Star Realms > Forums > Strategy

Star Realms Simulator!!(↦ /thread/1462398/star-realms-simulator)

John @icesphere Oct 30, 2015

Post on how to create a bot: Create bot

John @icesphere Oct 31, 2015 (edited)

I found an error in how my simulator handles bases 😕

I've updated the results.

All matchups for the main bots had the the same winners with the exception of VelocityBot now beats HareBot. Probably the biggest difference was in the DefenseBot v AttackBot matchup - which has the same result, but is a much closer matchup now.

The color bot results also changed. Green bot now beats all the other color bots.

Interestingly the first player advantage is even bigger now.

John @icesphere Nov 5, 2015 (edited)

I created a bot for each two color combination. They can still buy cards in other colors, but are much more likely to buy cards in their colors.

Here are the results:

--BlueGreenBot--
BlueGreenBot v BlueYellowBot - 58.43% - 41.57% (Avg # turns: 31.25)
BlueGreenBot v GreenYellowBot - 61.25% - 38.75% (Avg # turns: 29.13)
BlueGreenBot v RedBlueBot - 46.24% - 53.76% (Avg # turns: 28.57)
BlueGreenBot v RedGreenBot - 49.93% - 50.07% (Avg # turns: 27.29)
BlueGreenBot v YellowRedBot - 49.55% - 50.45% (Avg # turns: 26.88)
BlueGreenBot v BlueGreenBot - 1st player wins: 55.48% (Avg # turns: 30.55)

--RedBlueBot--
RedBlueBot v BlueGreenBot - 53.04% - 46.96% (Avg # turns: 28.43)
RedBlueBot v BlueYellowBot - 63.12% - 36.88% (Avg # turns: 32.44)
RedBlueBot v GreenYellowBot - 57.57% - 42.43% (Avg # turns: 27.57)
RedBlueBot v RedGreenBot - 50.58% - 49.42% (Avg # turns: 28.80)
RedBlueBot v YellowRedBot - 50.79% - 49.21% (Avg # turns: 31.24)
RedBlueBot v RedBlueBot - 1st player wins: 58.47% (Avg # turns: 34.39)

--RedGreenBot--
RedGreenBot v BlueGreenBot - 50.54% - 49.46% (Avg # turns: 27.31)
RedGreenBot v BlueYellowBot - 58.07% - 41.93% (Avg # turns: 27.60)
RedGreenBot v GreenYellowBot - 56.41% - 43.59% (Avg # turns: 27.45)
RedGreenBot v RedBlueBot - 48.30% - 51.70% (Avg # turns: 29.08)
RedGreenBot v YellowRedBot - 50.73% - 49.27% (Avg # turns: 27.30)
RedGreenBot v RedGreenBot - 1st player wins: 56.60% (Avg # turns: 29.33)

--GreenYellowBot--
GreenYellowBot v BlueGreenBot - 38.05% - 61.95% (Avg # turns: 29.32)
GreenYellowBot v BlueYellowBot - 47.32% - 52.68% (Avg # turns: 30.54)
GreenYellowBot v RedBlueBot - 42.12% - 57.88% (Avg # turns: 27.73)
GreenYellowBot v RedGreenBot - 43.85% - 56.15% (Avg # turns: 27.58)
GreenYellowBot v YellowRedBot - 47.10% - 52.90% (Avg # turns: 27.33)
GreenYellowBot v GreenYellowBot - 1st player wins: 54.56% (Avg # turns: 30.41)

--BlueYellowBot--
BlueYellowBot v BlueGreenBot - 42.68% - 57.32% (Avg # turns: 31.26)
BlueYellowBot v GreenYellowBot - 52.71% - 47.29% (Avg # turns: 30.71)
BlueYellowBot v RedBlueBot - 36.44% - 63.56% (Avg # turns: 32.25)
BlueYellowBot v RedGreenBot - 42.33% - 57.67% (Avg # turns: 27.84)
BlueYellowBot v YellowRedBot - 42.33% - 57.67% (Avg # turns: 30.60)
BlueYellowBot v BlueYellowBot - 1st player wins: 55.65% (Avg # turns: 36.29)

--YellowRedBot--
YellowRedBot v BlueGreenBot - 49.34% - 50.66% (Avg # turns: 26.89)
YellowRedBot v BlueYellowBot - 57.78% - 42.22% (Avg # turns: 30.68)
YellowRedBot v GreenYellowBot - 52.45% - 47.55% (Avg # turns: 27.53)
YellowRedBot v RedBlueBot - 49.22% - 50.78% (Avg # turns: 30.96)
YellowRedBot v RedGreenBot - 49.39% - 50.61% (Avg # turns: 27.34)
YellowRedBot v YellowRedBot - 1st player wins: 57.31% (Avg # turns: 31.61)

John @icesphere Nov 5, 2015 (edited)

BlueGreenBot against the other bots:

BlueGreenBot v HareBot - 40.65% - 59.35% (Avg # turns: 28.20)
BlueGreenBot v AttackBot - 53.50% - 46.50% (Avg # turns: 28.80)
BlueGreenBot v DefenseBot - 53.56% - 46.44% (Avg # turns: 31.18)
BlueGreenBot v VelocityBot - 36.93% - 63.07% (Avg # turns: 27.31)
BlueGreenBot v TortoiseBot - 42.71% - 57.29% (Avg # turns: 28.19)
BlueGreenBot v ExpensiveBot - 64.41% - 35.59% (Avg # turns: 28.81)

Matt Schoonmaker-Ga... @railbaron Nov 6, 2015

Maybe you've already done this, but I was thinking about some bots to create to validate the simulator. Since it's a black box to us, I was just curious if these bots would have the expected results against your other bots.

DoNothingBot - Does not buy any cards. Plays with a 10 card deck the entire game.
OnlyBuyExplorerBot - Only buys explorers. Always scraps the explorers.

Here are some expected results when these bots play your other bots.

1) Both bots should lose to any other bot like 100% of the time, right? Maybe the OnlyBuyExplorer bot could eek out a win against some other bots in rare cases.

2) If you matched up a DoNothingBot against itself, we should be able to determine the expected win % between first and second player.

Determining expected win % between first and second player
Let's start with the 2nd player. So every time through the deck, 2nd player will deal 2 damage. After 24 times through the deck (48 individual turns, or overall turn 96), the 2nd player will have dealt 48 damage, guaranteed. So the 2nd player will deal lethal damage on turn 49 (overall 98) 22% of the time, and lethal damage on turn 50 (overall 100) 78% of the time.

So player 1 is a little trickier. After the second individual turn (overall turn 3), Player 1 could have dealt 0, 1, or 2 damage. From that point on, Player 1 will be guaranteed to deal 2 damage every 2 individual turns.

If dealt 0 damage first two individual turns, then it will take 48 additional individual turns to deal 48 damage, which is group turn 99. So Player 1 is guaranteed to lose, since Player 2 is guaranteed to deal lethal damage by turn 100.

If dealt 1 damage first two individual turns, then it will still take 48 additional turns to deal 48 damage, which is group turn 99. So Player 1 is guaranteed to lose, since Player 2 is guaranteed to deal lethal damage by turn 100.

If dealt 2 damage first two indvidual turns, then after 46 additional turns player 1 will deal 46 damage, for a total of 48 damage by group turn 95. So on group turn 97, player 1 will deal lethal damage 22% of the time, or else will deal lethal damage on turn 99 78% of the time.

Now let's add it all up.

Chance dealing 0 damage first two turns = 2.2%
Chance dealing 1 damage first two turns = 35.6%
Chance dealing 2 damage first two turns = 62.2%

Chance player 2 wins =
Player 1 deals 0 damage or
Player 1 deals 1 damage or
Player 1 deals 2 damage and Player 1 doesn't deal lethal damage turn 97 and player 2 does deal lethal damage on turn 98

Chance player 2 wins = 2.2% + 35.6% + (62.2% * 78% * 22%)
Chance player 2 wins = 48.47%
Chance player 1 wins = 100% - chance player 2 wins = **51.53%**

Hopefully someone is still reading :-). It would be helpful I think to know how long each game lasts, and then take an average of the overall turn number. I think it's interesting to analyze the length of the games, and I would hope it would be an easy thing to track in the simulator. If you added this in, you could also just verify that when a DoNothingBot plays a DoNothingBot, the games should 100% of the time last between 97 and 100 turns.

@greylag Nov 6, 2015 (edited)

Petition to rename OnlyBuyExplorerBot to ExploderBot plz

John @icesphere Nov 6, 2015 (edited)

As requested, here are the results from the DoNothingBot and the ExplorerBot:

--DoNothingBot--
DoNothingBot v HareBot - 0% - 100.00% (Avg # turns: 22.01)
DoNothingBot v AttackBot - .07% - 99.93% (Avg # turns: 22.94)
DoNothingBot v DefenseBot - .01% - 99.99% (Avg # turns: 25.08)
DoNothingBot v VelocityBot - 0% - 100.00% (Avg # turns: 21.83)
DoNothingBot v TortoiseBot - 0% - 100.00% (Avg # turns: 22.78)
DoNothingBot v ExpensiveBot - 0% - 100.00% (Avg # turns: 22.73)
DoNothingBot v ExplorerBot - 0% - 100.00% (Avg # turns: 25.19)
DoNothingBot v BlueGreenBot - .01% - 99.99% (Avg # turns: 22.68)
DoNothingBot v DoNothingBot - 1st player wins: 51.31% (Avg # turns: 98.83)

--ExplorerBot--
ExplorerBot v HareBot - .14% - 99.86% (Avg # turns: 22.16)
ExplorerBot v AttackBot - .94% - 99.06% (Avg # turns: 22.61)
ExplorerBot v DefenseBot - .52% - 99.48% (Avg # turns: 25.57)
ExplorerBot v VelocityBot - .09% - 99.91% (Avg # turns: 22.33)
ExplorerBot v TortoiseBot - .12% - 99.88% (Avg # turns: 23.32)
ExplorerBot v ExpensiveBot - .02% - 99.98% (Avg # turns: 23.28)
ExplorerBot v BlueGreenBot - .05% - 99.95% (Avg # turns: 23.04)
ExplorerBot v ExplorerBot - 1st player wins: 54.13% (Avg # turns: 24.52)

Matt Schoonmaker-Ga... @railbaron Nov 6, 2015

Woah, thanks for the quick turnaround!

So all the results look spot on, except for the very last line :-).

Why is the avg # of turns higher when it is ExplorerBot v ExplorerBot than it is for ExploreBot v DoNothingBot? And it is higher by a lot, like 17 turns! Maybe these two got switched? It should definitely be a faster game, since either ExplorerBot could win.

So I calculated the 1st player should win 51.53% of the time, and the simulation calculated 50.76%. I just remembered that I rounded some of my calculations, so when I don't round any of my calculations it is 51.4678%. That's still a difference of .7%, which isn't big but it's not great either. There's a chance my calculations aren't correct. Or maybe the random number generator your program uses isn't completely random, which could cause slight differences like this? Any other ideas to explain the difference? If you run enough simulations, it should converge to the mathematically calculated probability, right?

@greylag Nov 6, 2015

> **Matt Schoonmaker-Ga...** @railbaron wrote:

> Woah, thanks for the quick turnaround!
>
> So all the results look spot on, except for the very last line :-).
>
> Why is the avg # of turns higher when it is ExplorerBot v ExplorerBot than it is for ExploreBot v DoNothingBot? And it is higher by a lot, like 17 turns! Maybe these two got switched? It should definitely be a faster game, since either ExplorerBot could win.
>
> So I calculated the 1st player should win 51.53% of the time, and the simulation calculated 50.76%. I just remembered that I rounded some of my calculations, so when I don't round any of my calculations it is 51.4678%. That's still a difference of .7%, which isn't big but it's not great either. There's a chance my calculations aren't correct. Or maybe the random number generator your program uses isn't completely random, which could cause slight differences like this? Any other ideas to explain the difference? If you run enough simulations, it should converge to the mathematically calculated probability, right?

.7% is nothing in most situations. Whether or not it's within a few standard deviations depends on the sample size. But in general if two results are within .7% (in a population where results can freely vary between 0 and 100%), statistically they're pretty much identical.

Matt Schoonmaker-Ga... @railbaron Nov 6, 2015

> **@greylag** wrote:
>
> > **Matt Schoonmaker-Ga...** @railbaron wrote:
> > Woah, thanks for the quick turnaround!
> >
> > So all the results look spot on, except for the very last line :-).
> >
> > Why is the avg # of turns higher when it is ExplorerBot v ExplorerBot than it is for ExploreBot v DoNothingBot? And it is higher by a lot, like 17 turns! Maybe these two got switched? It should definitely be a faster game, since either ExplorerBot could win.
> >
> > So I calculated the 1st player should win 51.53% of the time, and the simulation calculated 50.76%. I just remembered that I rounded some of my calculations, so when I don't round any of my calculations it is 51.4678%. That's still a difference of .7%, which isn't big but it's not great either. There's a chance my calculations aren't correct. Or maybe the random number generator your program uses isn't completely random, which could cause slight differences like this? Any other ideas to explain the difference? If you run enough simulations, it should converge to the mathematically calculated probability, right?
>
> .7% is nothing in most situations. Whether or not it's within a few standard deviations depends on the sample size. But in general if two results are within .7% (in a population where results can freely vary between 0 and 100%), statistically they're pretty much identical.

But like you said, it depends on the number of simulations run (sample size). The more simulations you run, the closer it should get. It's not like these are two estimates: one of the numbers is the actual answer (hopefully). So if you run more simulations and it doesn't converge to the actual answer, then something is wrong, albeit a small something, but still something.

Ok, so I think to calculate the standard deviation, you multiply the observed proportion by 1 - observed proportion, divided by sample size, and take the square root.

[(.5076)*(.4924)/sample size]^.5

http://sites.stat.psu.edu/~dhunter/100/spring2005/lecture/le...

So I guess we'd need to know what the sample size was. I think it it's less than 20,000, then the observed amount is within the 95% confidence interval. If it's more than 20,000, then it is outside the 95% confidence interval. Hopefully I'm remembering this stuff correctly :-). I do like statistics, but haven't done this stuff anytime recently.

John @icesphere Nov 6, 2015

> **Matt Schoonmaker-Ga...** @railbaron wrote:
> Woah, thanks for the quick turnaround!
>
> So all the results look spot on, except for the very last line :-).
>
> Why is the avg # of turns higher when it is ExplorerBot v ExplorerBot than it is for ExploreBot v DoNothingBot? And it is higher by a lot, like 17 turns! Maybe these two got switched? It should definitely be a faster game, since either ExplorerBot could win.
>
> So I calculated the 1st player should win 51.53% of the time, and the simulation calculated 50.76%. I just remembered that I rounded some of my calculations, so when I don't round any of my calculations it is 51.4678%. That's still a difference of .7%, which isn't big but it's not great either. There's a chance my calculations aren't correct. Or maybe the random number generator your program uses isn't completely random, which

> There's a chance my calculations aren't correct. Or maybe the random number generator your program uses isn't completely random, which could cause slight differences like this? Any other ideas to explain the difference? If you run enough simulations, it should converge to the mathematically calculated probability, right?

Yeah, I was also confused by that result. I thought about it and figured it must be because both bots are fighting over the 10 explorers, so they don't get as many explorers per bot which makes it take longer. (I actually discovered there were 11 explorers available - after fixing that it increased the average turns to 79.88, and changed 1st player advantage to 52.23%).

I've been running each matchup with 10,000 games.

I re-ran the DoNothingBot v DoNothingBot two more times and it came out with 1st player wins 51.5% and 51.49%. Not sure why it was so different the first time.

Matt Schoonmaker-Ga... @railbaron Nov 6, 2015

> **John** @icesphere wrote:
>
> > **Matt Schoonmaker-Ga...** @railbaron wrote:
> >
> > Woah, thanks for the quick turnaround!
> >
> > So all the results look spot on, except for the very last line :-).
> >
> > Why is the avg # of turns higher when it is ExplorerBot v ExplorerBot than it is for ExploreBot v DoNothingBot? And it is higher by a lot, like 17 turns! Maybe these two got switched? It should definitely be a faster game, since either ExplorerBot could win.
> >
> > So I calculated the 1st player should win 51.53% of the time, and the simulation calculated 50.76%. I just remembered that I rounded some of my calculations, so when I don't round any of my calculations it is 51.4678%. That's still a difference of .7%, which isn't big but it's not great either. There's a chance my calculations aren't correct. Or maybe the random number generator your program uses isn't completely random, which could cause slight differences like this? Any other ideas to explain the difference? If you run enough simulations, it should converge to the mathematically calculated probability, right?
>
> Yeah, I was also confused by that result. I thought about it and figured it must be because both bots are fighting over the 10 explorers, so they don't get as many explorers per bot which makes it take longer. (I actually discovered there were 11 explorers available - after fixing that it increased the average turns to 79.88, and changed 1st player advantage to 52.23%).
>
> I've been running each matchup with 10,000 games.
>
> I re-ran the DoNothingBot v DoNothingBot two more times and it came out with 1st player wins 51.5% and 51.49%. Not sure why it was so different the first time.

Ok awesome. You answered all my questions, and everything looks good! Based on my math, a 10,000 sample set could be off by .7% and it wouldn't be a big deal. It's nice to know that after rerunning it twice, the results were closer to 51.5%. Also your explanation about fighting over the explorers makes a lot of sense, so I agree that would increase the game length.

So now I'll have to start thinking about how to make a competitive bot :-).

John @icesphere Nov 6, 2015

> **Matt Schoonmaker-Ga...** @railbaron wrote:
>
> > **John** @icesphere wrote:
> >
> > > **Matt Schoonmaker-Ga...** @railbaron wrote:
> > >
> > > Woah, thanks for the quick turnaround!
> > >
> > > So all the results look spot on, except for the very last line :-).
> > >
> > > Why is the avg # of turns higher when it is ExplorerBot v ExplorerBot than it is for ExploreBot v DoNothingBot? And it is higher by a lot, like 17 turns! Maybe these two got switched? It should definitely be a faster game, since either ExplorerBot could win.
> > >
> > > So I calculated the 1st player should win 51.53% of the time, and the simulation calculated 50.76%. I just remembered that I rounded some of my calculations, so when I don't round any of my calculations it is 51.4678%. That's still a difference of .7%, which isn't big but it's not great either. There's a chance my calculations aren't correct. Or maybe the random number generator your program uses isn't completely random, which could cause slight differences like this? Any other ideas to explain the difference? If you run enough simulations, it should converge to the mathematically calculated probability, right?

> > Yeah, I was also confused by that result. I thought about it and figured it must be because both bots are fighting over the 10 explorers, so they don't get as many explorers per bot which makes it take longer. (I actually discovered there were 11 explorers available - after fixing that it increased the average turns to 79.88, and changed 1st player advantage to 52.23%).
> >
> > I've been running each matchup with 10,000 games.
> >
> > I re-ran the DoNothingBot v DoNothingBot two more times and it came out with 1st player wins 51.5% and 51.49%. Not sure why it was so different the first time.
>
> Ok awesome. You answered all my questions, and everything looks good! Based on my math, a 10,000 sample set could be off by .7% and it wouldn't be a big deal. It's nice to know that after rerunning it twice, the results were closer to 51.5%. Also your explanation about fighting over the explorers makes a lot of sense, so I agree that would increase the game length.
>
> So now I'll have to start thinking about how to make a competitive bot :-).

So I just tried buying lots of Explorers on the digital game and it looks like they are unlimited. I changed my simulator to have unlimited Explorers and now the ExplorerBot v ExplorerBot match averaged just 24.53 turns.

I would definitely like to see the simulator with more competitive bots so if you can make one that would be awesome! I was planning on working on a more competitive bot soon, but the more the better!

Matt Schoonmaker-Ga... @railbaron Nov 6, 2015

Woah, that's really interesting that just buying explorers and the game only takes 24.5 turns. Of course that's with neither player having any healing or base protection.

Scott Heise Designer @HomerJr Nov 6, 2015

> **Matt Schoonmaker-Ga...** @railbaron wrote:
>
> Woah, that's really interesting that just buying explorers and the game only takes 24.5 turns. Of course that's with neither player having any healing or base protection.

And good deck velocity, with all that self-scrapping (a very underrated ability, IMO). 🙂

John @icesphere Nov 11, 2015

I've made some improvements and fixes to the simulator. I've updated the results in the initial post. I also removed the single color bot match-ups since they take a lot longer to simulate, and they are not as interesting since they only buy one color.

The main bot match-ups still all have the same winners. DefenseBot seems to have benefited the most from the changes. Also the first player advantage is not as big anymore. First player only wins about 57% of the time now.

John @icesphere Nov 11, 2015

I've also updated the two color bots. These results had significant changes. RedBlueBot now does the best - which makes more sense to me. I was surprised when BlueGreenBot did so well in the previous results.

Chris @LetsGetTrivial Nov 11, 2015

> **John** @icesphere wrote:
>
> I've made some improvements and fixes to the simulator. I've updated the results in the initial post.

Can you let us know what was "fixed" to have some context for the changes in results?

Flying Arrow @FlyingArrow Nov 11, 2015

Any chance of sharing the source code?

John @icesphere Nov 11, 2015

> **Chris** @LetsGetTrivial wrote:
>
> > **John** @icesphere wrote:

> **John** @icesphere wrote:
>
> I've made some improvements and fixes to the simulator. I've updated the results in the initial post.

Can you let us know what was "fixed" to have some context for the changes in results?

One big improvement was to re-evaluate card play order after each card played. This is important especially when a card draws another card.

Things that were fixed included ally abilities being triggered when they shouldn't have been, and base abilities not always being used.

John @icesphere Nov 11, 2015

> **Flying Arrow** @FlyingArrow wrote:
>
> Any chance of sharing the source code?

There are still some things I want to cleanup and do first, but I plan on eventually putting it up on GitHub.

John @icesphere Nov 12, 2015

Source code is now available!!

https://github.com/icesphere/star-realms-simulator

Feel free to download and run it yourself. If you do, please post any interesting results you find.

I'm also hoping that some of you will be interested in making the simulator better, or adding new features. If so please submit a pull request.

John @icesphere Nov 17, 2015

I've created a Trello board to track updates to the simulator.

https://trello.com/b/OYgPboK0/star-realms-simulator

If you are interested in contributing to the simulator I can add you as a member to the Trello board.

Matt Schoonmaker-Ga... @railbaron Nov 30, 2015

Hey, I admit I haven't tried downloading the simulator myself yet (I don't know much coding or java). And maybe I've already mentioned this, but... it would be awesome to be able to create a real mid-game situation, and run simulations to determine the probability of each deck winning. We would probably just use simple inputs in this situation, like

1) Don't buy any cards
2) Scrap all cards for draws or damage

So this wouldn't be used to determine the best bot or play, but would give you an idea of your chances of winning if players just played their cards and dealt damage.

I think having this information could be a stepping stone to help create better bots though. This basically would tell you 'In the current situation, the game is likely to last X many turns, and I will win Y% of the time'. You may want to switch your strategy based on this result.

Anyway, just thought I'd throw this out there :-). I'll let you know if I get around to trying to figure it out myself.

John @icesphere Nov 30, 2015

> **Matt Schoonmaker-Ga...** @railbaron wrote:
>
> Hey, I admit I haven't tried downloading the simulator myself yet (I don't know much coding or java). And maybe I've already mentioned this, but... it would be awesome to be able to create a real mid-game situation, and run simulations to determine the probability of each deck winning. We would probably just use simple inputs in this situation, like
>
> 1) Don't buy any cards
> 2) Scrap all cards for draws or damage
>
> So this wouldn't be used to determine the best bot or play, but would give you an idea of your chances of winning if players just played their cards and dealt damage.
>
> I think having this information could be a stepping stone to help create better bots though. This basically would tell you 'In the current situation,

> the game is likely to last X many turns, and I will win Y% of the time'. You may want to switch your strategy based on this result.
>
> Anyway, just thought I'd throw this out there :-). I'll let you know if I get around to trying to figure it out myself.

Good idea!

I'll add it to my list of things to work on. Right now I'm working on adding in all the cards available in the digital version.