

به نام خدا



دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

محمد کاظم هرندی

۴۰۰۳۶۲۳۰۳۹

مبانی و کاربردهای هوش مصنوعی

گزارشکار پروژه اول

پاییز ۱۴۰۳

۱. معرفی پروژه

هدف از این پروژه، طراحی و پیاده‌سازی یک بازی مار با استفاده از الگوریتم‌های مختلف مسیریابی هوش مصنوعی است. این پروژه شامل سه بخش اصلی است:

- طراحی محیط گرافیکی بازی با استفاده از Pygame.
 - پیاده‌سازی سه الگوریتم مسیریابی A*, BFS و DFS.
 - مقایسه عملکرد این الگوریتم‌ها از نظر سرعت، دقت، و نتایج خروجی.
- در این پروژه، تلاش شده است تا بازی از لحاظ گرافیکی بهبود یابد و تجربه کاربری روانی ارائه شود. همچنین پیاده‌سازی‌ها در محیط GitHub مستندسازی شده‌اند تا قابلیت مشاهده و مشارکت در کد فراهم باشد.

۲. روند کلی پروژه

۱. طراحی محیط بازی:

- محیط بازی شامل یک شبکه از سلول‌ها است که مار (Snake) و میوه (Fruit) در آن قرار دارند.
- مار باید با استفاده از الگوریتم‌های هوش مصنوعی، مسیری برای رسیدن به میوه پیدا کند.
- عناصر گرافیکی شامل مار، میوه، و پس‌زمینه به‌گونه‌ای طراحی شده‌اند که جذابیت بصری و شفافیت گیم‌پلی را تضمین کنند.

۲. ساختار کد: پروژه از چند فایل اصلی تشکیل شده است:

- GameGUI.py: مسئولیت گرافیک بازی و تعامل با کاربر.
- GameController.py: مدیریت وضعیت بازی و ارتباط میان الگوریتم‌ها و محیط بازی.
- Algorithms.py: شامل پیاده‌سازی الگوریتم‌های A*, BFS، و DFS برای مسیریابی.

۳. مراحل اجرا:

- انتخاب الگوریتم از منوی اصلی.
- اجرای بازی با استفاده از الگوریتم انتخاب‌شده.
- نمایش عملکرد الگوریتم و جمع‌آوری آمار.

۴. مستندسازی و بهبود:

- نتایج و مقایسه‌ها در قالب اسکرین‌شات‌های بازی و توضیحات در این داکيومنت ارائه شده‌اند.
- تغییرات گرافیکی و بهبودهای بصری برای جذاب‌تر شدن بازی پیاده‌سازی شده‌اند.

۳. توضیح الگوریتم‌های مسیریابی

۱. الگوریتم A*

- شرح: الگوریتم A* یکی از قوی‌ترین و سریع‌ترین الگوریتم‌های مسیریابی است که از ترکیب دو رویکرد استفاده می‌کند:

۱. هزینه واقعی پیموده‌شده.

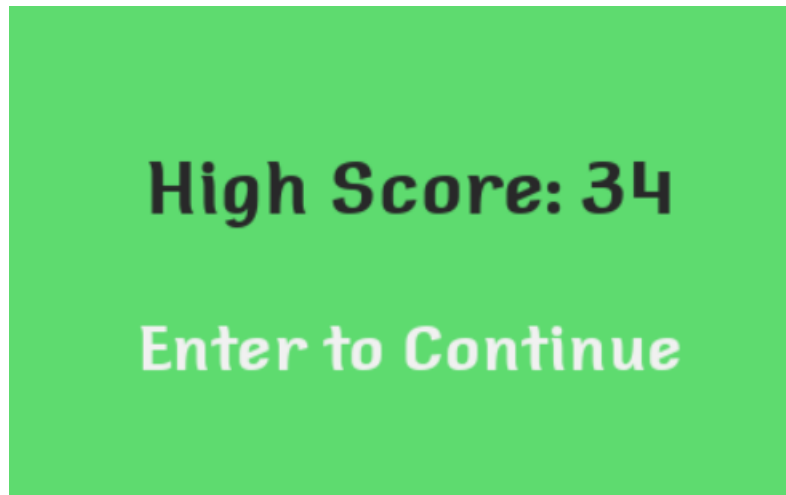
۲. هزینه تقریبی باقی‌مانده تا مقصد. این الگوریتم به دلیل بهینه‌سازی عملکرد و دقت بالا در بازی استفاده شده است.

- ویژگی‌ها:

○ عملکرد سریع‌تر نسبت به BFS و DFS در محیط‌های بزرگ.

○ ارائه کوتاه‌ترین مسیر با استفاده از تابع هزینه.

- تصویر خروجی:



۲. الگوریتم BFS

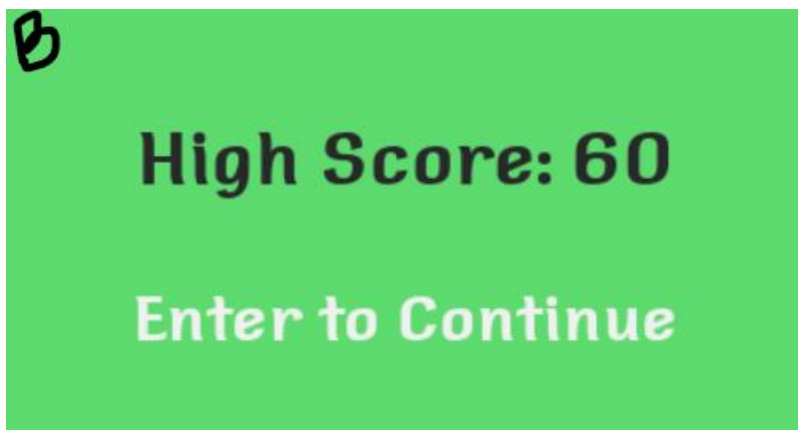
- شرح: BFS یک الگوریتم جستجوی گراف است که تمام مسیرهای ممکن را به ترتیب لایه‌ای بررسی می‌کند. این الگوریتم برای یافتن مسیرهای کوتاه مناسب است.

- ویژگی‌ها:

○ تضمین پیدا کردن کوتاه‌ترین مسیر.

○ کندتر از A* در محیط‌های پیچیده.

- تصویر خروجی:



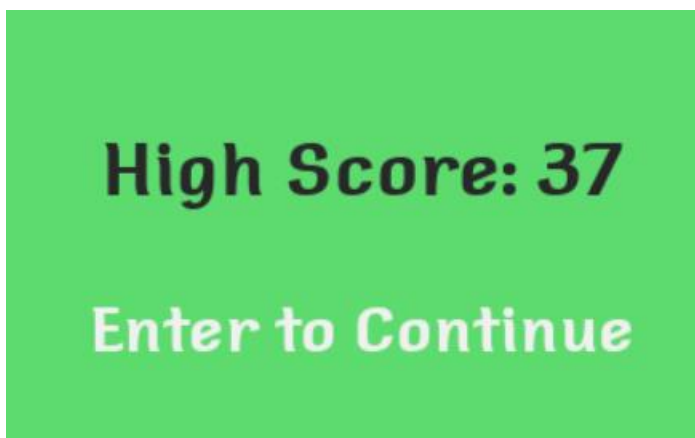
۳. الگوریتم DFS

- شرح: DFS از عمق برای جستجوی گراف استفاده می‌کند. این الگوریتم با تمرکز بر یک شاخه، تا انتها آن را جستجو کرده و سپس به شاخه‌های دیگر بازمی‌گردد.

- ویژگی‌ها:

- مناسب برای یافتن مسیر در گراف‌های کوچک و ساده.
- ممکن است مسیر بهینه را پیدا نکند.

- تصویر خروجی



۴. مقایسه الگوریتم‌ها

ویژگی	A*	BFS	DFS
سرعت	سریع	متوسط	کند در برخی موارد
کوتاه‌ترین مسیر	تضمینی	تضمینی	تضمین‌شده نیست
پیچیدگی حافظه	متوسط	بالا	کم
کاربرد در بازی	بهترین عملکرد	مناسب	کمتر مناسب

توجه شود تستی که برای این پروژه نوشته شده بود تلاش می‌کرد الگوریتم‌ها را دچار چالش کند و به این دلیل نتایج با مواردی که در جدول یاد شده تطابق ندارد.

۵. تغییرات مثبت و بهبودها

در این پروژه، علاوه بر پیاده‌سازی الگوریتم‌ها، تغییرات گرافیکی و بهبودهایی اعمال شده‌اند که شامل موارد زیر است:

۱. افکت‌های بصری:

- نمایش مسیر به‌صورت خط‌های رنگی برای الگوریتم‌ها
- انیمیشن پویای مار و میوه.

۲. حالت‌های مختلف بازی:

- افزودن امکان مشاهده مسیرهای مختلف (highlight) برای الگوریتم‌ها.

۳. مستندسازی و نسخه‌سازی:

- استفاده از GitHub برای ذخیره و مدیریت پروژه.
 - فراهم کردن یک README در GitHub برای توضیح نحوه اجرا.
- کدهای پروژه در یک مخزن GitHub ذخیره شده‌اند تا قابلیت مشاهده و مشارکت برای دیگران فراهم شود. این مخزن شامل:

- فایل‌های کد بازی.
- اسکرین‌شات‌ها و نتایج اجرا.