمان ارائه حداقل به اندازه تلاش های کامل شده خود نمره دریافت کنید.
- لطفاً در هنگام نوشتن کد به تمیز بودن و رعایت اصول توجه و رعایت کنید.

قسمت های امتیازی:

- افزودن قابلیت ایجاد زمین ورودی بر اساس ورودی کاربر. (اندازه زمین و/یا خانه های زنده اولیه)
- قابلیت نمایش مراحل طی شده توسط الگوریتم مسیر یابی به مرحله به مرحله.

- قابلیت مپ چسبیده یا چرخشی. (بالاترین ردیف به پایین ترین ردیف متصل است و همسایگی هم محسوب می شوند. همچنین برای ستون ها. (شبیه قابلیت مار بازی که اگر مار از سمت چپ زمین خارج شود از سمت راست وارد زمین می شود.))
 - موارد دیگری که در ذهن خودتان باشد با هماهنگی TA قابل اضافه شدن می باشد.
- از همگی پروژه ها ارائه گرفته خواهد شد ولی به اندازه حداقل دو صفحه داکویمنت توضیحات پروژه (از مراحل انجام شده و شرح الگوریتم ها و ...) نیاز می باشد. پس از انتهای پروژه نوشتن داکویمنت را فراموش نکنید.
- موفق باشید.