

KRONOS

MODULTESTPLAN



PrecipitationTypeFilter.cpp

TestPrecipitationTypeFilter.cpp	This filter can extract data from precipitation point sets read by a Kronos reader depending on the precipitation type of each point.
TestPrecipitationTypeFilter	Read some test data that is useful for testing precipitation types Set up the filter and its input
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Test the filter's output without excluding any precipitation type	res/test-data/precipitation-types.kJson	expectedPointAmount = 400 (the whole test dataset contains 400 points)	400	✓
Exclude data points without precipitation type information	res/test-data/precipitation-types.kJson QList<PrecipitationDataPoint::PrecipitationType>() << PrecipitationDataPoint::NONE	expectedPointAmount = 320 (the test data set contains 80 data points without precipitation data)	320	✓



Description	Input	Expected Result	Result	Success
Additionally, exclude data points with the precipitation type rain	<pre>res/test-data/precipitation- types.kJson QList<PrecipitationDataPoint ::PrecipitationType>() << PrecipitationDataPoint::NONE << PrecipitationDataPoint::RAIN</pre>	<pre>expectedP ointAmou nt = 240 (the test data set contains 80 rain data points)</pre>	240	✓
Additionally, exclude data points with the precipitation type snow	<pre>res/test-data/precipitation- types.kJson QList<PrecipitationDataPoint ::PrecipitationType>() << PrecipitationDataPoint::NONE << PrecipitationDataPoint::RAIN << PrecipitationDataPoint::SNOW</pre>	<pre>expectedP ointAmou nt = 160 (the test data set contains 80 snow data points)</pre>	160	✓
Additionally, exclude data points with the precipitation type sleet	<pre>res/test-data/precipitation- types.kJson QList<PrecipitationDataPoint ::PrecipitationType>() << PrecipitationDataPoint::NONE << PrecipitationDataPoint::RAIN << PrecipitationDataPoint::SNOW << PrecipitationDataPoint::SLEE T</pre>	<pre>expectedP ointAmou nt = 80 (the test data set contains 80 sleet data points)</pre>	80	✓



Description	Input	Expected Result	Result	Success
Additionally, exclude data points with the precipitation type hail, therefore eliminating all data points	<pre>res/test-data/precipitation- types.kJson QList<PrecipitationDataPoint ::PrecipitationType>() << PrecipitationDataPoint::NONE << PrecipitationDataPoint::RAIN << PrecipitationDataPoint::SNOW << PrecipitationDataPoint::SLEE T << PrecipitationDataPoint::HAIL</pre>	<p>all data points have been removed:</p> <p>expectedPointAmount = 0</p>	0	✓
Finally, enable everything again, which should give us back all points we started with in the first place	<pre>res/test-data/precipitation- types.kJson</pre>	400	400	✓



SphericalToCartesianFilter.cpp

TestSphericalToCartesianFilter.cpp	The SphericalToCartesianFilter class is a vtk filter which is able to convert the coordinate system of the data. The input of a SphericalToCartesianFilter is a data set which uses gps data (latitude, longitude, height). These are converted to cartesian coordinates.
TestSphericalToCartesianFilter	Insert some points on a 2D plane Then do the transformation Check if the transformed points are equal to the expected transformed points
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Test transformation of 0,0,100	points->InsertNextPoint(0, 0, 100);	expectedPointsAfterTransformation->InsertNextPoint(0, 0, 200);	0,0,200	✓
Test transformation of 90,0,100	points->InsertNextPoint(90, 0, 100);	expectedPointsAfterTransformation->InsertNextPoint(200, 0, 0);	200,0,0	✓
Test transformation of 0,90,10	points->InsertNextPoint(0, 90, 10);	expectedPointsAfterTransformation->InsertNextPoint(0, 110, 0);	0,110,0	✓



Description	Input	Expected Result	Result	Success
Test transformation of -90,0,100	points->InsertNextPoint(-90, 0, 100);	expectedPointsAfter Transformation->InsertNextPoint(-200, 0, 0);	-200,0,0	✓



TemporalAggregationFilter.cpp

TestTemporalAggregationFilter.cpp	Aggregates temporal data points.
TestTemporalAggregationFilter, TestTemperatureData	Read some test data Set up the filter and its input Run the filter on the input data Extract the filter's output Test the data that the filter has produced
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
	res/test-data/temporal-aggregation-test/temperature-test-data.kJson temperatureArray->GetNumberOfTuples()	2	2	✓
	res/test-data/temporal-aggregation-test/temperature-test-data.kJson temperatureArray->GetTuple(0)	8.874	8.874	✓
	res/test-data/temporal-aggregation-test/temperature-test-data.kJson temperatureArray->GetTuple(1)	16.42	16.42	✓



TemporalAggregationFilter.cpp

TestTemporalAggregationFilter.cpp	Aggregates temporal data points.
TestTemporalAggregationFilter, TestCloudCoverageData	Read some test data Set up the filter and its input Run the filter on the input data Extract the filter's output Test the data that the filter has produced
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
	res/test-data/temporal-aggregation-test/cloud-coverage-test-data.kJson cloudCoverageArray->GetNumberOfTuples()	2	2	✓
	res/test-data/temporal-aggregation-test/cloud-coverage-test-data.kJson cloudCoverageArray->GetTuple(0)	0.626	0.626	✓
	res/test-data/temporal-aggregation-test/cloud-coverage-test-data.kJson cloudCoverageArray->GetTuple(1)	0.17	0.17	✓



TemporalAggregationFilter.cpp

TestTemporalAggregationFilter.cpp	Aggregates temporal data points.
TestTemporalAggregationFilter, TestWindData	Read some test data Set up the filter and its input Run the filter on the input data Extract the filter's output Test the data that the filter has produced
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Test the aggregation of wind speeds	res/test-data/temporal-aggregation-test/wind-test-data.kJson windSpeedsArray->GetNumberOfTuples()	2	2	✓
Test the aggregation of wind speeds	res/test-data/temporal-aggregation-test/wind-test-data.kJson windSpeedsArray->GetTuple(0)	0.782	0.782	✓
Test the aggregation of wind speeds	res/test-data/temporal-aggregation-test/wind-test-data.kJson windSpeedsArray->GetTuple(1)	0.7766667	0.7766667	✓
Test the aggregation of wind direction	res/test-data/temporal-aggregation-test/wind-test-data.kJson windDirectionsArray->GetNumberOfTuples()	2	2	✓
Test the aggregation of wind direction	res/test-data/temporal-aggregation-test/wind-test-data.kJson windDirectionsArray->GetTuple(0)	244	244	✓



Description	Input	Expected Result	Result	Success
Test the aggregation of wind direction	res/test-data/temporal-aggregation-test/wind-test-data.kJson windDirectionsArray->GetTuple(1)	28	28	✓
test the wind velocities that have been computed from the aforementioned two for flow visualisation	res/test-data/temporal-aggregation-test/wind-test-data.kJson windVelocitiesArray->GetNumberOfTuples()	2	2	✓
test the wind velocities that have been computed from the aforementioned two for flow visualisation	res/test-data/temporal-aggregation-test/wind-test-data.kJson windVelocitiesArray->GetNumberOfComponents()	3	3	✓



Description	Input	Expected Result	Result	Success
test the wind velocities that have been computed from the aforementioned two for flow visualisation	<pre>res/test-data/temporal- aggregation-test/wind-test- data.kJson windVelocitiesArray- >GetTuple3(0) [0]</pre>	-0.702857 02	-0.702857 02	✓
test the wind velocities that have been computed from the aforementioned two for flow visualisation	<pre>res/test-data/temporal- aggregation-test/wind-test- data.kJson windVelocitiesArray- >GetTuple3(0) [1]</pre>	-0.342806 1	-0.342806 1	✓
test the wind velocities that have been computed from the aforementioned two for flow visualisation	<pre>res/test-data/temporal- aggregation-test/wind-test- data.kJson windVelocitiesArray- >GetTuple3(0) [2]</pre>	0	0	✓



Description	Input	Expected Result	Result	Success
test the wind velocities that have been computed from the aforementioned two for flow visualisation	res/test-data/temporal-aggregation-test/wind-test-data.kJson windVelocitiesArray->GetTuple3(1)[0]	0.36462292	0.36462292	✓
test the wind velocities that have been computed from the aforementioned two for flow visualisation	res/test-data/temporal-aggregation-test/wind-test-data.kJson windVelocitiesArray->GetTuple3(1)[1]	0.68575597	0.68575597	✓
test the wind velocities that have been computed from the aforementioned two for flow visualisation	res/test-data/temporal-aggregation-test/wind-test-data.kJson windVelocitiesArray->GetTuple3(1)[2]	0	0	✓



JsonReader.cpp

TestJsonReader.cpp	Reads our custom kJSON documents and loads them in our model.
TestJsonReader, ReadCityData	Read some test data into our model
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Check string of first city	res/test-data/cities.json firstDataPoint->getName()	"Los Angeles"	"Los Angeles"	✓
Check latitude of first city	res/test-data/cities.json firstDataPoint->getCoordinate().lat()	34.052223	34.052223	✓
Check longitude of first city	res/test-data/cities.json firstDataPoint->getCoordinate().lon()	-118.242775	-118.242775	✓



JsonReader.cpp

TestJsonReader.cpp	Reads our custom kJSON documents and loads them in our model.
TestJsonReader, ReadTwitterData	Read some twitter test data into our model
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Check author of test data point	res/test-data/tweets.json testDataPoint->getAuthor()	"elonmusk"	"elonmusk"	✓
Check content of test data tweet	res/test-data/tweets.json testDataPoint->getContent()	"Is this working?"	"Is this working?"	✓
Check time stamp of test data tweet	res/test-data/tweets.json testDataPoint->getTimestamp()	1439280065	1439280065	✓
Check latitude of test data point	res/test-data/tweets.json testDataPoint->getCoordinate().lat()	34.052223	34.052223	✓
Check longitude of test data point	res/test-data/tweets.json testDataPoint->getCoordinate().lon()	-118.242775	-118.242775	✓



JsonReader.cpp

TestJsonReader.cpp	Reads our custom kJSON documents and loads them in our model.
TestJsonReader, TestPointCoordinatesInVtkPolyData	Read some test data into our model
All Tests Succeeded	✓

Description	Input	Expected Result	Expected Result	Success
Checks if number of cities on high zoom level (9) is right	res/test-data/cities.json polyData->GetNumberOfPoints()	4	4	✓
Checks if number of cities on low zoom level (0) is right	res/test-data/cities.json lessPolyData->GetNumberOfPoints()	2	2	✓
Checks if the x-position of the generated vtkPolyData is equal to the original longitude	res/test-data/cities.json testPointCoordinates[0]	-122.4183 35	-122.4183 35	✓



Description	Input	Expected Result	Expected Result	Success
Checks if the y-position of the generated vtkPolyData is equal to the original latitude	res/test-data/cities.json testPointCoordinates[1]	37.775	37.775	✓
Checks if the z-position of the generated vtkPolyData is equal to the original height (height is 0 in dataset, because heightmap is added from other source)	res/test-data/cities.json testPointCoordinates[2]	0	0	✓



JsonReader.cpp

TestJsonReader.cpp	Reads our custom kJSON documents and loads them in our model.
TestJsonReader, WriteCitiesToVtkPolyData	Read some test data into our model Import the test data to vtkPolyData Check if the vtkPolyData has been generated correctly
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Check number of cities in array	res/test-data/cities.json cityNameArray->GetNumberOfValues()	4 (number of tests cities in file)	4	✓
Check name of second city in array	res/test-data/cities.json cityNameArray->GetValue(2)	"San Francisco"	"San Francisco"	✓
Check number of priorities in array	res/test-data/cities.json priorityArray->GetNumberOfTuples()	4 (number of tests cities in file)	4	✓
Check priority of San Francisco (because it is a big city it needs to have the highest priority available)	res/test-data/cities.json priorityArray->GetValue(2)	Configuration::getInstance().getInteger("dataReader.maximumPriority") - 1	9 (depends on configuration file)	✓



JsonReader.cpp

TestJsonReader.cpp	Reads our custom kJSON documents and loads them in our model.
TestJsonReader, WriteFlightsToVtkPolyData	Read some test data into our model Import the test data to vtkPolyData Check if the vtkPolyData has been generated correctly
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Check the number of components in test data set	res/test-data/flights.json destinationArray->GetNumberOfComponents()	2	2	✓
Check destination coordinate's latitude	res/test-data/flights.json destinationArray->GetTuple2(0)[0]	34.052223	34.052223	✓
Check destination coordinate's longitude	res/test-data/flights.json destinationArray->GetTuple2(0)[1]	-118.242775	-118.242775	✓
Check the number of components in test data set	res/test-data/flights.json originAirportCodesArray->GetNumberOfValues()	1	1	✓
Check origin airport code	res/test-data/flights.json originAirportCodesArray->GetValue(0)	"DXB"	"DXB"	✓



Description	Input	Expected Result	Result	Success
Check the number of components in test data set	res/test-data/flights.json destinationAirportCodesArray->GetNumberOfValues()	1	1	✓
Check destination airport code	res/test-data/flights.json destinationAirportCodesArray->GetValue(0)	"LAX"	"LAX"	✓
Check the number of components in test data set	res/test-data/flights.json airlineArray->GetNumberOfValues()	1	1	✓
Check airline name	res/test-data/flights.json airlineArray->GetValue(0)	"Lufthansa"	"Lufthansa"	✓
Check the number of components in test data set	res/test-data/flights.json flightLengthArray->GetNumberOfComponents()	1	1	✓
Check length of flight	res/test-data/flights.json flightLengthArray->GetValue(0)	12380.654	12380.654	✓
Check the number of components in test data set	res/test-data/flights.json priorityArray->GetNumberOfTuples()	1	1	✓
Check flight priority	res/test-data/flights.json priorityArray->GetValue(0)	Configuration::getInstance().getInteger("dataReader.maximumPriority")	10 (depends on configuration file)	✓





JsonReader.cpp

TestJsonReader.cpp	Reads our custom kJSON documents and loads them in our model.
TestJsonReader, WriteTweetsToVtkPolyData	Read some test data into our model Import the test data to vtkPolyData Check if the vtkPolyData has been generated correctly
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Check the number of components in test data set	res/test-data/tweets.json authorsArray->GetNumberOfValues()	3	3	✓
Check author	res/test-data/tweets.json authorsArray->GetValue(1)	"elonmusk"	"elonmusk"	✓
Check the number of components in test data set	res/test-data/tweets.json contentsArray->GetNumberOfValues()	3	3	✓
Check content	res/test-data/tweets.json contentsArray->GetValue(2)	"This volcano looks like it is erupting #lava #dangerous"	"This volcano looks like it is erupting #lava #dangerous"	✓
Check the number of components in test data set	res/test-data/tweets.json timestampArray->GetNumberOfTuples()	3	3	✓



Description	Input	Expected Result	Result	Success
Check timestamp	res/test-data/tweets.json timestampArray->GetValue(2)	1439288745	1439288745	✓
Check the number of components in test data set	res/test-data/tweets.json priorityArray->GetNumberOfTuples()	3	3	✓
Check priority	res/test-data/tweets.json priorityArray->GetValue(2)	Configuration::getInstance().getInteger("dataReader.maximumPriority") - 1	9 (depends on configuration file)	✓



JsonReader.cpp

TestJsonReader.cpp	Reads our custom kJSON documents and loads them in our model.
TestJsonReader, WritePrecipitationToVtkPolyData	Read some test data into our model Import the test data to vtkPolyData Check if the vtkPolyData has been generated correctly
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Check the number of components in test data set	res/test-data/precipitation.json precipitationRateArray->GetNumberOfComponents()	1	1	✓
Check precipitation rate	res/test-data/precipitation.json precipitationRateArray->GetValue(0)	24	24	✓
Check precipitation type	res/test-data/precipitation.json precipitationTypeArray->GetValue(0)	PrecipitationDataPoint::SNOW	PrecipitationDataPoint::SNOW	✓
Check the number of components in test data set	res/test-data/precipitation.json timestampArray->GetNumberOfTuples()	1	1	✓
Check timestamp	res/test-data/precipitation.json timestampArray->GetValue(0)	1439288745	1439288745	✓



Description	Input	Expected Result	Result	Success
Check the number of components in test data set	res/test-data/precipitation.json priorityArray->GetNumberOfTuples()	1	1	✓
Check priority	res/test-data/precipitation.json priorityArray->GetValue(0)	Configuration::getInstance().getInteger("dataReader.maximumPriority")	10 (depends on configuration file)	✓



JsonReader.cpp

TestJsonReader.cpp	Reads our custom kJSON documents and loads them in our model.
TestJsonReader, WriteTemperatureToVtkPolyData	Read some test data into our model Import the test data to vtkPolyData Check if the vtkPolyData has been generated correctly
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
	res/test-data/ temperature.json temperatureArray->GetNumberOfComponents()	1	1	✓
	res/test-data/ temperature.json temperatureArray->GetValue(0)	21.0	21.0	✓
	res/test-data/ temperature.json timestampArray->GetNumberOfTuples()	1	1	✓
	res/test-data/ temperature.json timestampArray->GetValue(0)	14392887 45	14392887 45	✓
	res/test-data/ temperature.json priorityArray->GetNumberOfTuples()	1	1	✓



Description	Input	Expected Result	Result	Success
	res/test-data/ temperature.json priorityArray->GetValue(0)	Configurat ion::getIns tance().ge tInteger("dataRead er.maximu mPriority") ,	10 (depends on configurat ion file)	✓



JsonReader.cpp

TestJsonReader.cpp	Reads our custom kJSON documents and loads them in our model.
TestJsonReader, WriteWindToVtkPolyData	Read some test data into our model Import the test data to vtkPolyData Check if the vtkPolyData has been generated correctly
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
	res/test-data/wind.json speedsArray->GetNumberOfComponents()	1	1	✓
	res/test-data/wind.json speedsArray->GetValue(0)	0.4	0.4	✓
	res/test-data/wind.json directionsArray->GetNumberOfComponents()	1	1	✓
	res/test-data/wind.json directionsArray->GetValue(0)	120	120	✓
	res/test-data/wind.json timestampArray->GetNumberOfTuples()	1	1	✓
	res/test-data/wind.json timestampArray->GetValue(0)	14392887 45	14392887 45	✓
	res/test-data/wind.json priorityArray->GetNumberOfTuples()	1	1	✓



Description	Input	Expected Result	Result	Success
	res/test-data/wind.json priorityArray->GetValue(0)	Configuration::getInstance().getInteger("dataReader.maximumPriority") ,	10 (depends on configuration file)	✓



JsonReader.cpp

TestJsonReader.cpp	Reads our custom kJSON documents and loads them in our model.
TestJsonReader, WriteCloudCoverageToVtkPolyData	Read some test data into our model Import the test data to vtkPolyData Check if the vtkPolyData has been generated correctly
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
	res/test-data/cloud-coverage.json coverageArray->GetNumberOfComponents()	1	1	✓
	res/test-data/cloud-coverage.json coverageArray->GetValue(0)	0.4	0.4	✓
	res/test-data/cloud-coverage.json timestampArray->GetNumberOfTuples()	1	1	✓
	res/test-data/cloud-coverage.json timestampArray->GetValue(0)	14392887 45	14392887 45	✓
	res/test-data/cloud-coverage.json priorityArray->GetNumberOfTuples()	1	1	✓



Description	Input	Expected Result	Result	Success
	res/test-data/cloud-coverage.json priorityArray->GetValue(0)	Configuration::getInstance().getInteger("dataReader.maximumPriority") ,	10 (depends on configuration file)	✓



JsonReaderFactory.cpp

TestJsonReaderFactory.cpp	Creates a JSON reader from a JSON file.
TestJsonReaderFactory, ReadValidFiles	Read some test data into our model Check if Json reader recognized the data format correctly
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
	res/test-data/cities.json cityReader->getDataType()	Data::CITIES	Data::CITIES	✓
	res/test-data/cities.json cityReader->hasTemporalData()	false	false	✓
	res/test-data/tweets.json tweetReader->getDataType()	Data::TWEETS	Data::TWEETS	✓
	res/test-data/tweets.json tweetReader->hasTemporalData()	true	true	✓



JsonReaderFactory.cpp

TestJsonReaderFactory.cpp	Creates a JSON reader from a JSON file.
TestJsonReaderFactory, ReadInvalidFiles	Read some test data into our model Check if Json reader recognized the data format correctly, or if the correct exception is thrown
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Check that the JsonReaderFileNotFoundException is thrown	JsonReaderFactory::createReader("res/test-data/non-existent.json")	JsonReaderFileNotFoundException	JsonReaderFileNotFoundException	✓
Check that the JsonReaderParseException is thrown	JsonReaderFactory::createReader("res/test-data/invalid.json")	JsonReaderParseException	JsonReaderParseException	✓



TestRunner.cpp

TestRunner.cpp

KronosTestRunner::run()

initialize and run a QApplication which is needed for some QT functionality

All Tests Succeeded



Configuration.cpp

TestConfiguration.cpp	Includes all constants which can be modified in our configuration file.
TestConfiguration, ValidCalls	Checks that config contains valid values.
All Tests Succeeded	

Description	Input	Expected Result	Result	Success
Check that the configuration file contains a parameter named "globe.radius"	<code>Configuration::getInstance().hasKey("globe.radius")</code>	true	true	✓
Check that the configuration file doesn't recognize parameters that don't exist.	<code>Configuration::getInstance().hasKey("globe.invalid")</code>	false	false	✓
Check that the parameter "globe.radius" is 100	<code>Configuration::getInstance().getDouble("globe.radius")</code>	100	100	✓



Description	Input	Expected Result	Result	Success
Check that the parameter "globe.name" is "Foo"	<code>Configuration::getInstance().getString("globe.name").toString()</code>	"Foo"	"Foo"	✓
Check that the parameter "globe.spheric" is true	<code>Configuration::getInstance().getBoolean("globe.spheric")</code>	true	true	✓
Check that the parameter "latitude" (as double) is 12.2345	<code>Configuration::getInstance().getDouble("latitude")</code>	12.2345	12.2345	✓
Check that the parameter "latitude" (as float) is 12.2345f	<code>Configuration::getInstance().getFloat("latitude")</code>	12.2345f	12.2345f	✓
Check that the parameter "deeply.nested.configuration.group.longitude" is 5.234	<code>Configuration::getInstance().getDouble("deeply.nested.configuration.group.longitude")</code>	5.234	5.234	✓



Configuration.cpp

TestConfiguration.cpp	Includes all constants which can be modified in our configuration file.
TestConfiguration, InvalidCalls	Checks that config raises exception for wrong calls
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Check that the invalid key exception is thrown	<code>Configuration::getInstance().getString("globe.invalid")</code>	InvalidKeyException	InvalidKeyException	✓
Check that the invalid value exception is thrown	<code>Configuration::getInstance().getDouble("globe.name")</code>	InvalidValueException	InvalidValueException	✓



ResourcePool.cpp

TestResourcePool.cpp

TestResourcePool, CreateEmptyPool

All Tests Succeeded



Description	Input	Expected Result	Result	Success
Checkst that the event size is 0	DummyResource::events.size()	0	0	



ResourcePool.cpp

TestResourcePool.cpp

TestResourcePool, CreateSingleResource

All Tests Succeeded



Description	Input	Expected Result	Result	Success
Checkst that the event size is 1 after acquiring a resource pool	DummyResource::events.size()	1	1	
	DummyResource::Event(DummyResource::Event::CREATED, 0)	DummyResource::events[0]	DummyResource::events[0]	
	DummyResource::events.size()	2	2	
	DummyResource::Event(DummyResource::Event::DESTROYED, 0)	DummyResource::events[1]	DummyResource::events[1]	



ResourcePool.cpp

TestResourcePool.cpp

TestResourcePool, PoolOverflow

All Tests Succeeded



Description	Input	Expected Result	Result	Success
Sechs pools wurden akquiriert	EXPECT_EQ(6, DummyResource::events.size());	6	6	
Für i = 0...6	DummyResource::Event(DummyResource::Event::CREATED, i)	DummyResource::events[i]	DummyResource::events[i]	





ResourcePool.cpp

TestResourcePool.cpp

TestResourcePool, HandleManualRelease

All Tests Succeeded



Description	Input	Expected Result	Result	Success
Check id of 4th resource	<code>handles[4].getResource().id</code>	0	0	
Check if handle is expired	<code>handle.isExpired()</code>	true	true	



ResourcePool.cpp

TestResourcePool.cpp

TestResourcePool, HandleAutoRelease

All Tests Succeeded



Description	Input	Expected Result	Result	Success
Check id of 4th resource	handles[4].getResource().id	0	0	



ConfigUtil.cpp

TestConfigUtil.cpp	Load layer information from a file.
TestConfigUtil, LoadConfig	Check if the tile download properties are as expected.
All Tests Succeeded	✔

Description	Input	Expected Result	Result	Success
Check the size of the layers for the specified file	<code>./res/layers.json</code> <code>layers.size()</code>	2	2	✔
Check if the base url to download satellite tiles is correct	<code>satelliteImagery.getBaseUrl().toString()</code>	"http://worldwind25.arc.nasa.gov/wms?service=WMS&request=GetMap&version=1.3.0&crs=CRS:84&styles=&transparent=FALSE"	"http://worldwind25.arc.nasa.gov/wms?service=WMS&request=GetMap&version=1.3.0&crs=CRS:84&styles=&transparent=FALSE"	✔
Check if the media type for the satellite tiles is correct	<code>satelliteImagery.getMimeType().toString()</code>	image/jpeg	image/jpeg	✔
Check if the tile size is as expected	<code>satelliteImagery.getTileSize()</code>	512	512	✔



Description	Input	Expected Result	Result	Success
Check if the height map download base url is correct	<code>heightmap.getBaseUrl().toString()</code>	"http://worldwind26.arc.nasa.gov/wms?service=WMS&request=GetMap&version=1.3.0&crs=CRS:84&styles=&transparent=FALSE"	"http://worldwind26.arc.nasa.gov/wms?service=WMS&request=GetMap&version=1.3.0&crs=CRS:84&styles=&transparent=FALSE"	✓
Check if the height map's media type is as expected	<code>heightmap.getMimeType().toString()</code>	application/bil16	application/bil16	✓
Check if the height map's tile is as expected	<code>heightmap.getTileSize()</code>	512	512	✓



ConfigUtil.cpp

TestConfigUtil.cpp	Load layer information from a file.
TestConfigUtil, LayerSteps	Check if the tile download properties are as expected.
All Tests Succeeded	✓




Description	Input	Expected Result	Result	Success
Check if the layer size is as expected	ImageLayerDescription satelliteImagery = layers.value("satelliteImagery"); satelliteImagery.getLayerSteps().size()	2	2	✓
Check if the number of layers is correct	ImageLayerDescription satelliteImagery = layers.value("satelliteImagery"); satelliteImagery.getLayerSteps().at(0).minZoomLevel	0	0	✓
Check if we get the right layer information for this zoom level	satelliteImagery.getLayerSteps().at(0).layers.toString()	"BlueMarble-200405"	"BlueMarble-200405"	✓
	satelliteImagery.getLayerSteps().at(1).minZoomLevel	8	8	✓
	satelliteImagery.getLayerSteps().at(1).layers.toString()	BlueMarble-200405, esat	BlueMarble-200405, esat	✓
	ImageLayerDescription heightmap = layers.value("heightmap"); heightmap.getLayerSteps().size()	3	3	✓



Description	Input	Expected Result	Result	Success
	<pre>ImageLayerDescription heightmap = layers.value("heightmap"); heightmap.getLayerSteps().at (0).minZoomLevel</pre>	0	0	✓
	<pre>ImageLayerDescription heightmap = layers.value("heightmap"); heightmap.getLayerSteps().at (0).layers.toStdString()</pre>	std::string ("NASA_S RTM30_90 0m_Tiled")	std::string ("NASA_S RTM30_90 0m_Tiled")	✓
	<pre>ImageLayerDescription heightmap = layers.value("heightmap"); heightmap.getLayerSteps().at (2).minZoomLevel</pre>	8	8	✓
	<pre>ImageLayerDescription heightmap = layers.value("heightmap"); heightmap.getLayerSteps().at (2).layers.toStdString()</pre>	NASA_SR TM30_900 m_Tiled,a ster_v2,US GS-NED	NASA_SR TM30_900 m_Tiled,a ster_v2,US GS-NED	✓





ImageCache.cpp

TestImageCache.cpp		Cache image tiles for certain layers.		
TestImageCache, DirectorySetup		Test the cache setup.		
All Tests Succeeded				
Description	Input	Expected Result	Result	Success
Check if cache exists before setup	<code>QDir("cache").exists()</code>	false	false	
Check if cache exists after setup	<code>ImageCache::getInstance();</code> <code>QDir("cache").exists()</code>	true	true	






ImageCache.cpp

TestImageCache.cpp		Cache image tiles for certain layers.		
TestImageCache, CacheImage		Test the cache reader.		
All Tests Succeeded				
Description	Input	Expected Result	Result	Success
Check the reader's meta text	<code>reader.text("kronos-meta").toString()</code>	"1,42"	"1,42"	



ImageCache.cpp

TestImageCache.cpp	Cache image tiles for certain layers.
TestImageCache, CacheRetrieval	Check if a tile gets cached after is has been added.
All Tests Succeeded	

Description	Input	Expected Result	Result	Success
Check if the tile is not available when it hasn't been cached, yet.	<code>ImageCache::getInstance().isImageCached(QString("test-layer"), 8, 3, 7)</code>	false	false	
Now the image tile has been added. It should now be available in the cache.	<code>ImageCache::getInstance().cacheImage(MetaImage(image, 1, 42), QString("test-layer"), 8, 3, 7);</code> <code>ImageCache::getInstance().isImageCached(QString("test-layer"), 8, 3, 7)</code>	true	true	
Get Exception if not-cached image has been requested.	<code>ImageCache::getInstance().getCachedImage(QString("non-existent-layer"), 8, 3, 7)</code>	ImageNotCachedException	ImageNotCachedException	
	<code>MetaImage retrievedImage = ImageCache::getInstance().getCachedImage(QString("test-layer"), 8, 3, 7);</code> <code>retrievedImage.hasMetaData()</code>	true	true	



Description	Input	Expected Result	Result	Success
Check the minimum height of the loaded tile.	<pre>MetaImage retrievedImage = ImageCache::getInstance().ge tCachedImage(QString("test- layer"), 8, 3, 7); retrievedImage.getMinimumHei ght() retrievedImage.getMinimumHei ght()</pre>	1	1	✓
Check the maximum height of the loaded tile.	<pre>MetaImage retrievedImage = ImageCache::getInstance().ge tCachedImage(QString("test- layer"), 8, 3, 7); retrievedImage.getMaximumHei ght()</pre>	42	42	✓
Get the width of the loaded tile.	<pre>MetaImage retrievedImage = ImageCache::getInstance().ge tCachedImage(QString("test- layer"), 8, 3, 7); retrievedImage.getImage().wi dth()</pre>	512	512	✓
Get the height of the loaded tile.	<pre>MetaImage retrievedImage = ImageCache::getInstance().ge tCachedImage(QString("test- layer"), 8, 3, 7); retrievedImage.getImage().he ight()</pre>	512	512	✓



ImageCache.cpp

TestImageCache.cpp	Cache image tiles for certain layers.
TestImageCache, ClearCache	Check if the deletion of a layer in a cache works.
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
A tile has been added to the cache. It should therefore exist.	<pre>ImageCache::getInstance().cacheImage(MetaImage(image, 1, 42), QString("layer-to-clear"), 2, 1, 3);</pre> <pre>QDir("cache/layer-to-clear").exists()</pre>	true	true	✓
Now the tile is deleted from the cache. Now it shouldn't be available anymore.	<pre>ImageCache::getInstance().clearCache("layer-to-clear");</pre> <pre>QDir("cache/layer-to-clear").exists()</pre>	false	false	✓
The cache should still exist, tough.	<pre>QDir("cache").exists()</pre>	true	true	✓



ImageCache.cpp

TestImageCache.cpp		Cache image tiles for certain layers.		
TestImageCache, DeleteCachedImage		Check if the deletion of tiles in a cache works.		
All Tests Succeeded				
Description	Input	Expected Result	Result	Success
Cache image and check if the added tile exists in cache then.	<pre>ImageCache::getInstance().cacheImage(MetaImage(image, 1, 42), QString("layer"), 2, 1, 3);</pre> <pre>ImageCache::getInstance().isImageCached(QString("layer"), 2, 1, 3)</pre>	true	true	
Check for a tile which does not exist in the cache. It should throw an exception then.	<pre>ImageCache::getInstance().deleteCachedImage(QString("layer"), 42, 1, 3)</pre>	ImageNotCachedException	ImageNotCachedException	
A specific tile is deleted and therefore not available in the cache anymore.	<pre>ImageCache::getInstance().deleteCachedImage(QString("layer"), 2, 1, 3)</pre> <pre>ImageCache::getInstance().isImageCached(QString("layer"), 2, 1, 3)</pre>	false	false	
The cache folder should still exist though.	<pre>QDir("cache").exists()</pre>	true	true	



ImageDownloadWorker.cpp

TestImageDownloadWorker.cpp	ImageDownloadWorkers are responsible for downloading the given image, using a provided QNetworkAccessManager.
TestImageDownloadWorker, Download	Checks if an image download is successful.
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Download an image from the given URL and check if the downloaded image has the expected width.	<pre> QUrl imgUrl("http:// worldwind25.arc.nasa.gov/ wms? service=WMS&request=GetMap&v ersion=1.3.0&crs=CRS: 84&layers=esat&styles=&trans parent=FALSE&format=image/ jpeg&width=512&height=512&bb ox=-90,-180,-89.9973,-179.99 7"); img.getImage().width() </pre>	512	512	✓
Download an image from the given URL and check if the downloaded image has the expected height.	<pre> QUrl imgUrl("http:// worldwind25.arc.nasa.gov/ wms? service=WMS&request=GetMap&v ersion=1.3.0&crs=CRS: 84&layers=esat&styles=&trans parent=FALSE&format=image/ jpeg&width=512&height=512&bb ox=-90,-180,-89.9973,-179.99 7"); img.getImage().height() </pre>	512	512	✓



ImageDownloadWorker.cpp

TestImageDownloadWorker.cpp	ImageDownloadWorkers are responsible for downloading the given image, using a provided QNetworkAccessManager.
TestImageDownloadWorker, AbortDownload	Checks if a worker stops downloading an image when the download has been aborted.
All Tests Succeeded	✔

Description	Input	Expected Result	Result	Success
Download an image from the given URL and wait for the worker to start downloading the image. Then the download is aborted. Now the image should not be downloaded.	<pre> QUrl imgUrl("http:// worldwind25.arc.nasa.gov/ wms? service=WMS&request=GetMap&v ersion=1.3.0&crs=CRS: 84&layers=esat&styles=&trans parent=FALSE&format=image/ jpeg&width=512&height=512&bb ox=-90,-180,-89.9973,-179.99 7"); while (!worker.isRunning()); worker.abortDownload(); worker.getFuture().getImage() </pre>	Download AbortedException	Download AbortedException	✔



ImageDownloadWorker.cpp

TestImageDownloadWorker.cpp	ImageDownloadWorkers are responsible for downloading the given image, using a provided QNetworkAccessManager.
TestImageDownloadWorker, AbortDownloadBeforeStart	Checks if a worker stops downloading an image when the download has been aborted before the download actually has started.
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
The image URL is specified, and the worker will start downloading soon, but the download is already aborted. The image shouldn't be downloaded then.	<pre> QUrl imageUrl("http:// worldwind25.arc.nasa.gov/ wms? service=WMS&request=GetMap&v ersion=1.3.0&crs=CRS: 84&layers=esat&styles=&trans parent=FALSE&format=image/ jpeg&width=512&height=512&bb ox=-90,-180,-89.9973,-179.99 7"); worker.abortDownload(); worker.getFuture().getImage() </pre>	Download AbortedE xception	Download AbortedE xception	✓



ImageDownloader.cpp

TestImageDownloader.cpp

ImageDownload is responsible for managing and downloading images.

TestImageDownloader, GetAvailableLayers

Check the available layers.

All Tests Succeeded



Description	Input	Expected Result	Result	Success
The available layers should contain satellite imagery.	<code>availableLayers.contains("satelliteImagery")</code>	true	true	
The available layers should contain a height map.	<code>availableLayers.contains("heightmap")</code>	true	true	



ImageDownloader.cpp

TestImageDownloader.cpp	ImageDownload is responsible for managing and downloading images.
TestImageDownloader, GetTile	Check the available tiles.
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Fetch specific tile. There shouldn't be an exception.	<code>ASSERT_NO_THROW(downloader.fetchTile(10, 20, 30))</code>	true	true	✓
Get the downloaded tile.	<code>ASSERT_NO_THROW(tile = future.get());</code>	true	true	✓
The zoom level of the tile needs to be equal to the requested zoomlevel.	<code>tile.getZoomLevel()</code>	10	10	
The x coordinate of the tile needs to be equal to the requested tile coordinate.	<code>tile.getTileX()</code>	20	20	



Description	Input	Expected Result	Result	Success
The y coordinate of the tile needs to be equal to the requested tile coordinate.	<code>tile.getTileY()</code>	30	30	
Number of layers must be 2 (heightmap and satellite)	<code>tile.getLayers().size()</code>	2	2	
The image width is expected to be 512.	<code>image.width()</code>	512	512	
The image height is expected to be 512.	<code>image.height()</code>	512	5122	



ImageDownloader.cpp

TestImageDownloader.cpp	ImageDownload is responsible for managing and downloading images.
TestImageDownloader, AbortDownload	Download a tile and then abort the download.
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Download a tile and abort the tile download right after.	<pre> downloader.fetchTile(zoomLevel, tileX, tileY); downloader.abortAllDownloads(); future.get() </pre>	Download AbortedException	Download AbortedException	✓



ImageLayerDescription.cpp

TestImageLayerDescription.cpp	This class holds information containing a whole image layer and how it should be accessed using the API, all specified in the configuration JSON file. On top of that this class includes some static helper methods for API access.
TestImageLayerDescription, GetTilePositionFromCoordinates	Convert lat / lon coordinates to tile coordinates and check if they are correct.
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Insert lat, lon and zoomlevel to get tileX.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 0; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	0	0	✓
Insert lat, lon and zoomlevel to get tileY.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 0; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	0	0	✓



Description	Input	Expected Result	Result	Success
Insert lat, lon and zoomlevel to get tileX.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 1; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	1	1	✓
Insert lat, lon and zoomlevel to get tileY.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 1; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	0	0	✓
Insert lat, lon and zoomlevel to get tileX.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 2; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	3	3	✓



Description	Input	Expected Result	Result	Success
Insert lat, lon and zoomlevel to get tileY.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 2; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	0	0	✓
Insert lat, lon and zoomlevel to get tileX.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 3; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	7	7	✓
Insert lat, lon and zoomlevel to get tileY.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 3; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	1	1	✓



Description	Input	Expected Result	Result	Success
Insert lat, lon and zoomlevel to get tileX.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 4; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	15	15	✓
Insert lat, lon and zoomlevel to get tileY.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 4; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	3	3	✓
Insert lat, lon and zoomlevel to get tileX.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 5; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	31	31	✓



Description	Input	Expected Result	Result	Success
Insert lat, lon and zoomlevel to get tileY.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 5; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	6	6	✓
Insert lat, lon and zoomlevel to get tileX.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 6; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	63	63	✓
Insert lat, lon and zoomlevel to get tileY.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 6; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	13	13	✓



Description	Input	Expected Result	Result	Success
Insert lat, lon and zoomlevel to get tileX.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 7; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	127	127	✓
Insert lat, lon and zoomlevel to get tileY.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 7; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	27	27	✓
Insert lat, lon and zoomlevel to get tileX.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 8; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	255	255	✓



Description	Input	Expected Result	Result	Success
Insert lat, lon and zoomlevel to get tileY.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 8; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	54	54	✓
Insert lat, lon and zoomlevel to get tileX.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 9; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	511	511	✓
Insert lat, lon and zoomlevel to get tileY.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 9; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	109	109	✓



Description	Input	Expected Result	Result	Success
Insert lat, lon and zoomlevel to get tileX.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 10; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	1023	1023	✓
Insert lat, lon and zoomlevel to get tileY.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 10; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	219	219	✓
Insert lat, lon and zoomlevel to get tileX.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 11; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	2046	2046	✓



Description	Input	Expected Result	Result	Success
Insert lat, lon and zoomlevel to get tileY.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 11; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	438	438	✓
Insert lat, lon and zoomlevel to get tileX.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 12; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	4092	4092	✓
Insert lat, lon and zoomlevel to get tileY.	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 12; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	876	876	✓



Description	Input	Expected Result	Result	Success
	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 13; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	8185	8185	✓
	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 13; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	1752	1752	✓
	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 14; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	16371	16371	✓



Description	Input	Expected Result	Result	Success
	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 14; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	35904	3504	✓
	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 15; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileX</pre>	32742	32742	✓
	<pre>double latitude = 51.501414; double longitude = -0.141554; int zoomLevel = 15; ImageLayerDescription::getTilePositionFromCoordinates(latitude, longitude, zoomLevel, tileX, tileY); tileY</pre>	7008	7008	✓



ImageTile.cpp

TestImageTile.cpp	Contains images for each layer that may be present at this zoom and tile position.
TestImageTile, MetaData	Create tile and check its properties.
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Check zoom level of tile.	ImageTile tile(layers, 10, 2, 3); tile.getZoomLevel()	10	10	✓
Check x coordinate of tile.	ImageTile tile(layers, 10, 2, 3); tile.getTileX()	2	2	
Check y coordinate of tile.	ImageTile tile(layers, 10, 2, 3); tile.getTileY()	3	3	



ImageTile.cpp

TestImageTile.cpp	Contains images for each layer that may be present at this zoom and tile position.
TestImageTile, Layers	Create tile and check its layers.
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Check layer size	<code>layers.insert("layerOne", metaImageOne);</code> <code>layers.insert("layerTwo", metaImageTwo);</code> <code>layerTiles.size()</code>	2	2	✓
Check if inserted layer still exists	<code>layers.insert("layerOne", metaImageOne);</code> <code>layers.insert("layerTwo", metaImageTwo);</code> <code>layerTiles.contains("layerOne")</code>	true	true	✓
Check if inserted layer still exists	<code>layers.insert("layerOne", metaImageOne);</code> <code>layers.insert("layerTwo", metaImageTwo);</code> <code>layerTiles.contains("layerTwo")</code>	true	true	✓
Check layer image width	<code>layers.insert("layerOne", metaImageOne);</code> <code>layers.insert("layerTwo", metaImageTwo);</code> <code>layerOne.getImage().width()</code>	512	512	✓



Description	Input	Expected Result	Result	Success
Check layer image height	<code>layers.insert("layerOne", metaImageOne);</code> <code>layers.insert("layerTwo", metaImageTwo);</code> <code>layerOne.getImage().height()</code>	512	512	✓
Check layer image width	<code>layers.insert("layerOne", metaImageOne);</code> <code>layers.insert("layerTwo", metaImageTwo);</code> <code>layerTwo.getImage().width()</code>	512	512	✓
Check layer image height	<code>layers.insert("layerOne", metaImageOne);</code> <code>layers.insert("layerTwo", metaImageTwo);</code> <code>layerTwo.getImage().height()</code>	512	512	✓



MetalmImage.cpp

TestMetalmImage.cpp	An object of the type MetalmImage wraps the actual image data a layer in a specific tile offers as a QImage with potential meta data describing that image, such as for example the minimum and maximum height of a heightmap
TestMetaData, MetaData	Create meta data and check its properties.
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Check zoom level of tile.	MetaImage metaImageOne(image, 10, 42); metaImageOne.hasMetaData()	true	true	✓
Check x coordinate of tile.	MetaImage metaImageOne(image, 10, 42); metaImageOne.getMinimumHeight()	10	10	✓
Check y coordinate of tile.	MetaImage metaImageOne(image, 10, 42); metaImageOne.getMaximumHeight()	42	42	✓
Create a metaimage without meta data	MetaImage metaImageTwo(image); metaImageTwo.hasMetaData()	false	false	✓
Now add metadata, now it has metadata	MetaImage metaImageTwo(image); metaImageTwo.setMetaData(11, 43); metaImageTwo.hasMetaData()	true	true	✓



Description	Input	Expected Result	Result	Success
Check minimum height.	<pre> MetaImage metaImageTwo(image); metaImageTwo.setMetaData(11, 43); metaImageTwo.getMinimumHeight() </pre>	11	11	✓
Check maximum height.	<pre> MetaImage metaImageTwo(image); metaImageTwo.setMetaData(11, 43); metaImageTwo.getMinimumHeight() </pre>	43	43	✓



MetalmImage.cpp

TestMetalmImage.cpp	An object of the type MetalmImage wraps the actual image data a layer in a specific tile offers as a QImage with potential meta data describing that image, such as for example the minimum and maximum height of a heightmap
TestMetalmImage, Image	Check width and height of image.
All Tests Succeeded	✓

Description	Input	Expected Result	Result	Success
Check width of created MetalmImage image.	<pre>QImage image(512, 512, QImage::Format_RGB32); MetalmImage metaImage(image, 1, 2); metaImage.getImage().width()</pre>	512	512	✓
Check height of created MetalmImage image.	<pre>QImage image(512, 512, QImage::Format_RGB32); MetalmImage metaImage(image, 1, 2); metaImage.getImage().height()</pre>	512	512	✓

