

I strongly recommend not reading this, but instead DESCRIPTION.lean file, this is just copied from there.

/-This project is about planar Euclidean Geometry defined via complex numbers.
Pretty much everything I did, I proved from scratch, i.e. did not use mathlib (only sometimes rarely for small numerical lemmas).

Therefore, a big part of the project is defining geometric structures and proving elementary propositions about them,
which often seem quite trivial. Nonetheless, more advanced stuff was proved as well.-/

/-I originally set out to prove Feuerbach's theorem (Incircle and Nine-Point-circle of a triangle are tangent) within this framework,
however this turned out to be much too difficult for now, mainly because defining the incircle is quite problematic.-/

/-The entire project spans over 14.000 lines of code. Though, I suspect much of this could be saved with more efficient code, as there are a few redundancies and so on.-/

/-Now a brief description of what every file does in the order of their dependancies:-/

/- # Basic.lean -/ --Introduces points, basic properties about them and what it means for points to be colinear (+ properties)

/- # Lines.lean -/ --Introduces lines (basically: Set of colinear points) and shows basic properties (i.e. there is a unique line between two (disjoint) points)

/- # Parallels.lean -/ --Introduces parallels. Lines are said to be parallel if one can be obtained by "shifting the other". This is proven to be equivalent as being the same or disjoint (tedious!). On top of that, the parallel postulate is proven.

/- # qObject.lean -/ --Rather short file mostly simplifying "notation", i.e. introduces a simplified function for taking two points to the line between them, not requiring them to be disjoint. No actual mathematical results are proven.

/- # Triangles.lean -/ --Introduces triangles (3 noncolinear points) and areas of three points / triangles. Furthermore an important notion of a point being in between two others is introduced.

/- # Circles.lean -/ --Introduces circles and proves basic properties about them, i.e. uniqueness of radius and center

/- # Perpendiculars.lean -/ --Introduces perpendicular lines (and foot of a point on a line) and proves basic stuff about them, i.e. the corresponding "parallel postulate" for them or that being twice perpendicular is the same as being parallel.

/- # Pythagoras.lean -/ -- Proves Pythagoras theorem (not hard the way I defined geometry) and shows some direct consequences. Pythagoras theorem later becomes quite useful for various tasks.

/- # Auxiliary.lean -/ --Small file doing nothing important. Mainly the notion of lying on the same side of a line and lying inside a triangle is introduced, however nothing important is being proved.

/- # Circumcircle.lean -/ --Three noncolinear points lie on a unique circle ! This result is proven via the usual characterization of the perpendicular bisector.

/- # Reflections.lean -/ --Introduces reflection along points and lines and proves stuff like "reflecting twice" gives the original.

/- # Lineartrans.lean -/ --A not-so-interesting file showing that under $x \mapsto a*x + b$ for $a \neq 0$, pretty much every geometric structure is preserved. I only used this in very few occasions, but this would be a key technique necessary to proving complicated stuff analytically, as it basically allows us saying "wlog". This could be done via group actions, but I was too lazy.

/- # Tangents.lean-/ --Introduces tangents to circles and shows stuff like uniqueness of a tangent to a fixed point on the circle. Pythagoras theorem was needed for this.
 /- # CTangent.lean-/ --Introduces the notion of two circles being tangent to each other and proves consequences about colinearity and radii etc.
 /- # Powpoint.lean-/ --Introduces the power of a point in respect to the circle. In particular, for two nonconcentric circles the set of points with same power to them form a line, the "Powline" as I called it, which is very useful.
 /- # Angles.lean-/ --Introduces angles and proves basic stuff. I am not extremely happy with how everything was done, but they do the job just fine. Angles are only occasionally from this point on.
 /- # Thales.lean-/ --Proves both directions of Thales theorem. For the first one a (in Lean) painful angle chase was used, for converse an argument avoiding angles I thought of (via parallel lines / rectangles).
 /- # Orthocenter.lean-/ --Proves the existence of the orthocenter (as the intersections of the altitudes). This is really not that trivial: I used (converse of) Thales theorem combined with power of points, giving a neat proof.
 /- # Similar.lean-/ --Introducing similar triangles in several ways (orientation or not matters!) via functions described in Lineartrans.lean. Some characterizations, like angles being the same implying the triangles to be similar.
 /- # Ceva.lean -/ --By far the coolest thing I proved: Ceva's theorem, both directions in what is probably the most general form for the planar form. Proof was mainly the standard argument with areas, which while simple, is a bit annoying to set up. 2.7k lines (by far the largest file)

/- # Incircle.lean, Hilbert.lean-/ -- are (very!) unfinished and do not contain anything relevant (and nothing depends on them).

/- Sometimes theorems are not at the place they logically should be located due to various reasons, but this is not very frequent.-/

/-A small collection of some of the main/most interesting results I obtained:-/

#check colinear_trans --Basically: being colinear is transitive. This was the most central and hardest piece setting up the geometry.
 #check line_through_unique --Very useful: Given two points not equal, the line through them is unique
 #check parallel_def -- Being parallel is equivalent to being the same or not sharing a point (to appreciate this, one might want to check out my definition of Parallel)
 #check parallel_postulate --The parallel postulate!
 #check circle_unique -- Uniqueness of center and radius of a circle.
 #check perp_bisectors_copunctal -- Perpendicular bisectors of noncolinear points are copunctal
 #check circle_around_unique -- Therefore, there is a unique circle on which these three points lie
 #check tangent_is_perp -- A tangent to a circle is perpendicular to the radius
 #check tangent_through_unique --Therefore (use parallels) there is a unique tangent on which a point on a circle lies on
 #check coutside_ctangent --Two circles are tangent from the outside iff their radii add up to the distance of their centers.
 #check PowLine -- Existence of the power of point line.
 #check powline_perp -- Which is perpendicular to the line going through the centers of the circles
 #check anglesum_points --Angles sum up to π in a triangle
 #check angle_zero_imp_colinear --The title says it all (there's various versions of this)
 #check thales_theorem --Thales theorem, first direction
 #check thales_inverse -- And the converse

#check altitudes_copunctal --The altitudes of a triangle are copunctal, implying the existence of the orthocenter

#check aaa_dsimilar --Same angles imply being similar (theres 2 version for this, depending on orientation)

#check sas_dsimilar_a -- side-angle-side similarity (also multiple versions)

/-And once again, the by far most interesting result, which is also technically on the Freek 100 page, Ceva's theorem:-/

#check ceva

/-Above is the most general version. One direction / special case with another formulation that might be worth checking out is-/

#check ceva_spec

/-In general, I used "lemma" for the majority of results I proved and "theorem" only for those, which seemed significant.-/

/-Except for the two (unused) files I mentioned earlier, there should not be any sorry's or unfinished stuff.

Pasch' Axiom would have been nice to have in Auxiliary.lean (and maybe needed for the incircle), but oh well...-/

/-As the math behind all of this is quite simple, I did not really use any outside references/sources, although

for a small amount of formulas / definitions (maybe about 3 in total), I relied on the chapter about Complex Numbers of the

famous "Euclidean Geometry in Mathematical Olympiads" book by Evan Chen.-/

/-The powerpoint slides of my presentation are found in the Powerpoint.pdf file!-/