

# پروژه‌ی درس رباتیک

آریا پرویزی ۹۶۲۴۳۰۷۷

کوروش خاوری مقدم ۹۷۲۴۳۰۲۸

الگوریتم انتخابی:

توضیح الگوریتم bug1:

این الگوریتم به این صورت است که ابتدا در خط مستقیم در راستای هدف شروع به حرکت می‌کند. اگر در مسیر، ربات به مانعی برخورد داشته باشد، پیرامون آن مانع یک دور کامل می‌زند و در این حین کوتاهترین نقطه نسبت به هدف را می‌یابد. سپس بعد از رسیدن به اولین نقطه‌ای که با مانع برخورد داشته است (hit point)، به سمت نقطه‌ای که به هدف نزدیکتر بوده، مماس با مانع حرکت کرده و از آن نقطه در به سمت هدف دوباره شروع به حرکت می‌کند. و این چرخه ادامه دارد تا به هدف برسد.

مقایسه عملکرد bug1 با bug2:

مجموعه محیط آزمایشی الگوریتم های ربات در شکل زیر مشهود است



هدف نیز با دایره‌ی قرمز رنگی در ویدیوها مشخص شده است.

مسافت طی شده توسط باگ ۱

۳.۷۴۷

مسافت طی شده توسط باگ ۲

۱.۸۹۳

در ویدیوهای پیوست شده مسیرهای حرکت هر الگوریتم برای ربات مشخص است. همچنین ویدی‌های پیوست شده به علت کم شدن حجم فایل‌ها با سرعت ۶ برابر ضبط شده‌اند.

با توجه به نتایج مسافت طی شده توسط ربات در هر الگوریتم در شبیه ساز، می‌توان نتیجه گرفت که باگ ۲ عملکرد بهتری دارد و همچنین به تبعیت از آن باگ ۲ باتری و زمان کمتری مصرف خواهد کرد.

توصیفات اصلی کد:

در این شبیه ساز از ربات epuck از شرکت GCTronic استفاده شده است

سنسورهای مورد استفاده نیز قطب نما و جی پی اس است. همهی شرایط ایده آل و بدون نویز فرض شده اند.

وضعیت های اصلی باگ ۱:

CALCULATE\_TARGET\_ANGLE

در این حالت ربات شروع به محاسبه ی زاویه و خط بین نقطه ی جاری و نقطه ی هدف می کند.

ROTATE\_HEAD\_TOWARD\_GOAL

در این حالت اگر سر ربات در راستای خط متصل به نقطه ی هدف نباشد، سر خود را به اندازه ی زاویه ی بین این خط و سر خود می چرخاند.

MOVE\_TOWARD\_TARGET

در این حالت ربات فقط به سمت هدف با سرعت ماکسیمم شروع به حرکت می کند. (از هر نقطه ای که باشد و خط و شیبی که در حالت CALCULATE\_TARGET\_ANGLE محاسبه شده است انجام می شود برخلاف باگ ۲ که باید حتما در راستای خط واصل نقطه ی شروع و نقطه ی هدف).

BOUNDARY\_TANGENT\_DIRECTION

در این حالت اگر ربات به مانعی رسیده باشد، سر خود را موازی و به صورت مماس با مانع در می آورد.

OBTSACLE\_FULL\_SCAN

در این حالت ربات مانع به طور کامل دور زده و به نقطه ی اولیه ی این مانع بازمی گردد، همچنین در این حین کوچکترین فاصله تا هدف را محاسبه و مختصات آن را ذخیره می کند.

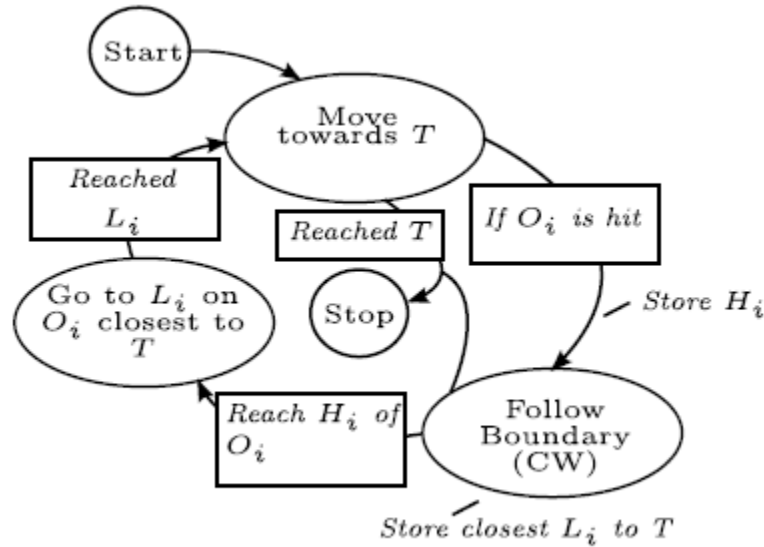
GOTO\_LEAVE\_POINT

در این حالت ربات به سمت نقطه ای از مانع و موازی با مانع که کمترین فاصله را تا هدف داشته است، حرکت می کند. که این فاصله در حالت محاسبه شده بود.

END

همان وضعیت پایانی یا رسیدن به مقصد است.

شکل ماشین حالت این الگوریتم به صورت زیر است.



وضعیت‌های اصلی باگ ۲:

#### CALCULATE\_TARGET\_ANGLE

در این حالت، خط M-line (همان خطی که نقطه‌ی ابتدایی ربات را به نقطه‌ی هدف متصل میکند) محاسبه می‌شود.

#### ROTATE\_HEAD\_TOWARD\_GOAL

در این حالت، ربات سر خود را به سمت هدف در راستای خط M-line می‌چرخاند.

#### MOVE\_TOWARD\_TARGET

در این حالت، ربات با حداکثر سرعت در راستای خط M به سمت هدف حرکت می‌کند.

#### BOUNDARY\_TANGENT\_DIRECTION

در این حالت، وقتی ربات به مانعی نزدیک شد، جهت سر خود را طوری تغییر می‌دهد تا بر محیط مانع مماس شود.

#### BOUNDARY\_WALL\_FOLLWING

در این حالت، ربات مانع را دور می‌زند تا موقعی به نقطه‌ی تقاطع خط M و محیط مانع برسد، به طوری که از نقطه‌ی شروع در مانع (hit point) به هدف نزدیکتر باشد، در غیر اینصورت به چرخیدن ادامه داده تا نقطه‌ی تقاطع دیگری پیدا کند یا به نقطه‌ی اولیه از مانع (hit point) برسد. اگر به نقطه‌ی اولیه برسد یعنی مسیر بازی به هدف پیدا نکرده است.

END

همان وضعیت پایانی یا رسیدن به مقصد است.

شکل ماشین حالت این الگوریتم به صورت زیر است.

