



DECEMBER 2023

PROJECT 2

M.Amin HosseinNiya

Presented to:
Dr. BourBour



این گزارش شامل جزئیات پیاده‌سازی یک اسکریپت پایتون است که برای حل یک معما به کمک الگوریتم جستجوی اول سطح (BFS) طراحی شده و نتایج حاصل از آن را تصویرسازی می‌کند. اسکریپت شامل توابعی برای parse کردن فایل ورودی که معما را نمایش می‌دهد، حل معما برای یافتن کوتاه‌ترین مسیر برای دو شخصیت (راکت و لاکه) و تصویرسازی کامل معما به همراه مسیرهای طی شده توسط راکت و لاکه با رنگ‌های متفاوت برای وضوح بهتر است.

بررسی اسکریپت

اسکریپت شامل قسمت‌های کلیدی زیر است:

Parse کردن معما

- `parse_maze(input_file)`: تابعی برای تبدیل معمای نمایش داده شده به صورت متنی به یک ساختار داده‌ای برای پردازش. این تابع فایل ورودی را برای تعیین تعداد گره‌ها (n)، رنگ‌های مرتبط با مسیرها، موقعیت‌های شروع برای راکت ($s1$) و لاکه ($s2$)، و ساختار گراف نماینده معما می‌خواند.

حل کردن معما

- `bfs(graph, start, goal)`: تابع الگوریتم BFS که با توجه به یک گراف و گره‌های شروع و هدف، کوتاه‌ترین مسیر بین این گره‌ها را به صورت لیستی از شناسه‌های گره برمی‌گرداند.

پرینت کردن مسیر

- `print_path(path_r, path_l, colors)`: این تابع مسیرهای یافت شده توسط راکت ($path_r$) و لاکه ($path_l$) را می‌گیرد و هر قدم از سفرهای مربوطه را به گونه‌ای که شناسه‌های گره با نمایش‌های الفبایی آن‌ها برای خوانایی انسانی همخوانی دارند، پرینت می‌کند.

تابع مصورسازی

`visualize_maze(graph, path_r, path_l, colors, n)`: این تابع تمثیل‌سازی دیداری از معما را با استفاده از کتابخانه‌های NetworkX و matplotlib می‌سازد. گراف نمایش دهنده معما ابتدا ایجاد شده و سپس مسیرهای طی شده توسط راکت و لاکه با فلش‌هایی برای نشان دادن جهت حرکت روی آن قرار می‌گیرند.

کارکرد کد

الگوریتم BFS

تابع bfs الگوریتم استاندارد BFS را برای گراف‌های بی‌وزن برای یافتن کوتاه‌ترین مسیر پیاده‌سازی می‌کند. این فرآیند شامل تکرار بر روی گراف سطح به سطح تا رسیدن به گره هدف و ثبت گره‌های بازدید شده برای جلوگیری از بازدید مجدد آن‌ها می‌باشد.

تکنیک تصویرسازی

تابع تصویرسازی (visualize_maze) از کتابخانه NetworkX برای ایجاد گراف استفاده می‌کند که در آن گره‌ها تقاطع‌ها و لبه‌ها مسیرهای بین آن‌ها هستند. سپس کتابخانه matplotlib برای نمایش دیداری این گراف به کار برده می‌شود که گره‌ها و لبه‌ها به ترتیب به صورت دایره‌ها و خطوط نمایش داده می‌شوند.

- گره‌ها و لبه‌های گراف: گره‌ها و لبه‌ها به گراف G افزوده شده و با استفاده از موقعیت‌های از پیش تعیین شده توسط الگوریتم spring_layout کشیده می‌شوند.
- نقشه‌برداری رنگ: یک فرهنگ رنگ‌های برداری ایجاد شده تا رنگ‌های واقعی را به رنگ‌های نمادین موجود در گراف اختصاص دهد.
- گراف مسیرها: یک گراف جهت‌دار DG جداگانه برای کشیدن مسیرها با فلش‌ها ایجاد شده است. این تضمین می‌کند که فلش‌ها فقط روی مسیرها و نه کل گراف نشان داده می‌شوند.
- برجسته کردن خروجی: گره خروجی (هدف) به گونه‌ای متفاوت برای تمایز آن از سایر گره‌ها در معما برجسته شده است.

اجرا

زمانی که اسکریپت اجرا می‌شود، با فراخوانی solve_maze('input.txt') شروع می‌کند، جایی که 'input.txt' فایلی است که معمای حل شده را دارد. مسیر فایل و محتویات آن ساختار معما و پیچیدگی مسیرهای درون آن را تعیین می‌کنند.