

## -سند طراحی بازی (Game Design Document)-

**تهیه کننده: دهکيا سامانی خوزانی**

**توجه!**

۱- این یک سند در حال تکمیل بوده و با پیشروی در روند ساخت بازی، این نوشته نیز بروز شده و ممکن است بخش‌هایی از آن به صورت کلی یا جزئی، دستخوش تغییر شوند. توصیه می‌شود که برای اطلاع از آخرین تغییرات و بروزرسانی‌های پروژه بازی، این نوشته را به صورت دوره‌ای و با دقت مطالعه کنید.

۲- تنها تهیه کننده سند اجازه ویرایش و ادیت این سند را داشته که با هماهنگی قبلی با سایر اعضای تیم صورت خواهد گرفت.

**توجه!**

## - مشخصات پایه‌ای -

**نام بازی:** (تایید نشده)

**نام اولیه بازی:** پروژه ترمیناتور (Project Terminator)

**سبک بازی:** اکشن-شوتر میدان نبردی (Arena)

(Shooter-Action)

**موضوع بازی:** نبرد ربات‌های جنگی در میدان مبارزه

**مخاطبان بازی:** برنامه‌نویسان و تا حدودی بازیکنان

(Gamers)

**موتور بازی‌سازی:** Godot

**زبان‌های برنامه‌نویسی:** C#- GDScript

**سبک هنری بازی:** ۲ بعدی مطلق با زاویه دوربین نیم‌رخ

(Sideways Camera)

**تعداد اعضای تیم سازنده:** ۴

**تعداد بازیکنان بازی:** ۲

**پلتفرم بازی:** کامپیوتر شخصی

**مهلت تحویل پروژه:** ۲۰ یا ۲۱ آبان ماه

**فازهای مورد نیاز برای تکمیل پروژه:** ۳ (۱- فاز تولید نمونه

اولیه یا اموی‌سی، ۲- فاز تکامل و ارتقای بازی ۳- فاز

نهایی‌سازی بازی)



## -وظایف اعضای گروه-

**\*طراح بازی (Game Designer):** دهکیا سامانی خوزانی  
**توسعه‌دهندگان بازی (Game Developers):** محمدسینا  
سلیمانی و صدرا برومندفر  
**مدیر دارایی‌ها (Assets) و منابع (Resources) بازی:**  
پوروشا شاملی

- طراح بازی مسئولیت تعیین و طراحی کلیات و جزئیات و ریزه‌کاری‌های بازی را بر عهده دارد. مواردی همچون تهیه GDD، طراحی مرحله، چالش‌های بازی، محدودیت‌ها، امکانات بازی، قوانین بازی، مکانیک‌های اصلی و فرعی، طراحی شخصیت‌ها و قابلیت‌های آنها (ربات‌ها) و دیگر موارد مربوط به طراحی بازی بر عهده گیم دیزاینر می‌باشد. همچنین، طراح بازی مسئول انجام تقسیم وظایف تیم توسعه نیز بوده و در کل نقش مدیر پروژه (نه رئیس آن!) را نیز ایفا می‌کند.

- توسعه‌دهندگان بازی مسئولیت کدزنی بازی، کار با موتور بازی‌سازی (Game Engine)، انتقال و اجرای طرح‌های داده شده توسط طراح بازی در بستر کامپیوتری و مواردی همچون ساخت توابع، کلاس‌ها، ساختمان‌های داده و استفاده از تکنولوژی‌های در دسترس مخصوص ساخت ویدئو گیم را به عهده دارند. آزادی عمل به حدی به توسعه‌دهندگان داده شده است که خود امکان انتخاب

موتور بازی سازی و زبان برنامه نویسی را داشته و درباره نحوه اعمال و استقرار طرح های گیم دیزاینر نیز اختیار عمل دارند.

- مدیر دارایی ها و منابع بازی مسئولیت جمع آوری، مدیریت، نام گذاری صحیح و در صورت لزوم ویرایش و تغییر دارایی های بازی سازی را بر عهده داشته؛ این دارایی ها شامل آیکون ها، اسپرایت ها (Sprites)، عکس ها، افکت های صوتی، موسیقی پس زمینه و دیگر موارد مشابه می شوند. لازم به ذکر است که ایشان باید دارایی ها و منابع را از سرتاسر اینترنت جمع آوری کرده و مطمئن شود که فایل های دانلودی انتخاب شده به صورت رایگان بوده و استفاده از آنها در این پروژه بلامانع است.

در حال حاضر، مخازن (Repositories) مورد نیاز برای تمامی تیم ها (طراحی، توسعه و هنر) در سایت گیت هاب (Github) ساخته شده اند: نام مخزن طراحی بازی در گیت هاب، `Game_Design_Project_Terminator`، نام مخزن توسعه بازی فعلا به صورت موقت `new-godot-game` بوده که باید به نام `Game_Development_Project_Terminator` تغییر نام پیدا کند (به دلیل نظم بهتر و مدیریت ساده تر) و در نهایت، نام مخزن هنر بازی نیز `Game_Art_Project_Terminator` می باشد. هیچ بخشی از کار، تنها به صورت محلی (Local) بر روی سیستم ها و کامپیوترهای اعضای تیم نباید باقی بماند و باید نسخه ای از آنها به صورت آنلاین نیز در سایت ذکر شده، قرار بگیرد؛ تمامی اعضای پروژه وظیفه دارند که در صورت به وجود آمدن کوچکترین تغییرات یا بروزرسانی در کار و حیطه خود،

سریعا محصول تولید شده جدید را بر روی سایت بارگذاری کنند.  
در صورت کمبود حافظه یا برخورد با محدودیت‌های سایت  
گیت‌هاب، از دیگر سرویس‌های ذخیره‌سازی و Version  
Control استفاده خواهد شد که با هماهنگی با تمامی سازندگان  
صورت خواهد گرفت.

□ [نام کاربری طراح بازی در گیت‌هاب: dehkiask]

□ [نام کاربری توسعه‌دهندگان بازی در گیت‌هاب:

msina-soleimani و arybrf]

□ [نام کاربری مدیر دارایی‌ها و منابع بازی: p0r0sha (به جای

0 انگلیسی، از عدد صفر استفاده شده است)]

\*بدیهی است که هر کدام از اعضای تیم با هماهنگی با طراح بازی،  
می‌توانند ایده‌های خود را (ترجیحا در کانال تلگرامی گروه و در  
بخش Game Design Ideas) به اشتراک گذاشته و در صورت  
امکان، وجود ضرورت، نیاز، زمان کافی برای تحویل پروژه و صلاح  
دیدن طراح بازی، برای بازی در نظر گرفته شوند.

تا حد امکان، انعطاف‌پذیر بودن نقش هر یک از اعضای گروه در  
نظر گرفته شده است، اما احتمال آن وجود دارد که در ادامه  
روند ساخت، نیاز به تغییر رویه به وجود آید و یا حتی تخصیص  
وقت بیشتر به یک رکن بازی (توسعه، طراحی یا هنر) تبدیل به  
اولویت جدیدی شود؛ سعی می‌شود تا حد امکان، هر تیم به  
انجام وظایف تعیین شده خود بپردازد اما در شرایط ضروری،

ممکن است که این موضوع تغییر کند که این اتفاق درک عمیق  
تمامی اعضای پروژه را می‌طلبد.

## -شرح روند بازی و گیم پلی (Gameplay)-

روند کل بازی در چند مرحله خلاصه می شود:

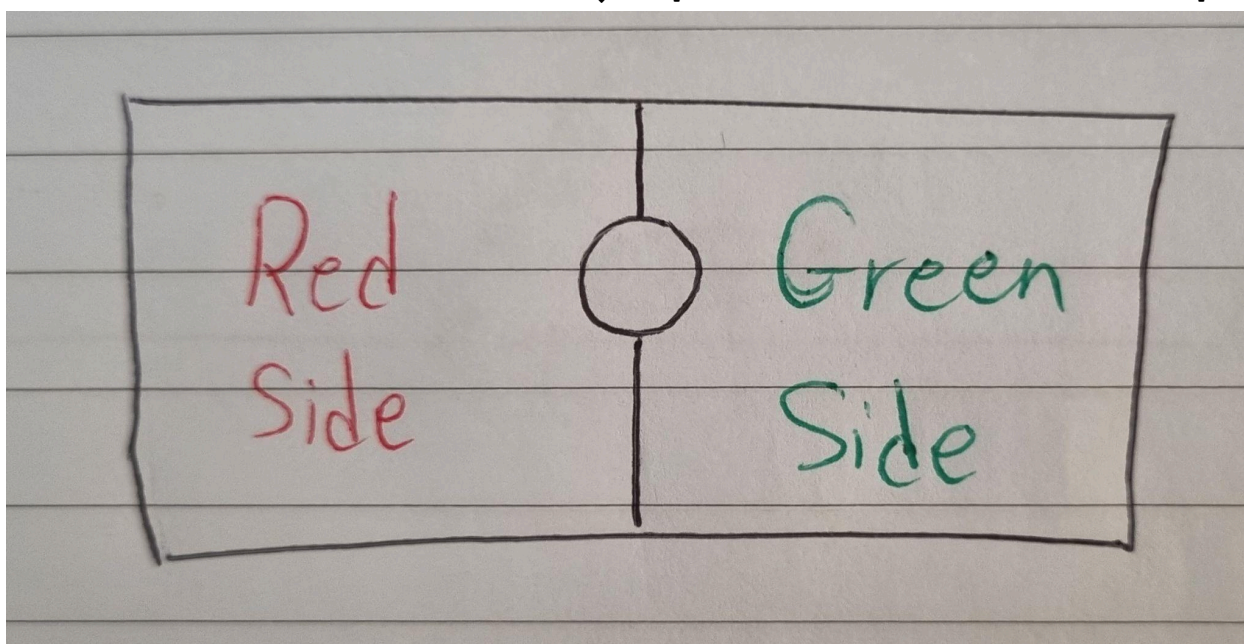
- ۱- نوشتن کد ربات ها و نبرد توسط هر تیم برنامه نویس در مدت زمان محدود (حدود ۱ ساعت)
- ۲- اجرای کد هر دو تیم بعد از اتمام زمان کدنویسی
- ۳- مبارزه بازیکنان با یکدیگر در طی **۱ دور (Round)** بازی بر اساس کدهای نوشته شده
- ۴- تمام شدن **محدودیت زمانی دور بازی (۵ دقیقه)** یا **نابودی مطلق تمامی ربات های یک تیم**
- ۶- اتمام بازی و اعلام تیم برنده



## -ویژگی‌های کلیدی‌ترین عناصر گیم‌پلی بازی-

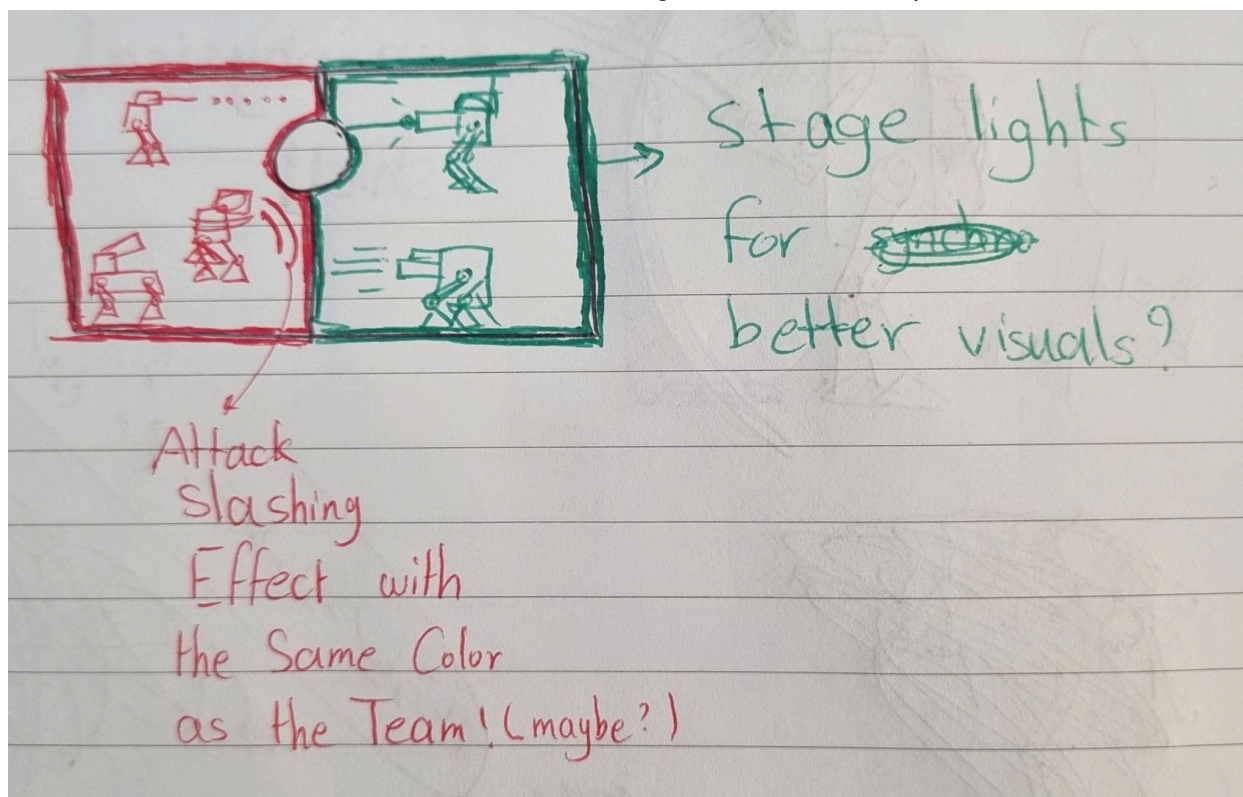
### میدان نبرد:

- میدان نبرد (Arena) بازی از نظر ابعاد، شبیه به یک زمین فوتبال می‌باشد. در هر کدام از طرفین میدان نبرد، تیم‌های ما مستقر می‌شوند و شروع به پیکار با یکدیگر می‌کنند. رنگ تیم راست **سبز** بوده و رنگی تیم چپ **قرمز** می‌باشد.

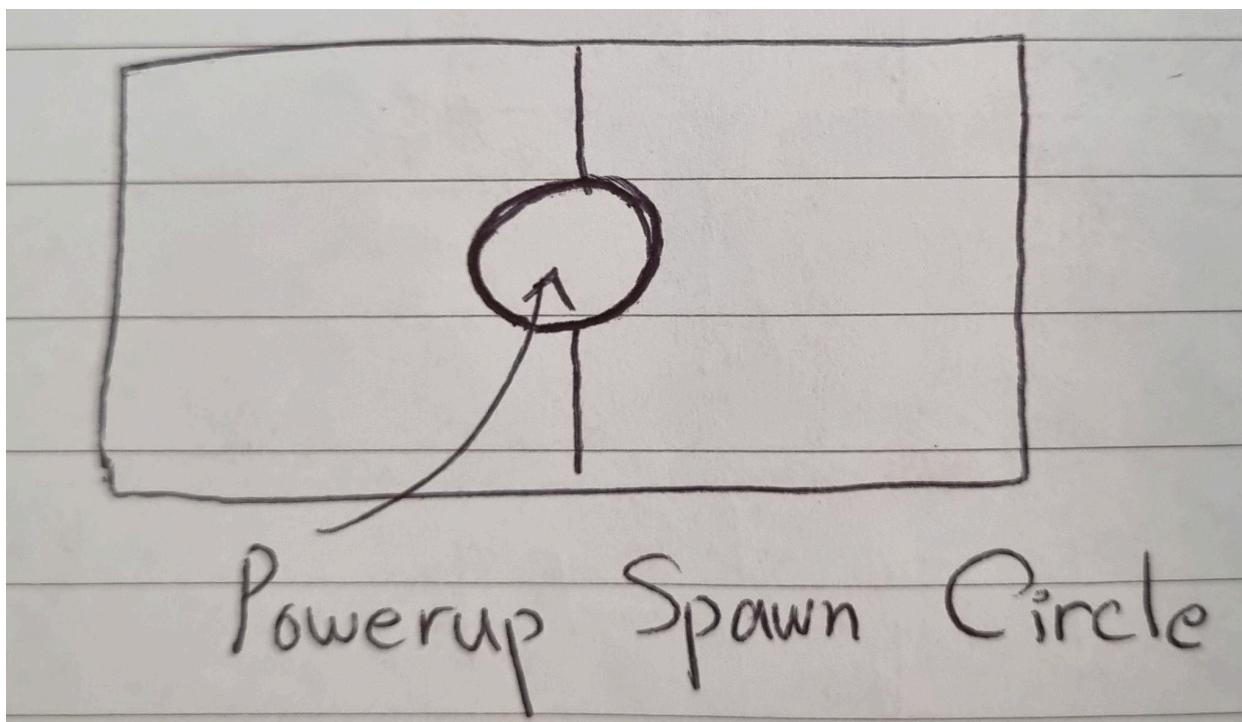


- چرا از رنگ سبز و قرمز استفاده می‌کنیم؟ به دو دلیل: ۱- رنگ سبز و قرمز در اصل دو رنگ مخالف یکدیگر می‌باشند، نه آبی و قرمز که بیشتر مرسوم است! ۲- می‌خواهیم تا تنوعی در ترکیب رنگ تیم‌های رقیب ایجاد کنیم و با این یک حرکت کوچک ولی هوشمندانه، بازی خود را منحصر به فرد می‌کنیم.
- تیم قرمز و سبز را سعی می‌کنیم تا جای ممکن، عناصر هم‌رنگ خودشان را به آنها تزریق کنیم (ربات‌های هم‌رنگ

تیم، تم و زمینه مناسب و هم‌رنگ هر تیم در هر طرف میدان  
و حتی شاید تیر و پرتابه‌های هم‌رنگ!

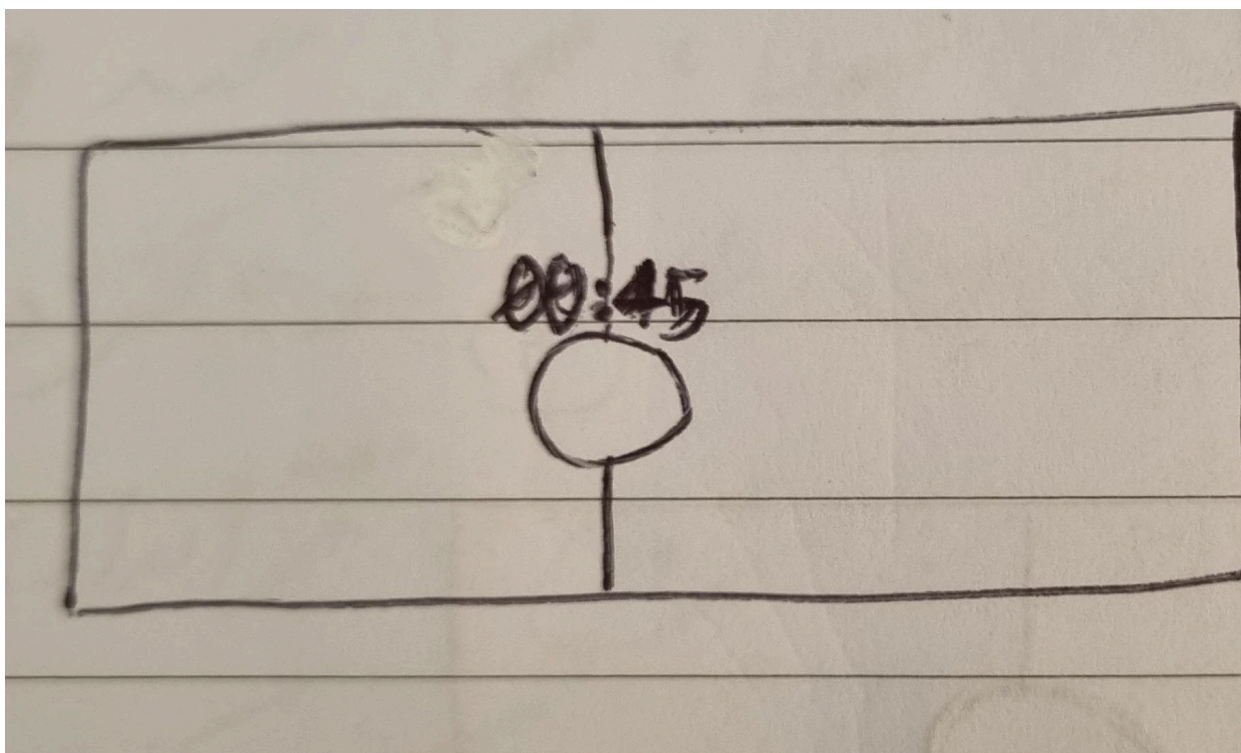


- دقیقا در وسط زمین، یک محوطه دایره‌ای وجود دارد که پس از گذشته ۲ دقیقه و ۳۰ ثانیه از شروع **دور بازی**، یک **آیتم قدرتی (Powerup Item)** درون آن ظاهر می‌شود/قرار می‌گیرد (Spawn/Deploy).

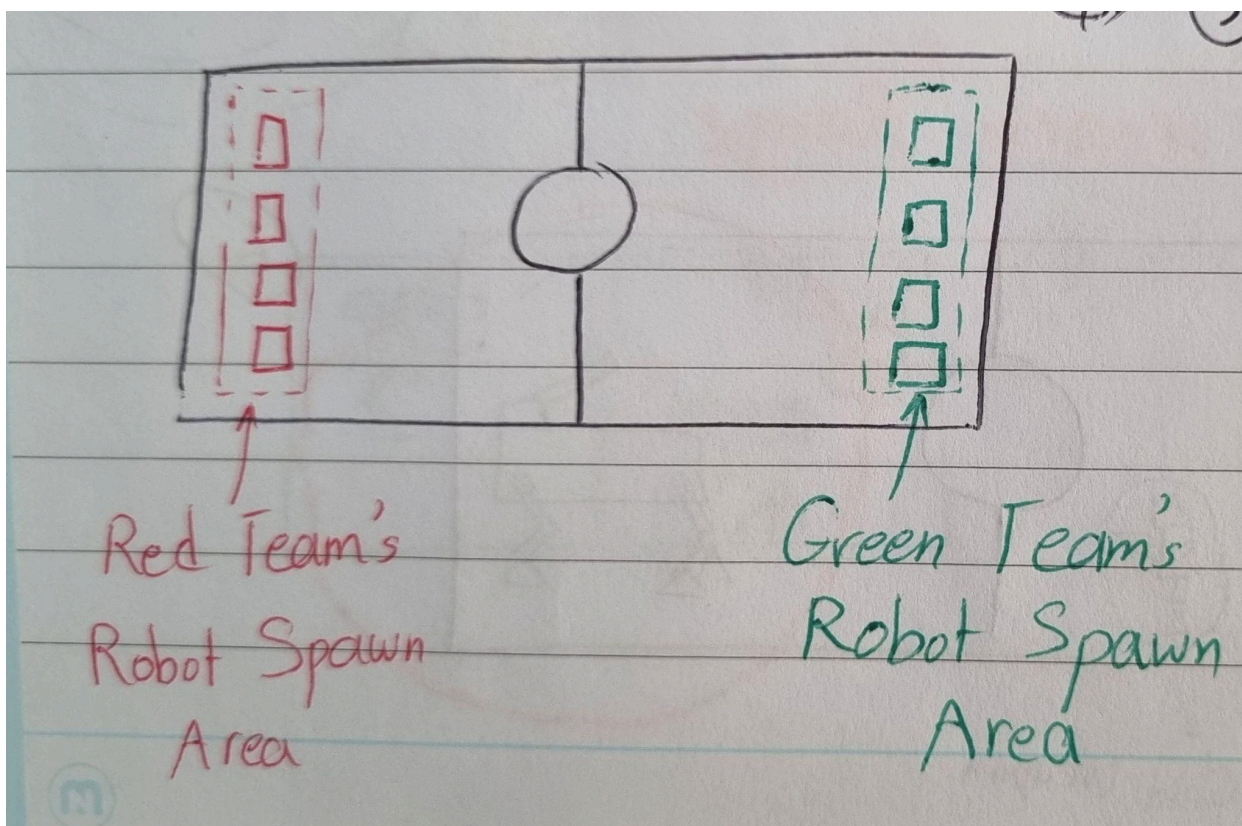


- هر رباتی که این **آیتم قدرتی** را بتواند بردارد، آسیب (Damage) آن ربات ۲ برابر می شود.
- **رباتی که آیتم قدرتی را برمی دارد،** هاله ای نارنجی یا زردرنگ اطرافش قرار می گیرد که نشان دهنده قدرتمند شدن آن ربات است.
- با شروع **دور بازی**، یک تایمر درون دایره قرار می گیرد که از ۴۵ ثانیه شروع شده و تا ۰ ثانیه را می شمرد؛ این زمان سنج در اصل به هر دو بازیکن، زمان باقی مانده تا ظاهر شدن **آیتم قدرتی** را نشان می دهد.





- با شروع بازی، ربات‌های بازیکنان به صورت ستونی در میدان  
نبرد ظاهر می‌شوند. ظاهر شدن ربات‌ها **در ابتدای دور** به  
صورت خودکار بوده و نیازی به کدنویسی توسط  
مسابقه‌دهنده نخواهد داشت.



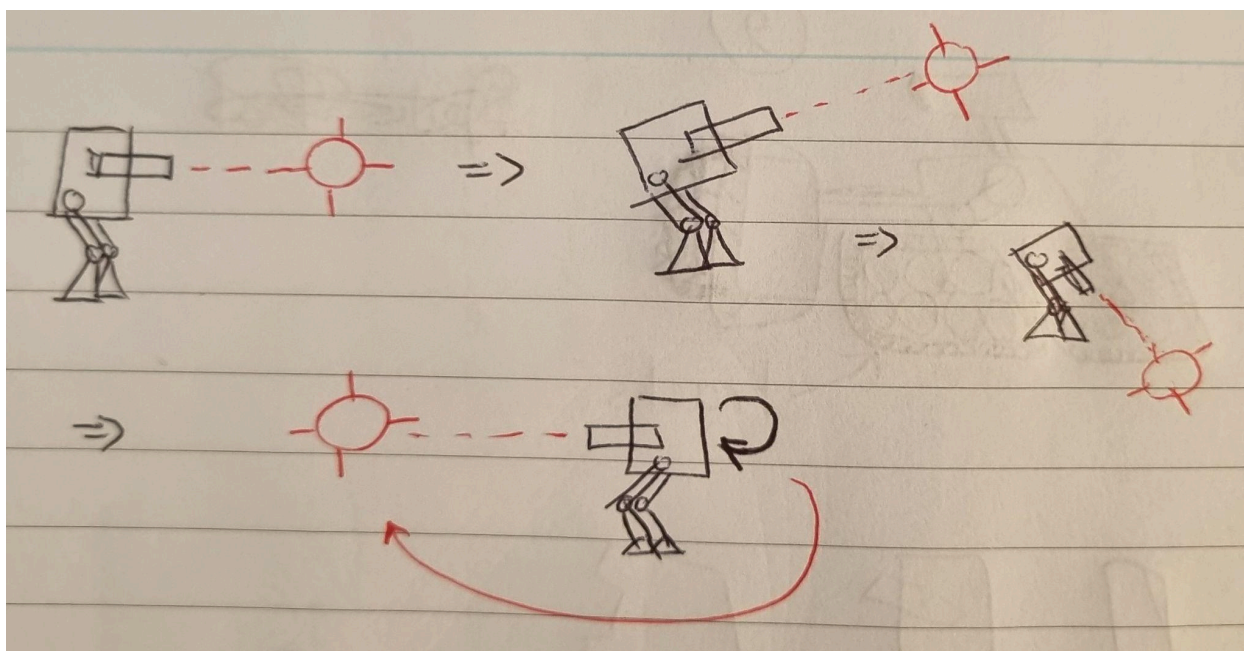
## ربات‌ها:

### تعریف ما از «ربات» در این بازی چیست؟

[ربات در این بازی، به ماشینی مکانیزه و قدم‌زن گفته می‌شود که شباهت زیادی به انسان نداشته و بیشتر مانند تانکی پادار می‌باشد.]

- در بازی، **۶ نوع** ربات قابلیت ظاهر شدن/قرار گرفتن را دارند و هر تیم نیز فقط می‌تواند با **۴ عدد** ربات مسابقه دهد؛ مثلاً بازیکن می‌تواند تصمیم بگیرد که هر **۴ عدد** ربات خود را از ربات **نوع ۱** انتخاب کند، یا اینکه **۲ عدد** ربات از **نوع ۳**، **۱ عدد**

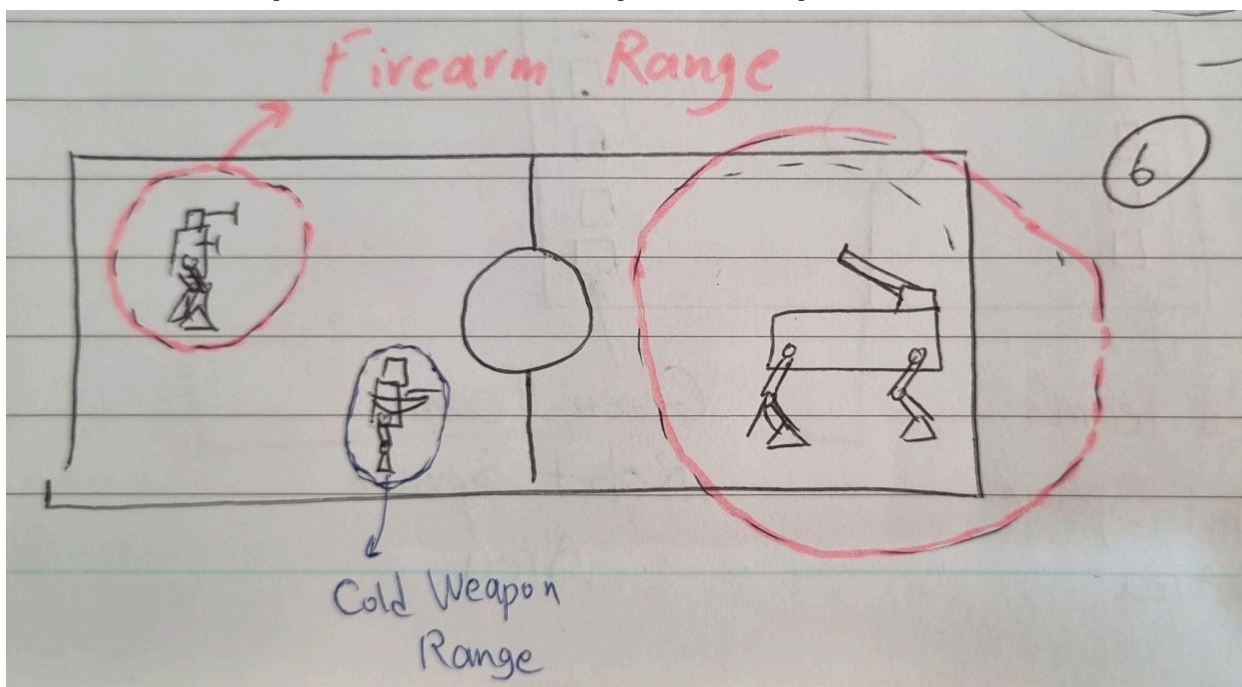
- ربات از نوع ۵ و ۱ عدد ربات از نوع ۶ انتخاب کند و یا هر ترکیب ممکن دیگری (حق انتخاب به عهده بازیکن می باشد).
- هر ربات می تواند حرکت کند (ربات ساکن در حالت معمول نداریم).
- هر ربات یک سلاح داشته و می تواند با آن حمله کند. این سلاح می تواند سرد (Cold Weapon) یا گرم (Firearm) باشد.
- انیمیشن حمله ربات ها باید به صورتی باشد که یا سلاح ربات (تیربار آن) چرخش داشته باشد و یا بالاته ربات متناسب با هدف اش تغییر جهت دهد. اگر هم هدف مورد نظر از یک دامنه به خصوصی خارج شد (مثلا پشت ربات ما قرار گرفت)، کل ربات باید به صورت عمودی (Vertically) معکوس شود (مانند ورق زدن صفحه کتاب).



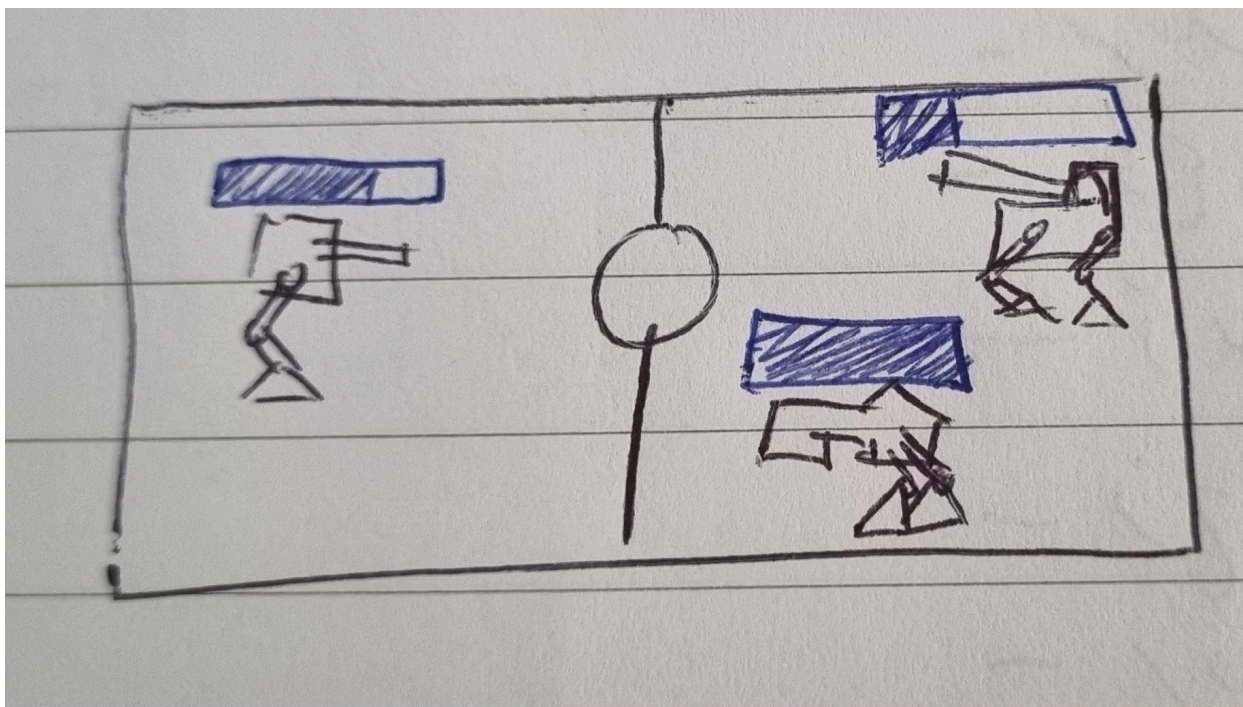
- هر ربات دارای برد سلاح (Weapon Range) خاصی می باشد که به شکل دایره ای با مرکزیت خود ربات تعریف می شود: سلاح های سرد، برد بسیار کمتری نسبت به



**سلاح‌های گرم** داشته (مجبور هستیم تا نزدیک‌تر از حد معمول به دشمن شویم تا بتوانیم به آنها حمله کنیم).



- هر ربات دارای **نوار سلامتی (Health Bar)** است که در طول کل بازی به هر دو تیم نشان داده می‌شود.



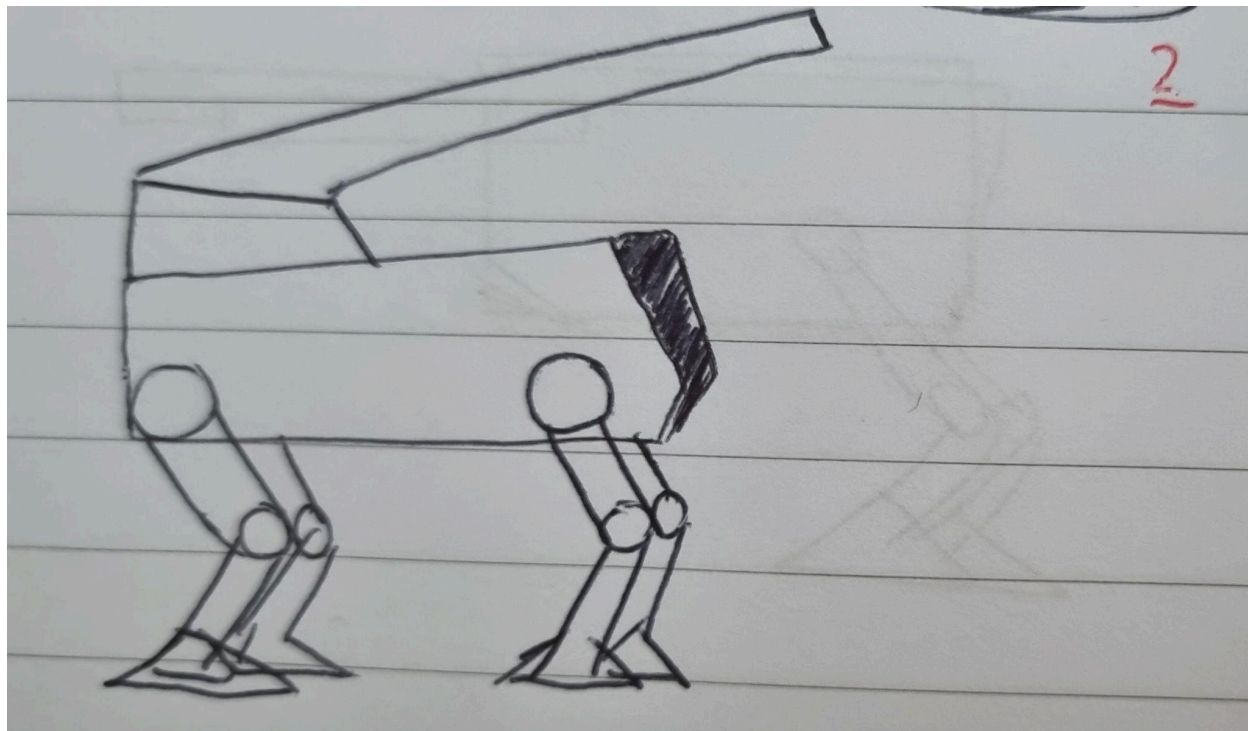
- رنگ نوار سلامتی حتما باید آبی‌رنگ یا طیفی از آبی باشد (از استفاده از رنگ قرمز یا سبز که معمولا در بازی‌های دیگر استفاده می‌شود جدا خودداری کنید!).
- با به اتمام رسیدن نوار سلامتی، ربات ما باید نابود شود و دیگر قابل استفاده نباشد (اگر از زمین بازی به سادگی حذف و ناپدید شود نیز در برهه فعلی، قابل قبول است).
- هر ربات دارای **سرعت حرکت (Movement Speed)** مخصوص به خود می‌باشد.
- هر ربات دارای **نرخ آتش (Fire Rate)** یا **نرخ حمله (Attack Rate)** منحصر به فرد می‌باشد.
- ربات‌ها می‌توانند به صورت تصادفی (یا غیرتصادفی) به یاران خود خسارت وارد کنند (تیر به خودی یا Friendly Fire در بازی فعال است).
- ربات‌ها می‌توانند با یکدیگر برخورد کنند و از لحاظ فیزیکی با هم اصابت کنند (Collision is enabled).

## انواع ربات‌ها:

دقت شود که برای تعیین حدودی مقادیر، از کلیدواژه‌های **بسیار کم، کم، متوسط، زیاد و بسیار زیاد** استفاده شده است. امکان آن وجود دارد که در نسخه‌های بعدی سند، به جای این کلیدواژه‌ها، از عدد و مقادیر واقعی استفاده شود.

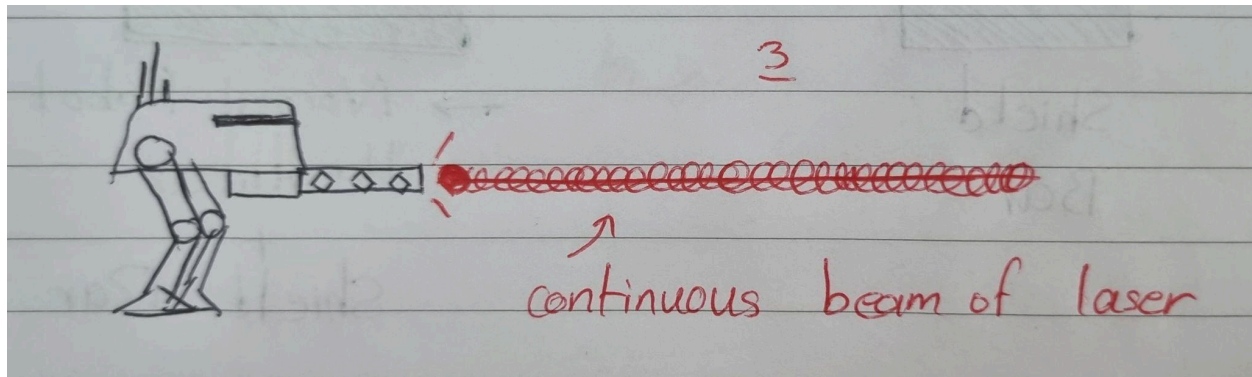
**ربات خمپاره‌انداز (Mortar Bot):**





- **سلامتی** آن زیاد است.
- **سرعت حرکت** بسیار کمی دارد.
- **سلاح گرم** دارد.
- آسیب زیادی وارد می‌کند؛ حملات این ربات یک منطقه دایره‌ای کوچک را در بر می‌گیرد و هر رباتی که در آن منطقه باشد، خسارت می‌بیند (Area of Effect Damage Type).
- **نرخ آتش** بسیار کمی دارد.
- می‌تواند از فاصله خیلی بیشتر نسبت به ربات‌های دیگر شلیک کند (**برد سلاح** بسیار زیاد دارد).
- ابعاد بسیار بزرگی دارد.
- توانایی شلیک به هدف‌های در نزدیکی خود را ندارد و باید با آنها فاصله داشته باشد تا به آنها حمله کند.

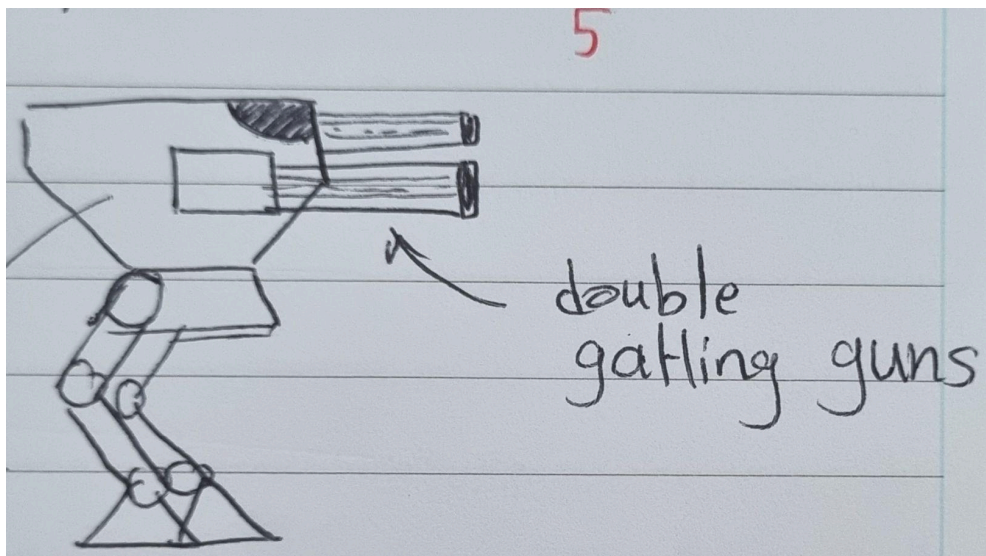
## ربات لیزری (Laser Bot):



- **سلامتی آن متوسط** است.
- **سرعت حرکت کمی** دارد.
- **سلاح گرم** دارد.
- هنگام حمله، یک پرتو قدرتمند پیوسته لیزری شلیک می‌کند (آسیب بسیار زیادی وارد می‌کند).
- این پرتو به مدت ۲ یا ۳ ثانیه به شکل ادامه‌دار شلیک می‌شود.
- در هنگام شلیک کردن، نیم‌تنه ربات (تیربار ربات) قفل شده و دیگر نمی‌تواند در خط یا جهت دیگری شلیک کند؛ به قولی اگر یک مختصات را مورد هدف قرار دهد، تا پایان یک بار حمله نمی‌تواند به مختصات دیگری حمله کند. روند حمله این ربات به این صورت می‌باشد: ۱- دریافت مختصات هدف ۲- قرار گرفتن در **برد سلاح** کافی برای حمله ۳- شارژ شدن لیزر ۴- شلیک لیزر به دقیقاً همان مختصات به مدت دو یا سه ثانیه ۵- پایان حمله و خنک شدن لیزر ۶- دریافت مختصات بعدی ۷- تکرار حلقه
- برای حمله کردن، مدتی نسبتاً کوتاه نیاز دارد تا سلاحش شارژ (Charge) شود.

- بعد از پایان یک بار حمله نیز باید مدت زمان کوتاهی بگذرد تا سلاحش خنک (Cooldown) شده و دوباره قابل استفاده شود.
- **نرخ آتش کمی** دارد.
- **برد سلاح متوسطی** دارد.
- ابعاد بزرگی دارد.
- لیزر این ربات خاصیت نفوذکننده (Piercing) داشته و از ربات حریف و حتی ربات خودی گذر کرده، به آنها نفوذ کرده و به تمامی آنها خسارت وارد کند.

### ربات رگباری (Gatling Bot):

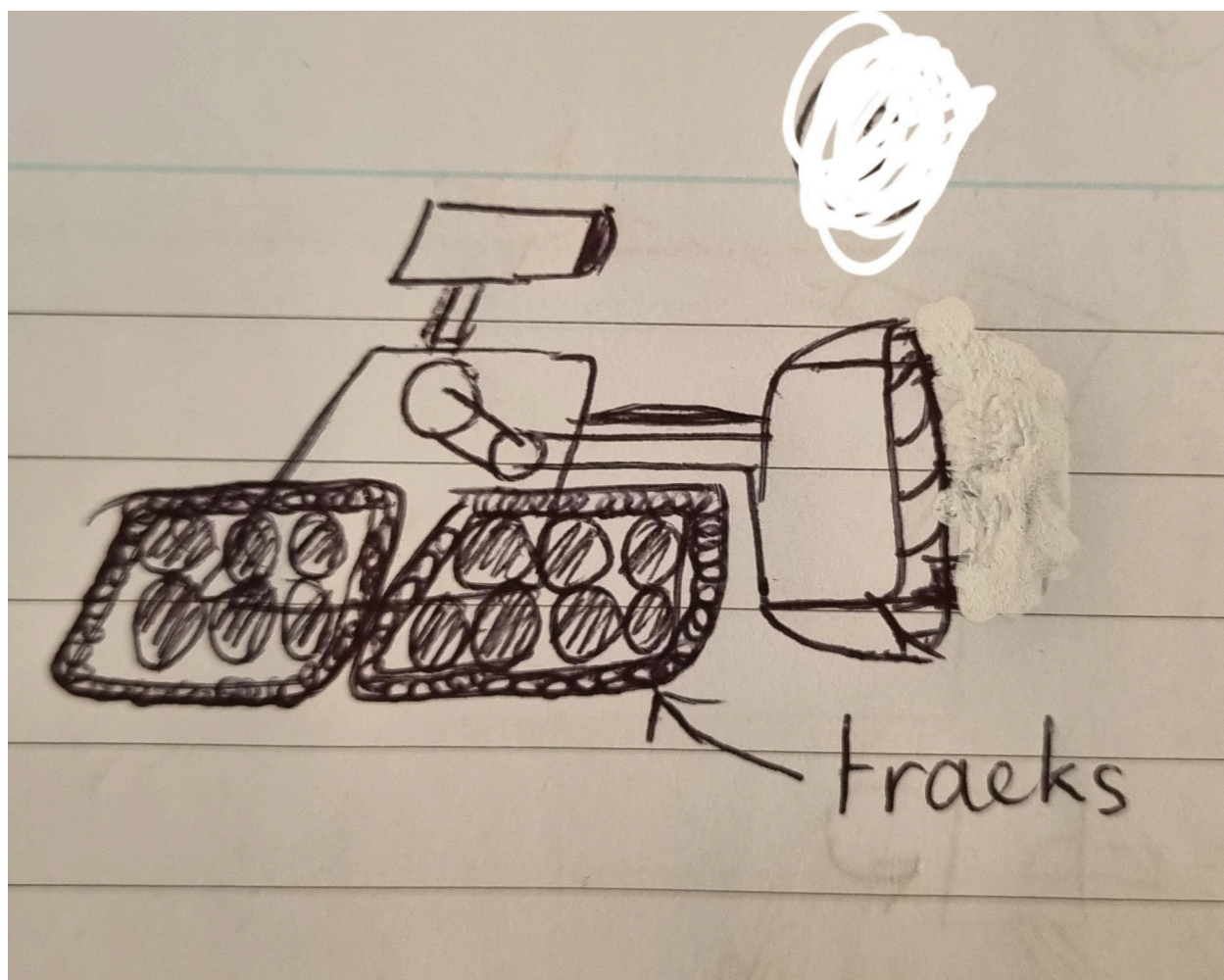


- **سلامتی کمی** دارد.
- **سرعت حرکت بسیار زیادی** دارد.
- **سلاح گرم** دارد.
- آسیب کمی وارد می‌کند.
- **نرخ آتش بسیار زیادی** دارد.



- **برد سلاح کمی** دارد (کمتر از دیگر ربات‌های دارای **سلاح گرم**، بیشتر از ربات خردکن و برد برابر با ربات شوالیه)
- ابعاد **کوچکی (کمی)** دارد.
- می‌تواند در حین حرکت تیراندازی کند.

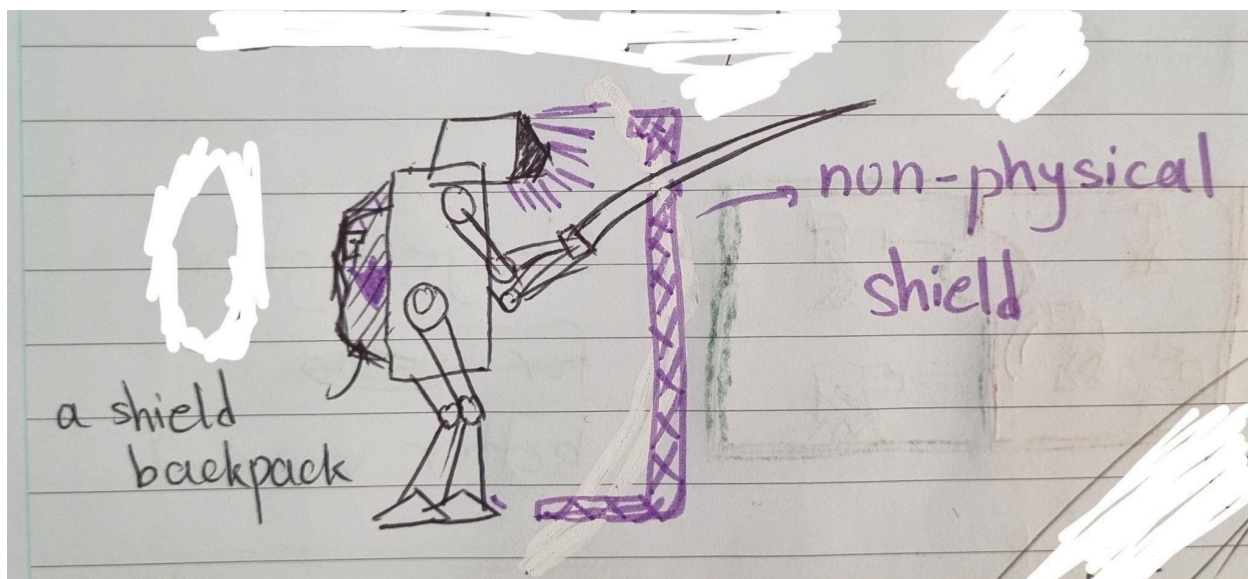
### ربات خردکن (Grinder Bot):



- **سلامتی بسیار بالایی** دارد.
- **سرعت حرکت متوسطی** دارد.
- **سلاح سرد** دارد.
- آسیب **متوسط** وارد می‌کند.

- **نرخ حمله** به شکلی است که به محض چسبیدن به هدف، سریع شروع به خرد کردن و وارد کردن آسیب می‌کند و می‌تواند در مدت زمان کمی، آسیب سرسام‌آوری وارد کند.
- **برد سلاح** بسیار کمی دارد (عملاً برد سلاح این ربات عدد صفر یا یک است).
- ابعاد بزرگی دارد.

### ربات شوالیه (Knight Bot):



- **سلامتی** متوسطی دارد.
- **سرعت حرکت** زیادی دارد.
- آسیب متوسط وارد می‌کند.
- **سلاح سرد** دارد.
- **نرخ حمله** زیادی دارد.
- **برد سلاح** کمی دارد (می‌تواند با شمشیر بلندش از فاصله دورتر از ربات خردکن به دشمن حمله کند).

- ابعاد متوسطی دارد.
- همواره سپری از جنس انرژی در جلوی خود دارد که به صورت غیرفیزیکی است (ربات‌ها می‌توانند از این سپر عبور کنند، اگر به اندازه کافی به این ربات نزدیک شوند).
- هنگام دریافت کردن خسارت از جلو، سپر انرژی از کار می‌افتد و بعد از گذشت مدت زمانی کم یا متوسط، دوباره فعال می‌شود (سپر قابل شارژ است).
- شارژ سپر به صورت خودکار باید پیاده‌سازی شود و خود بازیکن هیچ کدی در این باره نباید بنویسد.