

[Skip to main content](#)

Stack Overflow

1. [About](#)
2. Products
3. [For Teams](#)
1. [Stack Internal Implement a knowledge platform layer to power your enterprise and AI tools.](#)
2. [Stack Data Licensing Get access to top-class technical expertise with trusted & attributed content.](#)
3. [Stack Ads Connect your brand to the world's most trusted technologist communities.](#)
4. [Releases Keep up-to-date on features we add to Stack Overflow and Stack Internal.](#)
5. [About the company Visit the blog](#)



Loading...

[current community](#)

- [Stack Overflow](#)
- [help chat](#)
- [Meta Stack Overflow](#)

your communities

[Sign up](#) or [log in](#) to customize your list.

[more stack exchange communities](#)

- [company blog](#)
 - 3. [Log in](#)
 - 4. [Sign up](#)
- [AI Assist is now on Stack Overflow.](#) Start a chat to get instant answers from across the network. Sign up to save and share your chats.
- [Home](#)
 - [Questions](#)
 - [AI Assist](#)
 - [Tags](#)
 - [Challenges](#)
 - 7. [Chat](#)
 - [Articles](#)
 - [Users](#)
 - [Companies](#)
 - [Collectives](#)

Communities for your favorite technologies. [Explore all Collectives](#)

Stack Internal

Stack Overflow for Teams is now called **Stack Internal**. Bring the best of human thought and AI automation together at your work.

[Try for free](#) [Learn more](#)

[Stack Internal](#)

Bring the best of human thought and AI automation together at your work. [Learn more](#)

Collectives™ on Stack Overflow

Find centralized, trusted content and collaborate around the technologies you use most.

[Learn more about Collectives](#)

[Stack Internal](#)

Knowledge at work

Bring the best of human thought and AI automation together at your work.

[Explore Stack Internal](#)

[How do I iterate over the coordinates of a line between two coordinates? \[closed\]](#)

[Ask Question](#)

Asked 1 year, 10 months ago

Modified [1 year, 10 months ago](#)

Viewed 211 times

-3

Closed. This question needs [details or clarity](#). It is not currently accepting answers.

Want to improve this question? As written, this question is lacking some of the information it needs to be answered. If the author adds details in comments, consider [editing them into the question](#). Once there's sufficient detail to answer, vote to reopen the question.

Closed last year.

[Improve this question](#)

I have a struct called `Point` with `x` and `y` coordinates and a tuple struct called `Line` which has two `Points`. I want to have a method on the `Point` struct which returns an iterator of all the points between those two points in that struct, including the points themselves. I already have some checks for if it isn't horizontal or vertical, so you can just panic there. What's the idiomatic way to do this? Is there a trait that I should implement for the `iter` method? I just can't find out how to do this with a range nor by creating my own iterator struct.

I had a load of other, more specific problems, but I don't want to mention them because I think they probably aren't even on the correct track and if I solved it the idiomatic way, I probably won't need it anyway and it would be a waste of time. This is intentionally quite a broad question.

Edit 2: It's for Battleships. I have a HashSet of all hit targets, I need to check if a ship is sunk, among other things.

Edit 1: And just my code which doesn't work just in case:

```
struct Point {
    x: u8,
    y: u8,
}

impl Point {
    fn new(x: u8, y: u8) -> Self {
        Self { x, y }
    }
}

struct Line(Point, Point);

impl Line {
    fn iter(&self) -> impl Iterator<Item = u8> {
        if self.0.x == self.1.x {
            to_iter_u8(self.0.y, self.1.y).map(|y| Point::new(self.0.x, y))
        } else if self.0.y == self.1.y {
            to_iter_u8(self.0.x, self.1.x).map(|x| Point::new(x, self.0.y))
        } else {
            panic!("Points must be horizontal or vertical to each other");
        }
    }
}

fn to_iter_u8(a: u8, b: u8) -> impl Iterator<Item = u8> {
    if a > b {
        b..=a
    } else {
        a..=b
    }
}
```

```

- [rust](#)

[Share](#)

[Improve this question](#)

Follow

[edited Feb 9, 2024 at 7:08](#)

asked Feb 7, 2024 at 15:46

[ihatenullptrs](#)

4566 bronze badges

4

"All points" meaning, what? Integer pixel values? Increments of some length? Mathematically speaking there is an infinite number of points on a line of any length.

tadman – [tadman](#)

2024-02-07 16:07:56 +00:00

Commented Feb 7, 2024 at 16:07

1

This is how an iterator is created: [doc.rust-lang.org/rust-by-example/trait/iter.html](https://doc.rust-lang.org/rust-by-example/trait/iter.html) maybe you show what you already have, so we can help more on the problem

trust\_nickol – [trust\\_nickol](#)

2024-02-07 18:14:31 +00:00

Commented Feb 7, 2024 at 18:14

3

@ihatenullptrs They asked because it's an important and valid question. For example, if you are using floats then would you really want an iterator of every single possible set of floating point coordinates that may differ by only the epsilon? Your question is underspecified in this regard, and SO is not fond of making guesses regarding information that hasn't been provided to us.

cdhowie – [cdhowie](#)

2024-02-07 18:50:29 +00:00

Commented Feb 7, 2024 at 18:50

1

Please also note that your replies to comments have a hostile tone, which tends to frustrate people and make them less likely to help you. For a new user asking a (poor) first question, you may want to be especially mindful of your tone when replying to people who are only trying to help (instead, you appear to be vaguely insinuating that they are dumb, with statements like "it's quite obvious")

Codebling – [Codebling](#)

2024-02-08 21:48:16 +00:00

Commented Feb 8, 2024 at 21:48

1

@Codebling No, it's only horizontal or vertical lines, because ships in battleships can only be placed like that. It's confusing because the panic message was wrong and said horizontal and diagonal instead of horizontal and vertical. By diagonal I mean anything that is not horizontal or vertical.

ihatenullptrs – [ihatenullptrs](#)

2024-02-09 07:12:58 +00:00

Commented Feb 9, 2024 at 7:12

| [Show 11 more comments](#)

## 1 Answer

Sorted by: [Reset to default](#)

Highest score (default) Trending (recent votes count more) Date modified (newest first) Date created (oldest first)

1

The `iter` function fails to compile for a few reasons.

First, you've declared the type of `Iterator::Iter` to be `u8`, but the function is actually trying to return an iterator of `Point` values. This is easily solved by changing the declared return type to match.

Second, you accept `self` by reference but you don't indicate that the returned `Iterator` captures the lifetime of the reference. There's two ways to fix this:

1. Indicate the captured lifetime (`impl Iterator<Item = Point> + '_`).
2. Make `Point` implement `Copy` and take `self` by value.
3. Extract the `x` or `y` value in `iter` before returning the iterator.

The second option is probably better since the type is very small, and it should probably implement `Copy` anyway. You'll have to make your closures move in order to copy `self` into them.

Up to this point, the fixed code looks like this (code not changed has been omitted):

```
#[derive(Clone, Copy)]
struct Point {
 x: u8,
 y: u8,
}

impl Line {
 fn iter(self) -> impl Iterator<Item = Point> {
```

```

 if self.0.x == self.1.x {
 to_iter_u8(self.0.y, self.1.y).map(move |y| Point::new(self.0.x, y))
 } else if self.0.y == self.1.y {
 to_iter_u8(self.0.x, self.1.x).map(move |x| Point::new(x, self.0.y))
 } else {
 panic!("Points must be horizontal or diagonal");
 }
 }
}

```

The last problem is that `impl Iterator` doesn't mean "any type implementing `Iterator`," it means "one type implementing `Iterator` inferred from the function body." The two returning conditionals both return a value implementing `Iterator`, but the closure passed to `map` in both cases is of a different type (they have to be different types to behave differently), and the closure's type is part of the iterator's type. Therefore, the two conditional blocks do not return the same type, and compilation fails.

There's a few ways this problem can be solved.

The easiest (but least efficient) option would be to return a `Box<dyn Iterator>`. This requires a heap allocation to hold the iterator and dynamic dispatch to iterate over the items, but allows you to return iterators of different types.

```

impl Line {
 fn iter(self) -> Box

```

Another option would be to use the [either crate](#), which provides an enum type `Either` with two variants (Left and Right), each of which can hold a different type. This type implements many traits (including `Iterator`) as long as both the left and right types do.

```

impl Line {
 fn iter(self) -> impl Iterator<Item = Point> {
 if self.0.x == self.1.x {
 Either::Left(to_iter_u8(self.0.y, self.1.y).map(move |y| Point::new(self.0.x, y)))
 } else if self.0.y == self.1.y {
 Either::Right(to_iter_u8(self.0.x, self.1.x).map(move |x| Point::new(x, self.0.y)))
 } else {
 panic!("Points must be horizontal or diagonal");
 }
 }
}

```

The last option would be to roll your own iterator type. In many cases this can be more efficient than using `Either`, but in this particular case, the generated code will be very similar and have about the same performance characteristics: a custom iterator is going to require a conditional to handle vertical vs. horizontal iteration, just as `Either` requires a conditional to determine if the wrapped iterator is the left or right type.

[Share](#)

[Improve this answer](#)

Follow

answered Feb 8, 2024 at 16:36

[cdhowie](#)

172k2525 gold badges303303 silver badges324324 bronze badges

Sign up to request clarification or add additional context in comments.

## 2 Comments

Add a comment

[ihatenullptrs](#)

[ihatenullptrs](#) Over a year ago

I managed to do the `Box` pointer, `dyn` and lifetime with some help for the compiler on my own, (which I should have said but I didn't because it didn't feel like a real answer), but I wouldn't have thought of cloning it.

2024-02-09T07:19:17.33Z+00:00

0

Reply

- Copy link

ihatenumpptrs

[ihatenumpptrs Over a year ago](#)

How can I ask better questions next time?

2024-02-09T07:20:10.237Z+00:00

0

Reply

- Copy link

Start asking to get answers

Find the answer to your question by asking.

[Ask question](#)

Explore related questions

- [rust](#)

See similar questions with these tags.

- The Overflow Blog

[Introducing Stack Overflow AI Assist—a tool for the modern developer](#)

[Treating your agents like microservices](#)

- Featured on Meta

[Chat room owners can now establish room guidelines](#)

[AI Assist is now available on Stack Overflow](#)

[Policy: Generative AI \(e.g., ChatGPT\) is banned](#)

## Related

[1087](#)

[What are the differences between Rust's `String` and `str`?](#)

530

[What is the difference between iter and into\\_iter?](#)

241

[How do I iterate over a range with a custom step?](#)

509

[How do I print the type of a variable in Rust?](#)

560

[How do you disable dead code warnings at the crate level in Rust?](#)

298

[What is the difference between Copy and Clone?](#)

306

[What is the equivalent of the join operator over a vector of Strings?](#)

166

[How do you iterate over a string by character](#)

422

[Why does the Rust compiler not optimize code assuming that two mutable references cannot alias?](#)

345

[How can I include a module from another file from the same project?](#)

## Hot Network Questions

- [Best Welding Method to Fix a Damaged Steel Surly Frame](#)
- [Why Select Within Distance set to 1m selects every waterbody?](#)
- [Why is "cmd | xargs" to join lines of cmd with spaces wrong](#)
- [Storm Sphere and Obstruction](#)
- [On Day 14 of HORSES, what do I do after watching the VHS Tape?](#)
- [What's the point of swapping polarity between mod slots?](#)
- [How do you fight a Guardian Scout?](#)
- [Is "We will review application starting from..." a deadline?](#)
- [Is my homebrew Species "Attuned" balanced for D&D 2024?](#)

- [Swords and sandals movie where the ancient Greek hero fights in space](#)
- [Schottky OR-ing diodes passing reverse voltage; is it a faulty diode?](#)
- [Is there any way to restore hearts without consuming anything, without hot springs, and without shrines?](#)
- [Can I Automatically List Supported Kernel Parameters in GRUB?](#)
- [What international legal mechanisms are in place to stop a country from outright expropriating foreign assets?](#)
- [Is it possible to use LaTeX to create RPG-related PDFs, such as character sheets or adventure modules?](#)
- [Preservation of universal sentences under substructures in higher-order logic](#)
- [The fourth moment of the Riemann zeta function without absolute values](#)
- [When should I use tied notes vs. the whole note duration?](#)
- [PSE Advent Calendar 2025 \(Day 6\): Reindeer with presents](#)
- [King defense setups](#)
- [A hilarious parody of Western movies where an entire town collapses into tunnels](#)
- [Curve has weird bumps when triangulated](#)
- [Is there a term for forming emotional or relational bonds with AI chatbots? If not, is "cyberpomorphic" a valid neologism?](#)
- [Zero Inflated Beta Regression \(or not\)?](#)

[more hot questions](#)

## [Stack Overflow](#)

- [Questions](#)
- [Help](#)
- [Chat](#)

## [Business](#)

- [Stack Internal](#)
- [Stack Data Licensing](#)
- [Stack Ads](#)

## [Company](#)

- [About](#)
- [Press](#)
- [Work Here](#)
- [Legal](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [Contact Us](#)
- [Cookie Settings](#)
- [Cookie Policy](#)

## [Stack Exchange Network](#)

- [Technology](#)
- [Culture & recreation](#)
- [Life & arts](#)
- [Science](#)
- [Professional](#)
- [Business](#)
- [API](#)
- [Data](#)
- [Blog](#)
- [Facebook](#)
- [Twitter](#)
- [LinkedIn](#)
- [Instagram](#)