



## Path Finding

در این فاز از پروژه سعی داریم که یک مازول مسیریابی به فاز قبلی پروژه اضافه کنیم. به این صورت که کاربر دو نقطه از نقشه را انتخاب می‌کند و در نقشه مسیر مناسبی برجسته<sup>۱</sup> می‌شود. بدیهی است که مسیر مشخص شده باید کوتاهترین مسیر ممکن باشد و با حرکت روی نقشه و زوم کردن از روی صفحه نمایش پاک نشده و همچنان خیابان‌های مناسب را مشخص کند.

اطلاعات مناسب برای مسیریابی به صورت جداگانه در پروژه قرار می‌گیرد (با فایل‌های نمایش نقشه یکسان نیستند). این اطلاعات در قالب فایلی شامل یک گراف وزن‌دار و جهت‌دار خواهد بود و لیست مجاورت آن در فایل موجود است. رئوس گراف نماینده تقاطع‌ها و یال‌ها نماینده خیابان‌ها هستند. خیابان‌های یک‌طرفه به صورت یک یال جهت‌دار مشخص می‌شوند و به ازای خیابان‌های دو طرفه هر دو یال بین دو رأس وجود خواهد داشت. در خط اول فایل تعداد رئوس گراف (N) نوشته شده است. در N خط بعدی مختصات هر رأس به صورت دو عدد X و Y آمده است. در N خط بعدی لیست همسایه‌های هر رأس آمده است. در ابتدای هر خط عدد k آمده که تعداد همسایه‌ها را نشان می‌دهد. سپس k جفت عدد (index, width) در خط وجود دارد که شماره رأس همسایه و فاصله تا آن را مشخص می‌کند (شماره رأس از جنس integer و از صفر شروع می‌شود و طول یال float است). مثال:

3

1.0 1.0

1.0 2.0

2.5 1.0

2 1 1.0 2 1.5

1 2 1.803

1 0 1.5

## هدف اصلی فاز دوم پروژه:

در کد شما برای مسیریابی باید یک تابع وجود داشته باشد که مختصات source و destination را به عنوان ورودی بگیرد و یک عدد width و یک آرایه path را خروجی دهد. در path طول کوتاهترین مسیری که تابع

<sup>1</sup> highlight

یافته است قرار دارد و در path مجموعه مختصات رئوس مسیر قرار دارد (مختصات یک pair از X و Y هستند). با استفاده از این تابع در فاز اول و اضافه کردن UI مناسب به برنامه امکان مسیریابی را به برنامه خود اضافه کنید. دقت کنید که ممکن است مختصات source و destination روی تقاطع‌ها قرار نداشته باشند. شما باید روشی برای حل این مشکل ارائه دهید و آن را پیاده سازی کنید.

### نکات مهم در ارزیابی:

- نقشه باید دارای UI مناسب برای مسیریابی بوده و مسیر مد نظر (حتی بعد از حرکت دوربین) روی نقشه برجسته شود.
- برای مسیریابی باید تابع جداگانه‌ای وجود داشته باشد که بتوان بدون اجرای کل پروژه از آن استفاده کرد. (این تابع در تحویل حضوری بررسی می‌شود)
- طول مسیر بازیابی شده، در نمره شما تاثیر داشته و تا حد امکان باید به مقدار بهینه نزدیک باشد. (نسبت طول مسیر بهینه به طول مسیر پیشنهادی در نمره شما تاثیر دارد).
- یکی از ملاک‌های نمره، سرعت پیدا کردن مسیر مناسب توسط برنامه است. شما می‌توانید راه حلی برای کم کردن این زمان پیشنهاد داده و آن را پیاده‌سازی کنید. سعی کنید تا جای ممکن این زمان را کاهش دهید. (در زمان تحویل باید قادر به دفاع از راه حل خود باشید)
- دقت کنید که راه حل پیشنهادی شما ممکن است مسیر بهینه را پیشنهاد ندهد و در ازای آن زمان بازیابی مسیر به مقدار زیادی کاهش یابد. (شما می‌توانید دقت را تا حدی کاهش دهید و به جای آن تلاش کرده و زمان را بهبود دهید). در این حالت باید در مورد مقدار ماکزیمم خطای طول مسیر توضیح دهید (به طور مثال نشان دهید که راه حل پیشنهادی حداکثر ۲۰٪ خطا دارد و زمان بازیابی آن ۴۰٪ زمان بازیابی معمولی است).

### نکات پایانی:

- ✓ برنامه‌ی شما باید به زبان پایتون نوشته شده و با مفسر Python 2.7 روی لینوکس اجرا شود.
- ✓ در صورت مشاهده‌ی هرگونه تشابه در کدها نمره‌ی هر دو طرف ۱۰۰- منظور می‌گردد و در بار دوم نمره‌ی صفر برای درس منظور می‌گردد.
- ✓ در صورت وجود هر گونه سوال می‌توانید به فروم درس مراجعه کنید و سوال خود را مطرح کنید.