

# Projektarbeit GameDev: Meilenstein 4

Projekt «Rogue-like mech game»

Jonathan Anderfuhren

## Git und Webseite

Das Projekt befindet sich auf Github hier:

[https://github.com/castawaywars/rogue-like\\_mech\\_game\\_2d](https://github.com/castawaywars/rogue-like_mech_game_2d)

Online findet sich das Programm hier:

[https://castawaywars.github.io/rogue-like\\_mech\\_game\\_2d/](https://castawaywars.github.io/rogue-like_mech_game_2d/)

Diese Orte sind seit dem letzten Meilenstein unverändert, ich verlinke sie aber nochmals, um das Finden der relevanten Dinge zu vereinfachen.

## Unterschiedliche Aktionsmodi

### Grundideen

1. Das Ziel der Gegner ist, die Mechs zu besiegen.
2. Die Gegner folgen immer einer gemeinsamen Strategie. Zu jedem Zeitpunkt gibt es genau eine Strategie, der alle Gegner folgen.
3. Solange es Spawner hat, wird die Anzahl Gegner hoffentlich steigen.
4. Es ist einfacher, die Mechs zu überwinden, wenn mehr Gegner verfügbar sind.

Aus 3 und 4 folgt:

5. Wenn nur wenige Gegner vorhanden sind, zurückziehen, wenn hingegen viele Gegner vorhanden sind, angreifen.
6. Wenn keine Spawner da sind, ist ein Rückzug sinnlos, daher angreifen.

### Spezifisches

Beim Rückzug gilt es, Schusslinien unbedingt zu vermeiden, ausser der Gegner befindet sich in einer Schusslinie und die einzigen Bewegungs-Optionen sind auch in einer Schusslinie. Daher ist die Vermeidung von Schusslinien beim Rückzug absolut, wenn nicht die erwähnte Ausnahme eintritt.

Beim Angriff hingegen kann es notwendig sein, in eine Schusslinie zu laufen. Deshalb ist beim Angriff die Vermeidung der Schusslinien nur als Art Präferenz einprogrammiert, kann aber problemlos ignoriert werden.

Es gibt zwei unterschiedliche Angriffs-Modi: Der gewöhnliche Angriff versucht, ausserhalb der Schusslinien zu bleiben. Der unvorsichtige Angriff hingegen ignoriert Schusslinien komplett. Der unvorsichtige Angriff kommt zum Zug, wenn keine Spawner auf dem Feld sind, da dann kein Zögern akzeptiert werden kann, sowie wenn es so viele Gegner auf dem Feld hat, dass Verluste toleriert werden können.

Die Berechnung, welcher Modus verwendet wird, wenn es Spawner hat, bezieht sich auf die Grösse des Spielfeldes und die aktuelle Anzahl Gegner.

Die spezifischen Anzahlen sind abhängig von der Kartengrösse, da die Werte als Prozentzahlen agieren.

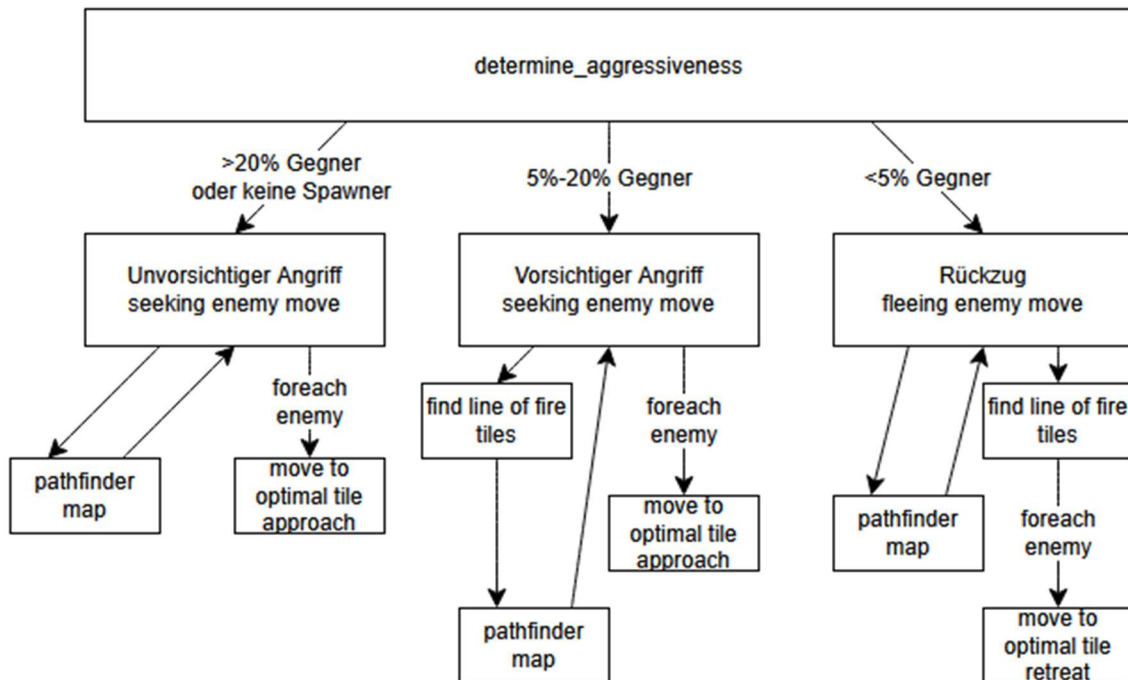
Wenn die Gegner mehr als 20% der Felder der Karte einnehmen, wird unvorsichtiger Angriff eingesetzt.

Bei 5% oder weniger wird auf Rückzug geschaltet.

Wenn die Gegner zwischen 5% und 20% der Karte einnehmen, kommt der gewöhnliche Angriff zum Zug.

## Sonstiges

### Funktions-Struktur



`Determine_aggressiveness()` ist die Kopf-Funktion der KI. Diese Funktion wird aufgerufen, wenn der Zeitpunkt gekommen ist, alle Feinde zu bewegen.

`Determine_aggressiveness` erarbeitet, welche Strategie verwendet werden soll, und ruft danach die korrekte Funktion auf die korrekte Art auf.

`Seeking_enemy_move()` wird für Angriffe verwendet, `fleeing_enemy_move` für Rückzug.

`Seeking_enemy_move` benötigt als Argument eine pathfinder-Karte, `fleeing_enemy_move` ebenfalls, aber letztere benötigt auch noch explizit die Schusslinien.

Diese zwei Funktionen finden dann für jeden Gegner alle Bewegungs-Optionen und übergeben den Gegner und die Bewegungs-Optionen dann einer der beiden `move-to-optimal-tile`-Funktionen, die dann den Gegner ideal bewegt.

Die pathfinder-Karte wird von einer spezifischen Funktion erstellt. Diese Funktion erstellt eine Karte mit Distanz-Werten. Dieser Funktion muss ein Array mit den Schusslinien übergeben werden. Will man, dass die pathfinder-Karte keine Rücksicht auf Schusslinien nimmt, übergibt man ein leeres Array.

Für einen unvorsichtigen Angriff erhält die `seeking_enemy_move`-Funktion eine pathfinder-Karte, die mit leerem Array für Schusslinien erstellt wurde.

Für einen gewöhnlichen Angriff erhält die `seeking_enemy_move`-Funktion eine pathfinder-Karte, die die Schusslinien miteinbezieht.

Für einen Rückzug erhält die `fleeing_enemy_move`-Funktion eine Schusslinienlose pathfinder-Karte sowie die Schusslinien separat, da, um die Schusslinien ideal vermeiden zu können, die Funktion die Schusslinien kennen muss, während sie Bewegungsoptionen sucht.

## Grafiken

Es wurden auch noch ein paar Grafiken hinzugefügt, um das Spiel ansprechender zu gestalten.



Flammenwerfer-Muniton



Reperatur/Heilungs-Paket



Feind-Spawner



Feind

## Herausforderungen und Lösungen

### Schusslinienerkennung: Duplikate vermeiden

Die Schusslinien der zwei Mechs im Mehrspielermodus können sich überschneiden. Wenn ein Mech Flammenwerfer-Muniton hat, überschneiden sich die Schusslinie der Hauptwaffe des Mechs mit der Schusslinie des Flammenwerfers.

Wenn alle diese Schusslinien zusammengezählt werden, führt das zu unerwünschtem Verhalten der Gegner.

Daher suchen die Funktionen, die später mit den Schusslinien-Ergebnissen arbeiten, nur danach, ob ein Feld in irgendeiner Schusslinie vorkommt, und überprüft nicht, in wie vielen Schusslinien das Feld abgedeckt ist.

### Vermeiden, in Schusslinien zu laufen

Um zu einem Mech zu finden, muss ein Pathfinding-Algorithmus laufen.

Das Resultat des Pathfinding- Algorithmus wird dann an eine andere Funktion übergeben, die die Gegner entsprechend bewegt.

Gegner versuchen meistens, Schusslinien der Mechs zu vermeiden, da sie dort sofort abgeschossen werden könnten.

Wird die Vermeidung von Schusslinien erst in der Bewegungsfunktion eingebunden, können die Gegner nicht mehr erkennen, wenn der einzige Weg, zu den Mechs zu kommen, durch die Schusslinie der Mechs führt, und können in solchen Situationen nicht mehr zu den Mechs gelangen.

Daher musste der Einfluss der Schusslinien bereits im Pathfinding eingebunden werden.

Dies wurde gemacht, indem die Bewegungskosten von Feldern, die in einer Schusslinie liegen, um eins erhöht werden. Dadurch vermeiden die Gegner ganz natürlich die Schusslinie, ohne blockiert zu werden, und können, wenn nötig, durch die Schusslinie marschieren.

### **Rückzug, auch wenn direkt neben einem Mech**

Wenn die Strategie Rückzug lautet, ist das Ziel der Gegner, ihre Einheiten zu konservieren und später, wenn mehr Gegner gespawnt wurden, wieder anzugreifen. Die Hoffnung ist, dass mit mehr Einheiten es leichter ist, die Mechs zu überwältigen und erfolgreich anzugreifen. Daher ziehen sie sich zurück.

Es ist aber sinnlos, dass sich Gegner, die direkt neben einem Mech stehen, sich zurückziehen, da sie stattdessen einen erfolgreichen Angriff durchführen könnten.

Daher wurde die Rückzugs-Funktion so angepasst, dass, wenn ein Gegner direkt neben einem Mech steht, der Gegner den Mech angreift, statt sich zurückzuziehen.