

Semesterarbeit
Adhoc Game

Christian Gegenschatz
gegensch@student.ethz.ch

Departement Informatik
Eidgenössische Technische Hochschule Zürich
Winter 2002 / 2003

Prof. Dr. Roger Wattenhofer
Distributed Computing Group
Betreuer: Aaron Zollinger

Inhaltsverzeichnis

1	Einführung	2
1.1	Gründe und Ziele	2
1.1.1	Handelswelt	2
1.1.2	Alle gegen alle	3
1.1.3	Zweikämpfe	3
1.2	Die Entscheidung	3
2	Das Spiel	4
2.1	Kurze Spielbeschreibung	4
2.2	Das Display	4
2.2.1	Mitspielerliste	4
2.2.2	Infozeile	5
2.2.3	Nachrichtenbox	5
2.2.4	Anzeigetafel	5
2.2.5	Routen	5
2.3	Fangen	5
2.4	Was hinter den Kulissen geschieht	6
2.4.1	Aktualisierung der Mitspielerliste	6
2.4.2	Routensuche	6
2.4.3	Werfen	6
2.4.4	Balltimer	7
2.4.5	Nachladen	7
2.4.6	Parameter	7
3	Die neuen Pakete	8
3.1	Nachrichtenformat	8
3.2	Reaktion	9
4	Schlussbemerkungen	10
4.1	Probleme	10
4.1.1	Wirelesskarten	10
4.1.2	Timeout	10
4.1.3	Ballverlust oder -flaute	10
4.1.4	Adhoc-Netzwerk	11
4.2	Erweiterungen	11
4.3	Ziele erreicht?	11
4.4	Tests	12

Kapitel 1

Einführung

1.1 Gründe und Ziele

Wir alle kennen Spiele, bei denen man sich viel bewegen muss, um in einer besseren Position oder Ausgangslage zu sein. Davon hängt meistens der ganze Spielspass ab, wie beim Fussball, Fangen, Völkerball, Räuber und Polizist, Verstecken... Auf der anderen Seite macht es bei klassischen Computerspielen überhaupt keinen Sinn, sich mit dem ganzen Computer von einem Ort zum anderen zu begeben. So entstand die Idee, diese zwei Komponenten zu verbinden und ein Computerspiel zu kreieren, bei welchem man sich bewegen und verschieben muss, um überhaupt ein gewisses Level zu erreichen. Die verschiedenen mobilen Computer kommunizieren untereinander in einem Adhoc-Netzwerk, was auch ausgenutzt werden soll, d.h. das Game sollte v.a. den Multihopmechanismus gebrauchen müssen. Die letzte Anforderung war, dass jeder Rechner gleichberechtigt sein soll, es also keinen Master oder etwas in dieser Richtung geben soll, jeder Computer muss jederzeit ins Spiel eintreten oder aus ihm austreten können. Die ersten Ideen von möglichen Spielen entstanden:

1.1.1 Handelswelt

Jeder Rechner steht für ein mobiles Gebiet. Jedes dieser Gebiete kann Handel mit allen anderen betreiben. Je weiter der Handelspartner entfernt ist, desto mehr Gewinn kann man erwirtschaften. Man muss jedoch an jedes Gebiet, welches man (mit dem Handelskarren) durchquert, einen Zoll abgeben. Die Höhe dieses Zolles kann der Besitzer des Gebiets selber bestimmen. Somit hat jeder Spieler zwei Einnahmequellen: den Handel mit anderen Gebieten und die Zolleinnahmen. Mit diesem Geld kann man verschiedene Dinge kaufen:

- mehrere Handelskarren
- verbesserte beziehungsweise schnellere Handelswagen

Mögliche Erweiterungen:

- Handelsgut muss zuerst hergestellt werden, d.h. man muss zu bezahlende Arbeitskräfte haben, um überhaupt handeln zu können
- Handelspartner muss einverstanden sein, damit man mit ihm handeln kann

- Einsatz einer Armee zur Verteidigung des eigenen Gebietes und Angriff auf die Gegner
- nicht jedes Gebiet besitzt die gleichen Rohstoffe, d. h. um spezielle Waren zu kaufen oder sich verbessern zu können, müssen diese Güter durch Handel erworben werden

1.1.2 Alle gegen alle

Jeder Spieler bekommt einen Ball. Mit diesem kann er seine Mitspieler „treffen“. Gelingt dies, erhält er Punkte, wird er damit getroffen, werden Punkte abgezogen. Da in einem Adhoc-Netzwerk nicht alle Pakete ankommen, kann dies ausgenutzt werden: Wirft man einen Ball, so schickt man ein Paket. Kommt es an, hat man getroffen, sonst nicht. Versucht man einen weit entfernten Gegner zu treffen, so ist die Wahrscheinlichkeit eines Erfolges natürlich viel kleiner. Gelingt die Aktion jedoch, so wird dies auch mit mehr Punkten belohnt. Den Ball erhält jener, bei welchem er hängen geblieben ist, d.h. beim letzten erfolgreichen Hop.

1.1.3 Zweikämpfe

Vorbereitung: Es gibt eine grosse Sammlung von Angriffs- und Verteidigungskarten mit jeweils unterschiedlich hohen Werten. Tritt jemand einem Spiel bei, bekommt er zufällig je zwei Karten mit geringem Wert.

Kampf: Mit einem direkten Nachbarn kann man einen Zweikampf eingehen. Bei einem Sieg erkennt man die vom Gegner verwendeten Karten und kann im weiteren Verlauf auch diese benutzen. Zusätzlich erhält man Punkte je nachdem, ob man einen Stärkeren oder Schwächeren besiegt hat. Mit diesen Punkten kann man sich aus der Sammlung Karten mit höherem Wert erkaufen. Der Unterlegene lernt nur eine Karte des Gegners und bekommt keine Punkte. Einen Kampf muss man nicht zwingend annehmen.

Vorteil: Je mehr verschiedene Gegner man bekämpft hat, desto breiter ist die eigene Auswahl an Karten, und man kann spezifischer auf bessere Gegner reagieren.

1.2 Die Entscheidung

Die Entscheidung, welches Spiel wirklich implementiert werden sollte, war nicht einfach. Der zweite Vorschlag schien am besten für diese Semesterarbeit geeignet, da keine mühsamen Vorbereitungsarbeiten wie das Erstellen geeigneter Spielkarten (Vorschlag 3) nötig sind und man schon sehr rasch eine primitive Form des Spieles implementieren kann. Es wäre auch interessant zu wissen, wie weit die unsichere Verbindung eines Adhoc-Netzes direkt verwendbar ist. Ich entschied mich deshalb für das Game „ALLE GEGEN ALLE“.

Leider erwies sich die Annahme als falsch, dass man die fehlerhaften Verbindungen ausnützen könnte. Und zwar, weil die gesendeten Pakete zwingend ankommen müssen, oder es ganz klar sein muss, bei welchem Hop das Paket nicht weitergekommen ist, da sonst Bälle oder Punkte verloren gehen und inkonsistente Zustände vorkommen könnten.

Kapitel 2

Das Spiel

2.1 Kurze Spielbeschreibung

Jeder Mitspieler spielt gegen jeden, mit dem Ziel, möglichst viele Gegner zu treffen. Zuerst wird ein Spieler aus der Mitspielerliste ausgewählt und erhält dafür eine Auswahl Routen zurück, über welche man versuchen kann, den Spieler zu treffen. Jeder Hop funktioniert nur mit einer bestimmten Wahrscheinlichkeit, d.h. man schickt den Ball nur ab einer genug hohen Wahrscheinlichkeit weiter, sonst behält man ihn. Falls schon Bälle im Eigenbesitz sind, ist die Wahrscheinlichkeit des Weiterleitens noch viel grösser. Kommt der Ball beim Gegner an, besteht die Gelegenheit, den Ball zu fangen oder ihm auszuweichen um so dem Treffer zu entgehen. In diesem Fall wird der Ball an dem Nachbar weitergegeben, der sich am schnellsten meldet. Trifft man einen Gegner zum ersten Mal, erhält man z.B. acht Punkte, bei jedem weiteren Mal immer weniger, im Minimum aber einen Punkt. Pro Hop, den der Ball zurückgelegt hat, gibt es zusätzliche Punkte (z.B. $2^{anz.Hop}$). Wird man jedoch selber getroffen, werden wieder Punkte abgezogen.

Das Spielende ist erreicht, sobald jemand 100 Punkte erzielt hat.

In einer anderen Version des Spieles werden zu Beginn drei Fertigkeitpunkte auf Werfen, Fangen und Ausweichen verteilt. Jeweils nach zehn gewonnenen Punkten wird ein weiterer Fertigkeitpunkt vergeben. Je nach dem, wo die Punkte eingesetzt werden, bestehen Vorteile im Treffen, Abfangen von Bällen oder im Ausweichen.

2.2 Das Display

Die grafische Oberfläche besteht aus:

2.2.1 Mitspielerliste

Hier werden alle gefundenen und noch aktiven Mitspieler aufgelistet. Neben dem Namen und der Adresse wird die Punktzahl, die aktuellste Entfernung und die Anzahl der Treffer auf diesen Host aufgeführt. Um jemandem eine Nachricht, Routenanfrage oder einen Treffversuch zu schicken, muss jener Mitspieler angewählt werden. Sobald dies geschehen ist, wird sofort eine Routensuche gestartet, welche beim nochmaligen Anklicken wiederholt wird.



Abbildung 2.1: Das Display

2.2.2 Infozeile

Neben der aktuellen Punktzahl kann hier auch die momentane Anzahl von Bällen im eigenen Besitz entnommen werden.

2.2.3 Nachrichtenbox

Wie beim originalen Instant Messenger (siehe Kapitel 3) kann man eigene Nachrichten und Kommentare über Broadcast oder an einzelne Mitspieler direkt senden.

2.2.4 Anzeigetafel

Alle wichtigen Informationen über den Verlauf des Spieles werden hier automatisch angezeigt, wie zum Beispiel allfällige eigene Treffer, oder, falls man selbst getroffen worden ist, wer den Ball abgefangen hat oder ob das Ziel ausgewichen ist. Auch normale Nachrichten werden wie früher hier angezeigt.

2.2.5 Routen

Alle gefundenen Routen werden hier mit den einzelnen Stationen namentlich aufgelistet. Sobald man eine Route markiert hat, wird ein Schuss auf den Gegner abgefeuert. Um mit der gleichen Route noch einmal zu werfen, muss diese Route nochmals ausgewählt werden.

2.3 Fangen

Erscheint ein Fangen-Button auf dem Bildschirm, bleiben zwei Sekunden Zeit, um auf diesen zu klicken. Je schneller dies geschieht, desto grösser ist die Wahrscheinlichkeit, den Ball abzufangen oder ihm auszuweichen kann.



Abbildung 2.2: Fangen-Button

2.4 Was hinter den Kulissen geschieht

2.4.1 Aktualisierung der Mitspielerliste

Alle paar Sekunden wird ein Ping gesendet, um die Nachbarn zu finden und die anderen Mitspieler auf sich aufmerksam zu machen. So ist sichergestellt, dass man zu Beginn zuerst alle Nachbarn entdeckt. Jedes Paket, das der Computer empfängt oder weiterleitet, wird nach noch nicht bekannten Adressen durchsucht. Findet man eine solche, wird ein „WHOAREYOUPacket“ an die entsprechende Adresse geschickt. Zu jeder bekannten Adresse wird der Zeitpunkt des letzten „Lebenszeichens“ gespeichert. Hat man schon zu lange nichts mehr von einem bekannten Host gehört, wird dieser aus der Mitspielerliste entfernt und in einer separaten Liste aufbewahrt (UpdateAllHost). Falls man wieder etwas von diesem Mitspieler empfängt, wird er reaktiviert und wieder in die aktive Liste übertragen.

2.4.2 Routensuche

Da man zu jedem Gegner die vermutete Entfernung kennt, kann man die TTL (Time-ToLive) zu Beginn einer Routensuche etwas grösser setzen (z.B. $1.5 * Entfernung + 2$, damit auch noch etwas längere Routen zur Auswahl stehen) und sich somit die anfänglichen iterativen Vergrößerungen der TTL sparen. Damit bei einer Routensuche mehrere verschiedene zur Verfügung stehen, kann nicht wie bis anhin jedes Paket mit bekanntem (Sender, Id)-Paar ignoriert werden. Vielmehr muss jede akzeptierte Route (d.h. Routenanfragen, die man weiterleitet) zusammen mit dem Sender, der Paket-ID, einem Zeitstempel und den übrigen Routen gespeichert werden. Empfängt man einen weiteren Request dieses Anfragevorgangs, so kann er nur weitergeleitet werden, falls er sich genug stark von allen anderen unterscheidet. D.h. es werden die übereinstimmenden Zwischenstationen gezählt und durch die mittlere Routenlänge dividiert.

Damit wird sichergestellt, dass nicht das ganze Netzwerk geflutet wird, sondern mehrere verschiedene Routen für das gleiche Ziel zur Verfügung stehen.

Von allen zurückgesendeten Routen wird jeweils die kürzeste als aktuelle Entfernung angenommen.

2.4.3 Werfen

Versucht man jemanden zu treffen, wird ein Wurfpaket der selbstgewählten Route entlang geschickt. Nach jedem Hop wird mit kurzer Bestätigung die vorhergehende Station benachrichtigt, dass der Wurf auch wirklich angekommen ist. Bleibt diese aus, wird der Wurf noch zweimal versucht. Falls immer noch keine Bestätigung zurückkommt, behandelt man den Ball, als wäre er abgefangen worden.

Kommt der Wurf beim eigentlichen Empfänger an, wird eine „grosse Bestätigung“ der

ganzen Route entlang geschickt, um den Sender über den Treffer zu informieren, damit dieser die verdienten Punkte erhält. Die Bestätigung erfolgt jeweils hopweise.

Diese zugegeben etwas komplizierte Vorgehensweise ist nötig, um möglichst sicher zu gehen, dass kein Ball verloren geht, denn das würde von niemandem direkt bemerkt werden. Wenn man nur auf eine grosse Bestätigung wartet, wüsste man nie, wo und wann der Wurf gescheitert ist. Auch wäre bei den einzelnen Zwischenstationen jeweils nötig, genau zu speichern, wie sie bei welchem Paket reagiert haben (weiterleiten, abfangen), damit das erste Paket wahrheitsgetreu wiederholt werden kann.

2.4.4 Balltimer

Damit man nicht zu lange einen Ball festhält und somit das ganze Spiel unter Umständen bremst oder aufhält, muss ein Ball, falls er noch nicht weitergeworfen wurde, nach 20 Sekunden fallengelassen werden. Der Mechanismus ist derselbe, wie wenn man einem Ball ausweichen würde: Der schnellste der direkten Nachbarn bekommt ihn. Falls man keine Nachbarn besitzt, kann man ihn selber wieder aufheben.

2.4.5 Nachladen

Damit nicht eine ganze Salve von Bällen auf einen Mitspieler abgefeuert werden kann, wird nach einem Schussversuch der nächste Ball für kurze Zeit zurückgehalten, um nicht sofort abgeschossen zu werden.

2.4.6 Parameter

Im Spiel werden verschiedene Parameter verwendet, welche das Game beeinflussen:

Basiswahrscheinlichkeiten:

Handicap pro Ballbesitz(2.1) = 0.1

Die Wahrscheinlichkeiten des Abfangens oder Ausweichens werden nur durch die Reaktionszeit des Spielers bestimmt (2.3)

Wartezeiten:

Balltimer(2.4.4) = 20s

UpdateAllHost(2.4.1) = 6s

Hoptimeout = 2 * 0.2s

Nachladen(2.4.5) = 1s

Abfangzeit(2.3) = 2s

Allgemein:

Anzahl Versuche einen Hop zu erreichen(2.4.3) = 3

Bälle zu Beginn(4.1.3) = 2

maximale akzeptierte Routenübereinstimmung(2.4.2) = 30%

Spielziel(2.1) = 100 Punkte

Kapitel 3

Die neuen Pakete

Das Spiel baut auf dem Instant Messenger auf, einem Programm, mit welchem Nachrichten im Adhoc-Modus auch an entfernte Stationen über den Multihop-Mechanismus gesendet und empfangen werden können. Um aber all diese neuen Vorgänge realisieren zu können, braucht es natürlich eine Vielzahl von neuen Paketen:

3.1 Nachrichtenformat

Nachrichtentyp	Nachrichtenformat (Feldgrösse in Bytes)
WB	type(1) messageID(2) sender(2) receiver(2) points(1) route length(1) route index(1) route (variable)
WBACK	type(1) messageID(2) sender(2) receiver(2) flag(1) route length(1) route index(1) route (variable)
WBACKSHORT	type(1) messageID(2) sender(2) receiver(2) route length(1) route index(1) route (variable)
WBRREQ	type(1) messageID(2) sender(2) receiver(2) TTL(1) points(1) route length(1) route (variable)
WBRREP	type(1) messageID(2) sender(2) receiver(2) route length(1) route index(1) route (variable)
WBAW	type(1) messageID(2) sender(2) receiver(2) flag(1) route length(1) route (variable)
SIEG	type(1) messageID(2) sender(2) receiver(2) route length(1) route (variable)

3.2 Reaktion

Nachrichtentyp	Wert	Reaktion nach Empfang
WB	0x34	<p>Ein WBACKSHORT senden falls Ziel erreicht:</p> <ul style="list-style-type: none"> • mit Wahrscheinlichkeit x den Ball fangen • mit Wahrscheinlichkeit y dem Ball ausweichen: ein WBAW(1) senden • mit Wahrscheinlichkeit $1-x-y$ ist man getroffen: ein WBACK senden <p>sonst:</p> <ul style="list-style-type: none"> • mit Wahrscheinlichkeit x den Ball abfangen • mit Wahrscheinlichkeit $1-x$: weiterleiten
WBACKSHORT	0x35	
WBACK	0x36	Ein WBACKSHORT senden Falls Ziel noch nicht erreicht: weiterleiten
WBRREQ	0x37	Falls es eine neue oder stark veränderte Route ist: sich selbst hinzufügen und weitersenden, sonst ignorieren
WBRREP	0x38	Falls Ziel noch nicht erreicht: weiterleiten
WBAW(1)	0x39	Sende ein WBAW(2) zurück
WBAW(2)		Sende dem ersten erhaltenen ein WBAW(3) zurück, der Rest wird ignoriert
WBAW(3)		Du bekommst einen Ball
SIEG	0x40	Durch Broadcast weiterleiten Das Spiel wird beendet

Kapitel 4

Schlussbemerkungen

4.1 Probleme

4.1.1 Wirelesskarten

Eines der grössten Probleme stellte die Hardware dar, genauer gesagt die im mobilen Rechner standardmässig fest eingebauten Wirelesskarten. Sie haben eine sehr gute Sende- und Empfangsleistung, die über sehr grosse Distanzen reicht und bei denen auch Wände oder Türen kein grosses Hindernis darstellen. Bei meinem Spiel ist es von Vorteil, wenn man nicht alle Mitspieler direkt hört und so ein vollständiger Graph entsteht. Diese Sendeleistung lässt sich nirgends verstellen. Zudem gibt es grosse Probleme, falls sich die Karten einmal nicht mehr hören, sich aber wieder bis an die Sendereichweite nähern. Sie senden zwar noch tadellos, können aber keine Pakete mehr empfangen. Erst wenn man sie ausschaltet und wieder neu einstellt, funktionieren sie wieder richtig. So benutze ich die alten externen Cisco Aironet 350 Karten, bei welchen man die Leistung so verringern kann, dass nur noch im gleichen Zimmer die Verbindung besteht. Auch sonst funktionieren sie fehlerfrei.

4.1.2 Timeout

Eine sehr wichtige Frage in verteilten Systemen stellt sich auch hier: Wie lange soll ein Timeout gewählt werden? Da bei der Routensuche im erfolglosen Fall nach einem Timeout jeweils die TTL erhöht wird, bis das ganze Netzwerk geflutet werden muss, kann es vorkommen, dass dieser Vorgang über mehrere Sekunden andauert. Ist damit das Timeout zu hoch? Auf der anderen Seite war es in praktischen Tests möglich, dass die genau gleichen Routen mehrmals beim Anfragestellenden ankamen. Dies kann jedoch nur geschehen, wenn der „Routerequest-Routereply“ länger als das Timeout unterwegs ist. Ist das Timeout zu kurz gewählt?

4.1.3 Ballverlust oder -flaute

Leider kann nicht garantiert werden, dass immer die gleiche Anzahl von Bällen im Spiel vorhanden ist. Durch unerkannte Paketverluste oder auch nur vermeindliche Verluste kann es trotz dreifacher Bestätigung vorkommen, dass ein Ball verloren geht oder ein neuer dazukommt. Es ist auch möglich, dass jemand aus dem Spiel aussteigt, wenn er gerade im Besitz von mehreren Bällen ist, welche dann verloren gehen. Aber wie soll man sicher sein, dass dieser Mitspieler nicht nur ganz kurz für alle unerreichbar ist und

dann wieder normal weiterspielt?

Das Spiel ist nur interessant, so lange man immer wieder einen Ball erhält. Um es etwas wahrscheinlicher zu machen, wird zu Beginn jedem Spieler zwei Bälle gegeben, nicht nur einer. Zudem bewirkt der Balltimer eine möglichst gleichmässige Ballverteilung.

4.1.4 Adhoc-Netzwerk

Grundsätzlich lässt sich sagen, dass in einem Adhoc-Netzwerk wirklich alle Pakete jederzeit verloren gehen können. Dies ist in der Praxis sogar ziemlich oft der Fall. Vor allem in den Grenzgebieten, wo man sich nur knapp oder immer wieder ein wenig hört, sind die Verbindungen sehr instabil. Da man immer wieder etwas vom Mitspieler empfängt, lässt man ihn noch aktiv. Die Treffer oder Routensuchen kommen nur sehr selten an, dass es fast besser wäre, der Mitspieler würde gestrichen. Auch scheint es, als ob zwei Teilnehmer nicht immer symmetrisch sind, d.h. man empfängt zwar Pakete, aber auf die eigenen reagiert dieser nicht mehr.

Die Dynamik wirkt sehr zähflüssig, was auf das zwar nötige, aber doch hohe Timeout und die Tatsache zurückzuführen ist, dass die Pakete sehr unterschiedliche Laufzeiten haben, sich sogar überholen können.

4.2 Erweiterungen

Je länger ich mich mit der Semesterarbeit beschäftige, desto interessanter wird sie. Verbesserungen, Erweiterungen oder Neuerungen, die das Spiel witziger oder unterhaltsamer machen, tauchen immer wieder auf. Leider geht jede Arbeit einmal zu Ende und ein Schlussstrich muss gezogen werden. Trotzdem, es würde noch einiges geben, das man zusätzlich implementieren könnte.

- Würde es interessanter werden, wenn man einen Graphen zur Verfügung hätte, der anzeigt, wie das aktuelle Netzwerk aussieht, wo sich die anderen Mitspieler befinden? Würde das die Mobilität fördern?
- Soll die Wahrscheinlichkeit steigen, einen Ball zu fangen oder einen fallengelassenen zu erobern, wenn man schon lange keinen mehr besessen hat?

4.3 Ziele erreicht?

Der Multihopmechanismus wird sehr häufig gebraucht. Es gibt keine zentrale Station und man kann jederzeit in ein Spiel eintreten oder es verlassen. Mit der Mobilität ist es ein bisschen schwieriger. Klar lernt man das Netzwerk am schnellsten kennen, wenn man sich bewegt. In grossen Netzen ist es mit der Zeit fast zwingend, seinen Standort zu wechseln, um noch genügend Punkte sammeln zu können. Die einzelnen Routen sind viel weniger lange gültig, so dass man sicher weniger getroffen wird.

Theoretisch ist es aber möglich, am Spiel teilzunehmen, ohne sich zu bewegen. Es können also auch Spieler mitmachen, die auf ein Ladegerät angewiesen sind.

4.4 Tests

In zwei Spielsessionen wurde die Tauglichkeit des Spieles überprüft. Sieben Spieler konnten sich bis zu einer Stunde dazu begeistern, mit einem mobilen Rechner durch die Gänge zu gehen und die Mitspieler abzuschliessen. Was bedeutet, dass das Spiel einen gewissen Reiz besitzen muss. Durch diese Tests kamen glücklicherweise keine sehr grossen und unerwartete Probleme zum Vorschein, doch gab es einige sehr gute Ideen, wie man das Spiel noch interessanter und unterhaltsamer gestalten könnte, was sich beim zweiten Test auch bestätigt hat. Eine konkrete Strategie in Bezug auf die Positionierung oder Mobilität konnte bis jetzt noch nicht gefunden werden, was vor allem auf die schlechte Kenntnis und lahme Aktualisierung des Netzwerkes zurückzuführen ist. Im Spiel selbst ergaben sich zwei verschiedene Taktiken:

- warten, bis man einige Bälle gesammelt hat und dann möglichst rasch über eine relativ grosse Distanz diese auf einen Mitspieler abfeuern, damit die Zwischenstationen nicht mehr alle Bälle abfangen können.
- bei einem erfolgreichen Ballgewinn sofort einen direkten Nachbar unter Beschuss nehmen.

Einig sind sich alle, dass man jeden Spieler im Durchschnitt gleich oft treffen soll, damit die Punkteverteilung optimal ist.