

به نام خدا

راهنمای بیس مرصاد : راهنمای تابع های حرکت (BasicDash) کلاس هایی که برای Dash کردن در بیس مرصاد وجود دارند در فایل BasicDash.h تعریف شده اند. (کد آن ها در فایل BasicDash.cpp وجود دارد) مهمترین این کلاس ها PerfectGotoPoint و ForbiddenAreaGotoPoint هستند. شکل کلی استفاده از این کلاس ها به این شکل است که ابتدا یک شی از یکی از کلاس های GotoPoint ساخته می شود که تمام اطلاعات به آن داده می شود، مانند نقطه هدف، dashPower و ... (هر کدام از این کلاس ها اطلاعات مورد نیاز خود را به عنوان ورودی در تابع constructor خود میگیرند که خاص آن کلاس است و لزوماً در تمام کلاس های GotoPoint مشترک نیست) سپس با استفاده از تابع getCommand یک خروجی از جنس Command* گرفته میشود. (مراحل decide در تابع constructor انجام میشود) مثلاً در کلاس های DefensePlan یا Offense و ... در صورت انتخاب عمل Dash باید متغیر command که عضوی از کلاس Offense است برابر این خروجی قرار داده شود:

```
command = PerfectGotoPoint({inputs}).getCommand();
```

توضیح جزئی کلاس ها:

- PerfectGotoPoint(const Point target, const Body& body, double dist_thr, double dashPower, double dir_thr, bool useBackDash, bool forceBackDash)
Point target : یک شی از struct Point است که حاوی دو مقدار x و y است و برای مشخص شدن نقطه هدف می باشد.
const Body& body : شی خود agent است که با worldModel->getBody() بدست می آید.
double dist_thr : میزان فاصله که اگر agent با نقطه هدف باشد از dash کردن منصرف می شود (به نقطه هدف نزدیک تر نمیشود) که مقدار اولیه ۱۰۰ اندازه معقول آن است.
double dashPower : که حداکثر آن ۱۰۰ است و قدرت dash کردن است (هر چه بیشتر باشد dash قوی تر خواهد بود)
double dir_thr : اگر برای رسیدن به نقطه هدف نیاز به turn کردن با زاویه ای کمتر از dir_thr باشد از turn کردن صرف نظر میشود. مقدار معقول آن ۱۰ تا ۲۰ درجه است.
bool useBackDash : در صورت false بودن از backDash استفاده نمیشود.
bool forceBackDash : در صورت true بودن برای رسیدن به نقطه حتماً از backDash استفاده میشود.
PerfectGotoPoint کامل ترین کلاس GotoPoint است و هر حالتی را میتوان با توجه به ورودی های آن شبیه سازی کرد. (تمام نیاز ها با همین کلاس برطرف میشود)

- ForbiddenAreaGotoPoint(const Point point, const Point circleCenter, float circleRadius, const Body& body)
Point point : مشخص کننده هدف است.
Point circleCenter : مرکز دایره ای است که بازیکن وارد آن نمیشود.
float circleRadius : شعاع دایره ای است که بازیکن وارد آن نمیشود.
Body& body : خود agent که با worldModel->getBody() بدست می آید.

ForbiddenAreaGotoPoint : این کلاس به نقطه هدف به صورتی می رود که از داخل دایره ورودی نگذرد. مثلاً یکی از کاربردهای آن این است که در playMode های nonPlayOn که توپ برای حریف است بازیکن نباید از یک دایره به شعاع ۹ متری توپ عبور کند (اگر بخواهد وارد آن شود stamina آن کم می شود و نمیتواند وارد شود) دیگر constructor های این کلاس به جای دایره یک مستطیل یا ... مشخص میکنند که بازیکن داخل آن نمیرود (مثلاً در حالت ضربه دروازه حریف بازیکن های ما نمیتوانند داخل محوطه جریمه حریف شوند)

3. DangerAreaGotoPoint(const Point target, const Body& body, const Ball& ball, Point ballPos, float dashPower)

Point target : مشخص کننده هدف است.

Body& body : خود agent که با worldModel->getBody() بدست می آید.

Ball& ball : شی توپ که با worldModel->getBall() بدست می آید.

Point ballPos : محل قرار گرفتن توپ (ممکن است استفاده کننده بنا به دلایلی نخواهد از جای اصلی توپ استفاده کند، مثلاً اگر توپ در حال حرکت باشد جایی که توپ متوقف می شود را در نظر بگیرد)

float dashPower : قدرت Dash کردن.

DangerAreaGotoPoint : این GotoPoint به صورتی عمل میکند که بازیکن همیشه بتواند توپ را ببیند (اگر توپ پشت بازیکن باشد از backDash استفاده میکند) و چون در بسیاری از موارد backDash استفاده می شود (حداکثر زاویه چرخش گردن بازیکنان ۱۲۰ درجه است) تنها در محوطه جریمه که اهمیت دانستن جای دقیق توپ برای دفاع ها زیاد است نیاز به استفاده از این کلاس است. اگر در بقیه موقعیت ها هم از این GotoPoint استفاده شود بدلیل استفاده از backDash مشکل stamina برای بازیکنان به وجود میاید (stamina کم می آید)