Times will wait till the recording kicks in.
I don't know if it's instance there
ago started recording like we got now.
So an aspect of our assignment that we have
to do is interviewing AIT Professional.
You're trying your professional
loose terms at me.
To try and get a better understanding
of like the in's and outs of working
within the industry and see what the
day-to-day operations within data like.
So starting off what aspects of IT do
you work it? I'm a software developer.
It's as well we go sorry.
Yeah, but yeah,
software developer mostly working like C++,
little bit of HTML, JavaScript,
that kind of thing but mostly
back end stuff with C sharp.
And do you have any previous
projects you've worked on that you
can mention or hold us with us?
I guess yeah, the company I work
for does a lot of stuff we do.
We do the NAPLAN that online NAPLAN and
we do the drivers license in Australia.
Across Crystal version we do stuff
like the Singapore TAFE Institute.
We do their stuff.

We do the PBTS assignments which is
like a worldwide high school level exam.
Would you like as as well?
So do a lot of. Assessment based stuff.
Basically yeah cool.
So like could you breakdown?
I guess what?
Like your role has been within
these projects.
Sure, so predominantly I work on a
team of like 5 or 6 people and we
generally get work assigned to us from
the larger like planning committee.
So there will be stuff like we need
you to implement away for student
result data to get exported and you
know will will as a team work out.
The best way to do that.
And I'll generally be responsible for
coding all the back end part of it.
So that they could keep example
that would be.
Yeah, when we were exporting student data,
I wrote a background process
that gathered the data and then
exported it to a third party system.
OK cool so within.
So your team I guess.

Could you tell us a little bit more
about like that that team composition?
So you said you're the
software developer within that.
So what are some of the other
roles that you directly work with,
right?
So I actually work on a larger soft
like a software development team,
so on the team with me there are
think five other developers so that
split between front end and back end.
There's a tech lead in a scrum master.
He kind of takes care of
the day-to-day admin stuff.
And then there's someone from
the product team on there.
Take care of things like making sure
we're designing things that look correct,
making sure we will have
our requirements correct.
Make sure they're creating the right stuff,
and then we have testers which make
sure that it's all working properly.
What was the scrum Master scrum master?
So we work in a environment called Agile,
and what that means is we generally
do things in like short two week
bursts and the Scrum master is kind

of the master of that process,
making sure where it's you know,
planning things correctly,
taking in the right amount of work,
that kind of stuff.
So that agile environment is that,
I guess a practice or just is
that like a fancy term for a way
of distributing work?
Yeah,
it's actually quite a common one is
probably the most common across the
software development is industry,
where basically you you set aside
amount of time at my work said
website 2 weeks and we say our goal
at the end of the two weeks is to
have a certain shippable product.
So loud example of that might
be we want this screen to work.
So across the two week
period will create the.
The screen will create a user interface.
That screen will create all the
back end stuff for that screen.
Basically it's a way to work,
you know small period to
create shippable product.

Great small period.
How do you?
Mentioned working a developing the
NAPLAN like online computer system,
would you be able to kind of breakdown
how that goes from concept to right?
You're not trying to, you know,
figure out how to stop students from
messing around with the computer.
Would you be able to want to?
Yeah, sure.
Not blend is actually quite an old project.
It's one that only worked on the tail end of,
but I know a lot about it since
it's quite a big project for us.
So for that one particularly interesting
one cause prior to doing the NAPLAN
would primarily done like learning things
like courses people could do rather
than specific assessments so a big
part of that for us was reworking our
entire system to focus more on the like,
question and answer based model.
Like like I'm sure you both
probably done the NAPLAN online.
Yeah, you know, for years?
Yeah thanks yeah.
So yeah, big part of that was
focusing on reworking that.

So every part of that was making
the actual user experience,
and then we had to rework a
bunch of side systems.
Things like importing,
you know the entire Australian student
database into a program and then things
like exporting the results out of that.
So something like that is that.
I developed like similar to like a micro
service model where everything segmented.
Theoretically, that's how it
should have been done, yes.
Unfortunately, at the time
we were very much trying to
get it like across the line.
Our current development model
is very microservice based,
so we try to create everything as
very separate parts of our product.
I say it's cool, OK so.
Like the process with that.
So let's say, yeah, as you said.
So if you are working within
your team five or six,
you work in agile environment.
So you're basically aiming to
get components done within.

You know two week blocks water.
Have you been?
Is there sort of in some sort of?
Uh, my guess challenge or challenges that
that you faced even technical or I guess
both technical and whilst working within.
You know a team war on
some of these projects.
Sure, yeah.
Speak to one of the more recent product
problems we have is we had a lot of.
Separate code branches,
so essentially the same kind of product,
but split across a couple different
versions and what we've been trying
to focus on quite recently is trying
to get back to one singular version,
and it's in cases like this where the
Sprint and agile method kind of fall
apart a little bit because you know,
there's not really a discrete piece
of work that needs to be done
in this case of like you know,
keep putting stuff back into the one branch,
keep fixing the bugs that come out
of putting that into one branch,
so I guess a technical challenge
for us recently has been trying
to consolidate all this code.

All these different branches back
into one singular code branch and
quick like a lot of technical
challenges to come up with that,
as well as a lot of process
challenges have come up.
OK, So what about team based
challenges I guess yeah.
Again, like a lot of those process
challenges are being team based, so you know.
Yeah, but like I mentioned earlier,
we've got like a product person and they
kind of don't really know what to do.
They just like.
Keep working at what you're doing and
you know it's really hard to plan what
we're doing for two weeks when a lot
of that planning is just fix a bug and
fixing the bug could take two weeks.
Could take 2 minutes,
so it's very hard to focus on a lot
of those planning based aspects.
Yeah, OK cool so so like this is
getting back a bit more technical
so you mentioned that you you
sort of you'd be with anaplan.
They modified an existing system.
So do you have sort of a set?

Is there a set preset?
I guess that you've that like a set
of software that's been developed
that then is modified and distributed
as required for the clients and
not a full build from the start.
Is that? Yeah, so we definitely have.
We were working at the moment what
we call like a core product and it's
called product has all the features
that basically any client that we would
approach would want certain things.
Like you know we used to mark
student results, wait,
import student results,
way to actually take the test as a student.
So we have this very much a core product
and on occasion when a client comes
along and they're like we need another
feature we might add that feature to
the product or create a new branch
for them that has just that feature.
If no one else would want that feature.
So it's very much creating
one product and then.
Suiting clients needs from that core product?
Yeah,
definitely.
So is this something that's?

Did you guys just handle the the
development on or is this sort of sold as a
software as a service or anything like that?
How is, yeah, I guess what's
that aspect of it looking like?
Yeah yeah,
it's very much a software as a service so.
We recently acquired the UNSW
Global Assessment business,
which they are mostly like things like
test writers and things like that so.
Before what we normally did was
we didn't supply questions.
We took their questions and completely as
a service took the students in Marquette,
made this take the test,
got their Marks and then exported them out.
But recently we've taken a
more holistic approach,
so that things like actually
writing the test questions,
taking this test an this service,
and going to like schools and being like,
hey, do you want to buy?
We can deliver?
Yeah, that's a service to you, yeah, OK.
Jane,
do you have any input?

So I'm curious on kind of how you
got first got into the industry.
So what?
What did you study in high school in
University to acquire the skills
necessary for this drug? Sure,
in high school actually didn't do anything.
I was talk with some of the other day,
but I did. Is it mainly graphics design,
architecture, that kind of thing?
So nothing really relevant there.
Wixon High School or some **** like
that is but yeah, yeah, exactly.
You know as a kid I was always programming
things in her was creating custom
Maps in Warcraft 3 using the coding
engine and that was that was great.
That was more coffee on my ***
**** memory Lane. Yeah, right.
So yeah, it was very like I always.
I was always in programming as a kid's fault.
I didn't actually get into the industry
in terms of tertiary education.
I actually got a diploma of Programming
from SAE which is in the student Sydney.
Uhm?
And then I took that and went to do a
game design game development course
to actually get a degree in that side.

Mostly did games programming.
I kind of thing and then to
actually get into the industry.
Actually started as a tester.
Did software testing for about six months
and then junior development position
opened up and was like hey my new C Sharp.
I've proven myself to be a good employee.
Could I have that job and they just gave
it to me without without even an interview?
And I've been working as
a developer ever since,
which is about four years now,
so I guess it if you had to to
look at your previous programming
experience and also your tertiary ones,
you know what was, I guess,
not a percentage but but how beneficial
did you find the tertiary education?
I guess compared to?
Your own projects and what you are pursuing.
Sort of with self learning, yeah.
Definitely found it very useful.
You know,
I think something that people say a
lot is if you'd learn by yourself
what you're likely to do is just
reinforce your own mistakes,

and I think a big part of what tertiary
education offered me was the ability
to essentially do my own projects.
'cause I was doing game
design and development,
so I was doing a lot of
creating prototype games,
that kind of thing,
and then I would take that project that I
made and took it to the tutor or the teacher,
the lecture or whatever,
and they said to me,
hey, this is good work,
but here are the mistakes you
made and it's things like that
that I found the most valuable in.
University because it allowed
me to correct my mistakes,
whereas I think if I was going up by myself,
I just keep making those dumb
mistakes over and over again.
Yeah,
definitely OK.
So if you really found it beneficial
because it I guess.
The pathways a little bit.
Hammered in some best practices.
That's exactly how I would have put it,
and definitely when I came into the

workforce, a big part of that was, hey,
you've learned these best practices.
That's good, whereas if I was self taught,
I definitely would be making a lot of.
We had code nothing. Yeah.
Student's spaghetti code was
that not all strike back.
That's great stuff.
I don't know, I mean.
There wasn't many incident that was.
Yeah Oh my God, some curious so when
he did come out of University that
was the so the job you currently in
working in software development for,
you know schooling learning.
Was that your first job like
coming out of that or yeah?
Well, my first software company,
I worked stocking shelves or
burning you want to hear about that.
I mean, that's the 1st and then
moved up within the same company.
Yeah, exactly.
That's probably the biggest oddity
as they didn't actually started as a
as a developer have you is have you
done so obviously was this your first?
Uh.

Step into black software I when I say
software I mean non game related programming.
When you when you moved into this
position or was it was there some
projects that you are pursuing
on yourself first?
So I was mostly working in the game
like I was creating a little game
myself but I actually did a few
projects for people you know like
someone that close to the family
that had met somewhere needed
some small programming tasks done
and I did stuff like that.
But it's essentially my first big
proper shippable software type job.
Sure.
And yeah,
I think the company I work at
is quite like we have a lot
of really long term employees.
I think someone celebrating
their 20th year anniversary.
Oh my God next week.
Yeah, that never happens anymore.
That's insane.
Yeah,
right?
My boss just celebrated its 10

year so quite a lot of young
people at my work.
Hey, could you give a couple of
just a couple of quick examples of
those mini side projects you did
for people like all we talk about,
we talk like you know word play
WordPress plugins or like well
yeah things are we I think I think
everyone's had their fair share
of like oh help me I need a word,
a website and you just like log on
to square space making website.
But I think the biggest one was.
A friend of mine was running like a
GPS tracker system and he needed some
help just doing some kind of like
busy work on the side basically and
I took care a lot of that for him.
Things like small programming tasks
just to like logic checks and
stuff like that. So what was that?
Was that, like a custom site, he had an?
Or was that built using something?
It was a piece of like custom
software he was using.
I think it was in C++ actually 'cause a
lot of the early learning edges in C++,

not C sharp. Yeah, OK.

And have you found?

I guess that sort of blends

in nicely to this,

like have you found moving between

languages and learning additional

languages as you gone like?

Is it something that you've once you

started to pick up and click that then

you sound like you're quite confident now?

Jumped into.

Unfamiliar environments, yeah.

I mean,

there's definitely like lineages

of languages like.

Obviously you get like C + C + + C Sharp.

All that kind of things and jumping from

between those is like kind of trivial.

Just need to learn the differences,

but jumping from something like

C + + 2 Java or actually did

some coding in Lua recently,

and there's definitely a learning

curve there, but it's not.

You don't have to learn how to program,

I can use that to learn the syntax.

Basically.

Yeah, OK,

so sort of like learning the

grammar of the language exactly.
The whole language of the start,
yeah.
And so a lot of the same principles
that you would have picked up
and learn for writing efficient
code and things like that would
transfer quite nicely between.
Surprisingly enough,
that's actually one of the things that
are hardest to learn between link.
OK, like writing efficient code.
There's a lot of like base level things
that you can do right things efficiently,
but a lot of languages provide
a way to write efficient code,
like a great Wellings is like
no interfaces and stuff in C,
sharp is quite a common C sharp practice.
So translating that something
like lure is less applicable.
OK,
So what is an interface 'cause it's
probably beyond my understanding,
so I don't know.
I don't know what you guys are
in terms of understanding,
but in the basis terms it's kind of like.

A menu or an index page for a object.
So if an object can do things
like export users,
download users. Process uses, for example,
the interface will have those three on it,
but it won't actually have any code
on how to do it or just say, hey,
the thing I represent can do these things,
and this interface could represent
like 4 different objects.
One could be like a NAPLAN student processor,
the next one could be a RMS student
processor and by having that one interface,
you essentially saying I don't
care what actually process is this.
I just know that it can process
it in these three ways.
Right, so what's probably a
bit of a silly question,
but what what's the advantage of that?
So it means your code can be more
generic like it means I don't care
what client we're working with when
a student finishes their test,
I can now call the process student.
An interface, right?
And if it's working for the NAP line,
it'll go and do one thing.
If it's working for a Mexican girl

with completely different thing,
but I can call the same.
The same function, yeah, exactly.
How does it know that which one is?
It's called off.
You can essentially register different
implementations to the same interface,
so you like both of the,
like the NAPLAN, an the RMS.
Implementation will both
implement that same interface,
so just don't like you can switch
out which one it actually calls.
Yeah, OK, so how do these?
How do they actually look like I'm?
I'm just sort of trying to
visualize them a little bit here.
'cause so we sort of got classes
and functions within classes,
so how what's an interface
in relation to those?
Yeah, it's uh, if you think about a class,
it's essentially a class with
a bunch of functions on it,
but the functions don't have anybody's there,
just the names of the functions and
the arguments with the function.
So you just gotta say like process

student and that will have an argument,
which is like a string that is student ID.
But it won't have anybody.
And then as part of the class
itself that implements it,
it'll inherit the interface.
So you'll have, like NAPLAN student thing,
and then we'll have that same
function with the same arguments,
but this time it will have a body,
and then if you call the interface that
are called that function on the child.
OK, so it is.
Yeah, OK,
so you sort of call it in a similar
capacity to class functions.
Yeah, exactly.
It's it's more of a simplified way to handle.
Yeah, it's like a metaclass,
which could have three or four actual
different classes that sit beneath it.
Right, I'm following you and then yeah,
OK, depending on what what it
is that you're calling.
It will forward the data onto
yeah crippens each one of.
Yeah, so would you have like one for each?
Client and invited forward based on that?
Or is it not?

It's not.
Wouldn't be that specific?
That's a pretty common use case abuse
case we had recently as we with all the
lockdown we've had a lot of new clients
come aboard and they use different.
Like monitoring systems like to
make sure the students don't
cheating or anything like that,
so we've got an interface
that is like monitor,
student and then we have like.
Different sub clients like we have a
place called Procter Track which keeps
track of like if a students cheating
or whatnot and there will be different
implementations of the interface.
Yeah, OK. How does something
like that even work though?
Like it like, how do you?
Is it you just looking for like I need?
How would you even check to see if
someone's you know to mean got like a
second second tab open or something?
Thanks very high tracking.
It uses their webcam and a lot of
like low level stuff in the computer.
It's kind of like This is why we

hire a second company to do it,
but it's pretty Big Brother.
Pretty shady.
Yeah yeah.
Yeah wow,
I like throwback to Jayden for
some questions.
So you explained that when coming
out of University you had a lot of
skills that were able to be used in a
team environment such as having cleaner code,
being able to communicate that code
more effectively with other people.
Do you find people that may self
study or learn by themselves?
Are they easier to point out
and like is that like a big?
Yeah.
Like we had someone start kind
of recently and I was actually in
charge of trading that person and it
was really obvious that they were
mostly self taught because they just
didn't know any of the terms right there.
Be like you know it's the one with
the the curly thing and it does the
thingy thing and I'm like you mean.
That like a function and they're like,
yeah, that thing. And they they could.

They could code just fine, right?
But it was really hard to communicate.
Different like if I teach them a new concept.
I basically had to teach them all of
the concepts that led up to that and
they would understand most of them
'cause they had been using them already,
but they just didn't know warm specifically.
Like interfaces for example,
I might say, like, oh,
you can use interface to do that,
to teach them what interface was,
even though they already used one already.
Yeah yeah,
OK makes sense.
Change so I understand that you
work a lot in like learning and
different like schooling environments.
Understand that your brother mentioned to
me that you made a learn to code game.
I'm not sure if I'm like remembering this.
Right now you're hearing it correctly.
Have you always wanted to try and work within
the learning education side of code, or?
Oh yeah, I think the stupid term that they
use in my University was edutainment.
Yeah right, it's horrible,
and it's always been in the area that really

interested me because I like teaching.
I like education and I like
programming video games,
so it's been interesting to me,
but I've always struggled to find
that line between fun and teaching.
Like it's very easy to make game that's fun
and vaguely uses the concepts of programming.
And it's also easy to make a game
that's just programming.
But it's also gaming, but it's hard to find
that middle line which teaches someone.
How to do something while still being fun?
Yeah, OK. I feel like it comes to user.
Yeah, like you guys have probably
done some like coding games right?
Like very much like you code.
How an AI works in it.
Yeah I mean I I tried to play station is
with one of my engineering friends and I'm
going to be honest I didn't type it in it.
It was too hard.
It was too far beyond the scope of,
you know,
because MIPS is extremely low level.
Yeah, that literally just uses like
a virtual Maps so it's like now.
I'm just going to leave this to you.
And I'll just do the housework.

Yeah, Ann.

Games like that definitely just like help you

refine your programming while playing a game,

and there's the other type of game

which is like it uses vague scripting

concepts like you have to like.

Give a thing a piece of instructions.

So you learn basically how

to think like a programmer,

how to think about doing things in steps,

how to think about using conditions,

but they don't actually

teach you how to code,

and it's very hard to find

a middle ground there,

and I have a couple ideas

that I'd like to try,

but it's all very high concept at this point.

So what are your thoughts on something like

like a GML or game Maker language with their?

I guess really relaxed syntax to try and

lower the barrier of entry for people.

Yeah, I see there's a lot of

value in in as a teaching tool.

Or is it sort of just what?

Yeah,

I guess what's your thoughts on it?

Yeah, definitely like.

It's kind of hard for me 'cause I've always
worked in like quite quite hard languages.
Things like I originally coded in C.
C + + C Shop that a little bit in JavaScript,
but it's only recently that have really
touched on like hyper relaxed languages,
but I definitely think that like.
There better environment to teach someone
in 'cause you don't have to explain like
technically this is like a pointer.
So when you're using it,
you have to think about it in different ways.
Yeah yeah it gives them the option
to to make a couple of you know
grammatical errors and still have
you know functional code as well.
That wink yeah 22 strict.
OK this in a like an example
of that is that like the Unreal
Engine like Blueprint type setup.
Is that what you mean by visual scripting?
Note that visual scripting or
notes in Blender as well does.
I guess that's the same purpose.
Yeah, it's it's interesting.
You bring that up.
'cause that's kind of what
I mean when I say like.
I think if you were to create a this

is teaching to someone to be a program
without ever having them programmed before.
I think you'd want to use
something like that, 'cause like.
You'd want them to get get into the
mindset before they get into the syntax,
and I think things like scripting
languages are great for getting
people into the mindset.
And if you go back to that programming game
that I made my brother play at one point,
it was very much scripting, not programming.
Yeah, OK.
Did the game end up going
anywhere after about that story,
I think what happened was I
I spent a while making it.
I showed it to Steve and he shat on
it and I threw it in the garbage.
Sounds like Steven.
Yep, yeah.
The story of dreams.
He was very much his response to
that game was what made me formulate
my opinion on the fact that,
like.
You need to Mike again so the fun
and light or fun and very deep

'cause I was trying to strike that
middle ground and it wasn't that
fun so I think that's what made me
like come to the conclusion that.
Making a game that actually teaches
someone programming from scratch
and be fun is a very hard task.
I think it applies like any education,
not just programming.
Yeah, definitely well.
What what did the start of the
game sort of look like then?
I guess?
How were you trying to convey?
The concepts, so the gameplay itself
was akin to like age of empires.
That is very much like you would like
a different feel, like a Town Center,
there were Gold mines,
award etc etc.
And the way you made the game was
you couldn't control your dudes,
but you could scriptum.
So you could tell them like hey
you want to go to the gold mine
then come back and then go to
Goldman and come back etc.
And it taught people things like
conditional logic and loops and

right so you could you could go
like 40 while collecting like as in.
True yeah,
move to position gold mine and then
come back type of thing exactly
really simplified but still having
the same steps that would be required.
In a programming language,
to do that sounds good,
yeah,
and if you've ever looked at
something like Google,
does this a bunch where they
create like scripting games
where like you essentially have a
bunch of different commands that you
can drag onto a list rather than
having to type out if etc etc etc.
It had that kind of concept so you
could like drag, go to and then drag
object and then you click object and
then literally click the gold mind.
It would say go to gold mind right?
Yeah they make it used to have at
your base and more stuff to that.
Yeah, it was very much based
on like like I mentioned back.
In Warcraft three,

let kind of scripting language.
Yeah, cool, that sounds well actually,
but what he? Your brother, wasn't a fan.
Obviously he's like this is boring.
I'm like, yeah, you're right.
And I think there's around
that time that I was like.
I was doing something at the time so
I didn't have much time to work on it.
It's definitely on my list
of things to dig up there.
Yeah,
and one of the most advanced things
that I never finished the game,
but I I made a really crummy game in
game Maker that you could actually like.
Select a unit and then tell him to
like walk to a like a gold mine,
but I called them iron ore mines
and then he would like collect
stuff and then walk back.
But the code like looking on
reflection now was absolutely
atrocious for the way I achieved that.
But it was.
It was so advanced back then I was
like blown away that I had managed to do it.
Yeah, trust me as a programmer. Yeah,
if you look at code more than six months old,

you hate it.

You feel like this is the worst

thing I've ever done.

Well mate, why did I do this?

Yeah?

And then six months later you'll think

something about your code back then.

Well,

I was already saying that about the first

program assignment that we submitted.

I've looked back at that and I'm like,

oh, why did I do it like this?

It's like a 10 times longer

than it needed to be,

and it could have been condensed and it

just could have been so much better, yeah?

You're learning that comes up all the time.

You're like,

why did I not make this an if statement

instead of typing it out 10 times?

Yep.

Why did I misspell the quiz?

What I really hope that the teacher

read the menu so that they saw

you needed like quiet to leave,

not quit.

Otherwise they totally gonna

be stuck in that game.

My my my I've liked the worst spelling
out of everyone and every time I
put out a code review to my team,
they're like you spelled
these five variables wrong.
I'm like, yeah, they're variables, who cares?
Yeah, So what do you use to program?
We use Visual Studio like kind of
the stock standard Microsoft one.
Yeah, but.
It's trash.
OK,
so what's your preference then?
Provincial studio.
Trash, but it's still I I've used
it long enough that I can like
finesse it so it doesn't crash
into me every 10 minutes and it's
got a lot of robust features,
good debugging, good Intellisense,
that kind of thing, but it's. Painful.
It's like a big, strong, bloated thing,
hard to use, but if you know how to use it,
it's very powerful. Yeah, OK?
Gemini public questions Jen. Uhm so.
The son he's like getting into coding
or like maybe in high school right now.
Would you recommend games that try and teach
you how to code like an old while ago at

playing a little bit of script with you?
Which kind of felt like getting thrown
in the deep end. So yeah, that was.
So how did games that?
Grapes, grapes. What do you do?
It's very, very similar to like a StarCraft.
If it was, you needed to code.
Yo minions, yeah, OK doing that stuff.
Honestly, the game I was trying to
make but more programming, yeah.
Yeah, it's hard for me to recommend.
Like I was doing a lot of scripting,
a lot of light programming before I learned
how to program at University, right?
So I have essentially my basis is
informal education from where I learn
to like write if statement, right so?
When it comes to like how to
get started with programming,
my best recommendations are sites
like there a lot of games like Code
game as I believe the website name.
It's very basic where it literally
teaches you.
Peter High destroy these enemies by typing.
If enemy shoots right, Sir,
I think I'm a very beginner thing
like that might be the way to go,

but it's hard for me to say honestly.

Yeah.

Under the letter script,

first you know,

yeah.

Alright, I'm also I'm actually curious.

Do you think 'cause you

did a game development?

Yeah, it's great. I loved it.

It was probably not the most

practical thing if I'm honest.

I really enjoyed it.

Yeah, OK, it was a great for learning

'cause there's a lot of like.

There's a lot of self teaching.

A lot of like Go Creek this game

and tell me about the problems you

ran into while creating this game,

and I found that really enjoyable

and a great way to learn.

It's like if you have a company

like I'm a game programmer like

we need a software engineer.

I'm like **** we need anything

else like shucks did you find?

I guess this probably comes down to

a bit of personal preference but.

Is there any?

Did you ever find that being

added visually see?
The program that you're writing be reflected.
You know,
like in character movement
or something like that.
Was that you know quite rewarding
or beneficial, to be honest.
Actually get that proper visual feedback
that this is what your program is doing.
But then this is the actual you
know you can physically see it
moving around and doing stuff.
Yeah,
like it's super practical.
I definitely think that supposed
to be practicality.
I think one of the best things was
debugging a game is such a unique
experience because it's like.
K.
Why did that guy jump a 300 meters
forward when I press W right and
**** debugging stuff like that
really teaches you how to think
critically in a way that I think
even like I debug kernel that I'm
at my work and I think debugging a
website is a very different beast

than if you think of it like a game,
and I think that's probably one
of the best skills I learned in
game development was debugging.
OK, that's a really good takeaway, actually.
Right?
That's awesome, yeah.
This will wrap up their cool.
Yeah, Jayden has me on discord.
If you want to either you
wanna ask me anymore questions
I'm always open for anything.
Yeah, brilliant thank you for that.
That's really insightful cool.
I will wrap up the right things
come in problem. Thank you, bye.
Just openly wait for him to leave.
Yeah, yeah. OK, that was good.
That was actually pretty good, yeah?
Went for a. You know actually much
longer than I thought it would go,
which I'm happy about. That's good.
Yeah, that's Jack.
I can send you think why I got forgetting
his job was right place, right time.
That was the takeaway?
Yeah yeah, let's go to mention it
like I was gonna bring up late so
well but it's like as soon as he said

he kind of just got a lucky break.
It was like yeah, well ****.
Now that was good. I think that
that really fulfilled all of
the interview requirements.
Alright, yeah. Time to slowly go
through the transcript. Cool, yeah,
I'm just going to have some dinner.
'cause I'm yeah, I'm not.
I'm not actually calling for today,
but yeah, that was good.
That was good cool I'll get tomorrow.
I'll finish ideal jobs, yeah,
and the template pages and they'll
push that all back to GitHub as well.
And if you're on tomorrow,
I'll help with the project
stuff so we can get that. Yeah,
I'm trying to listen as much as I can, yeah?
Unfinished as an uh maybe Monday oh.
Go through the transcript I guess yeah.
But we do that all.
You know,
try to do that because looking at how it's
kind of laid out potentially we've got.
A week from tomorrow.
Yes, I kinda wanna get it done yet so much
as we can to make a bit of a boogie on.

Uhm?

Yeah, it's interesting.

What is discord?

And if we need,

it will send it will consent if we

later and try to send through that

picture tomorrow and yourself as well.

So I can throw that in, yeah?

Like a yeah,

alright catch you later.

Alright take care.