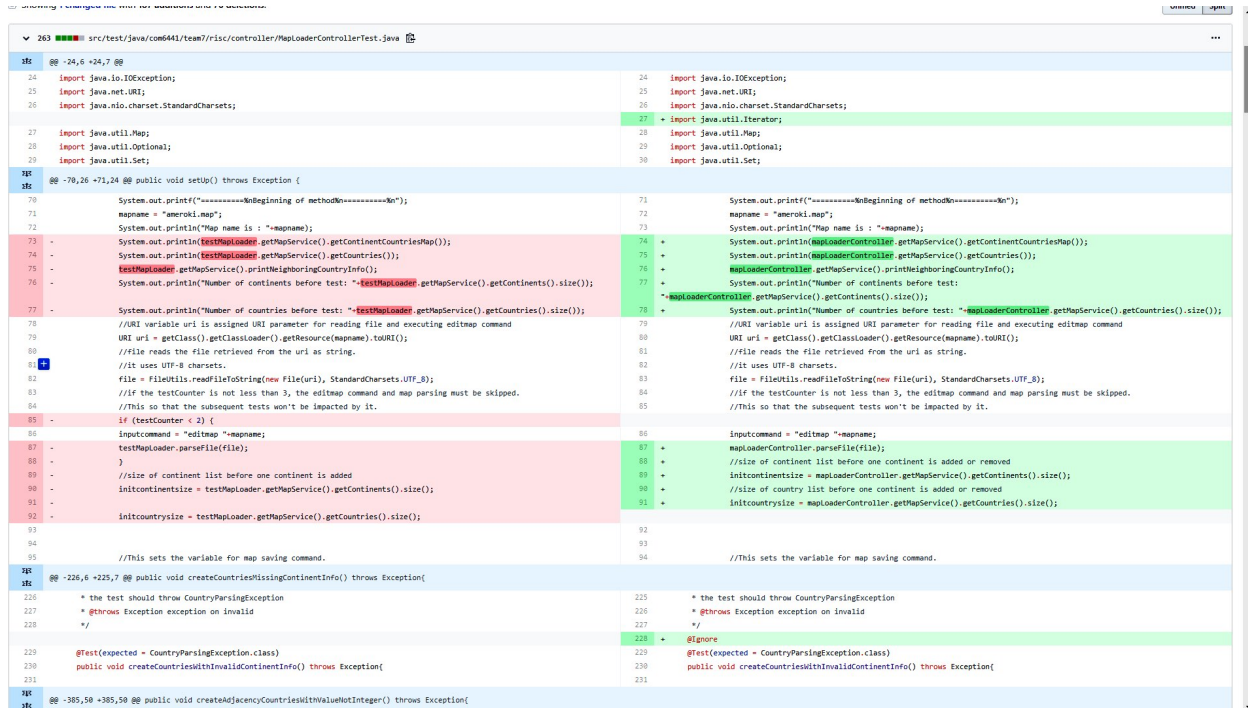
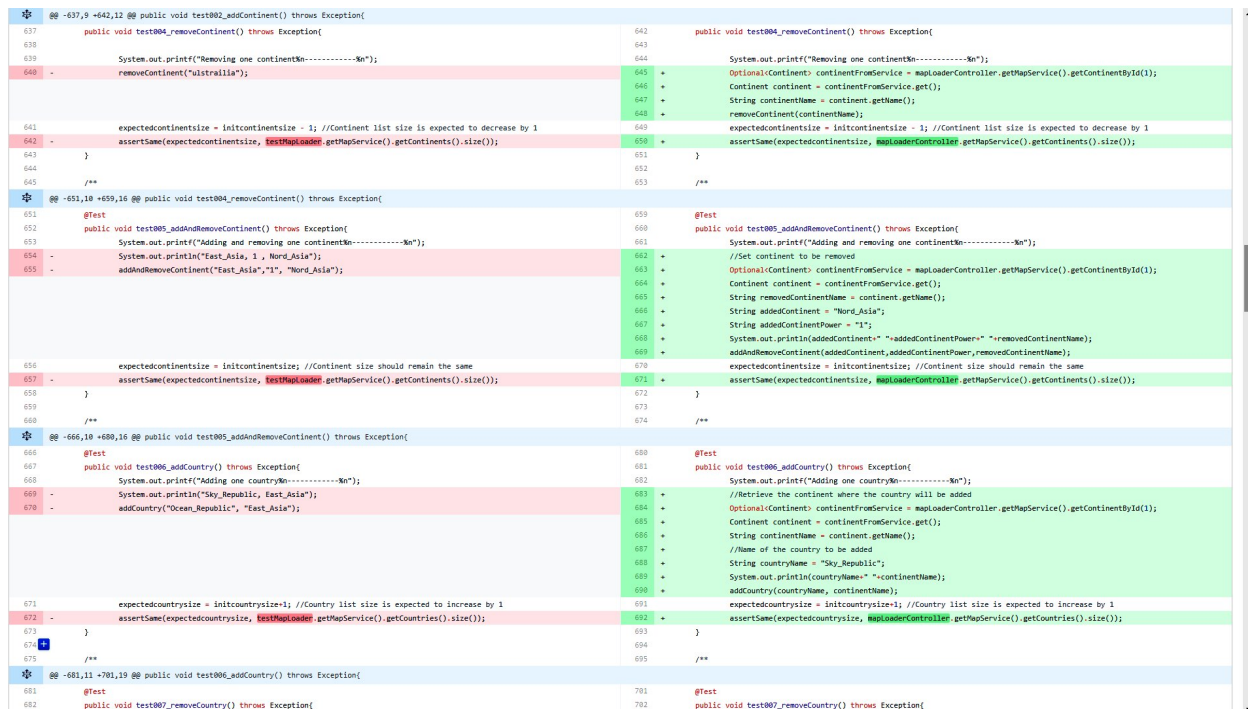


Refactoring List for build 3 – Team 7

1. We have revised the map test to make each test case independent from each other.



Using iterator type to dynamically traverse through list of countries in map file.



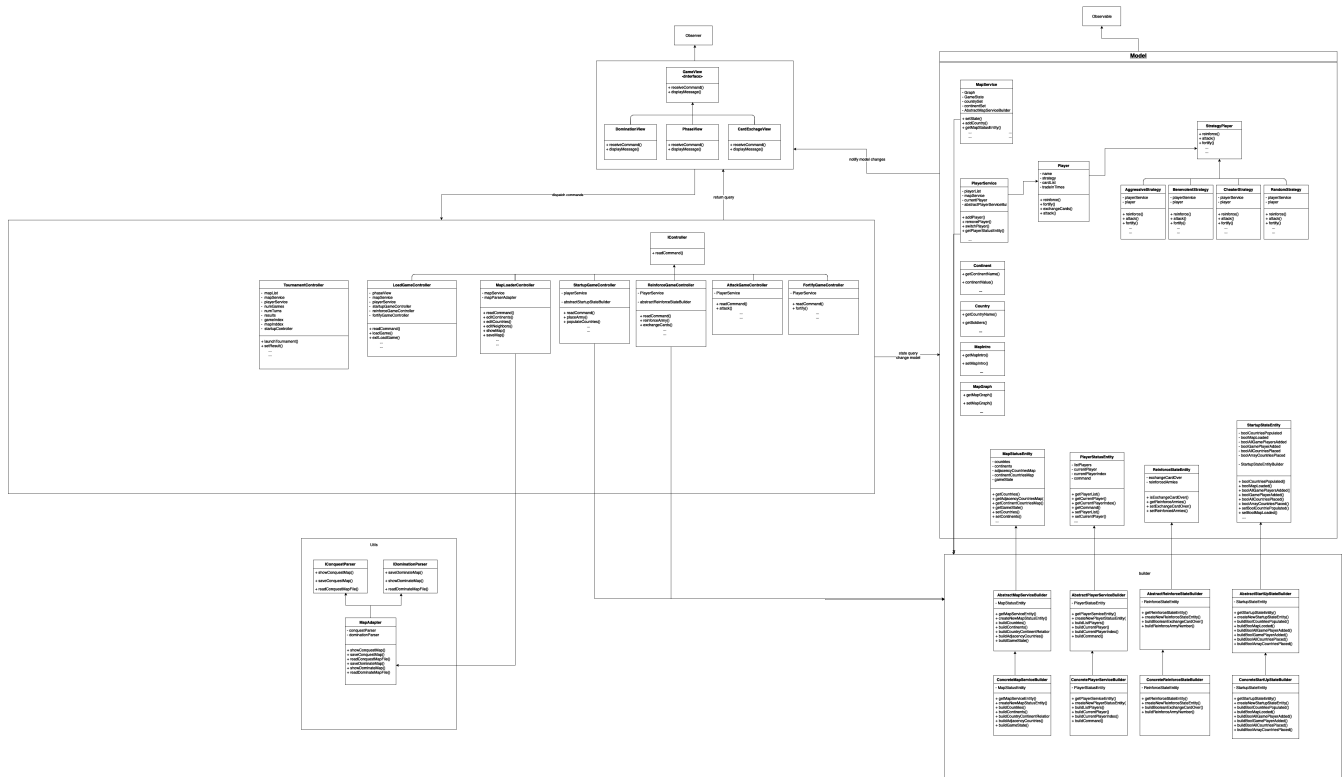
Removed all the hardcoding specific to a map file.

681	@Test	781	@Test
682	public void test007_removeCountry() throws Exception{	782	public void test007_removeCountry() throws Exception{
683	System.out.printf("Removing one country\n-----\n");	783	System.out.printf("Removing one country\n-----\n");
684	//size of country list before one country is removed	784	//Get set of countries using service
685	System.out.println("heal");	785	Set<Country> countriesFromService = mapLoaderController.getMapService().getCountries();
		786	//Get the first country in the set using iterator
		787	Iterator<Country> countryIterator = countriesFromService.iterator();
		788	//Set the country
		789	Country country = countryIterator.next();
		790	//Retrieve the country name
		791	String countryName = country.getCountryName();
		792	//Call the removeCountry method with countryName as the param
		793	removeCountry(countryName);
		794	System.out.println(countryName);
686	expectedCountrySize = initCountrySize-1; //Country list size is expected to decrease by 1	795	expectedCountrySize = initCountrySize-1; //Country list size is expected to decrease by 1
687	removeCountry("heal");	796	assertSame(expectedCountrySize, mapLoaderController.getMapService().getCountries().size());
688	assertSame(expectedCountrySize, testMapLoader.getMapService().getCountries().size());		
689	}	797	}
690		798	
691	/**	799	/**
✱	@ -697,9 +725,27 @@ public void test007_removeCountry() throws Exception{	725	@Test
697	@Test	726	public void test008_addAndRemoveCountry() throws Exception{
698	public void test008_addAndRemoveCountry() throws Exception{	727	System.out.printf("Adding and removing one country\n-----\n");
699	System.out.printf("Adding and removing one country\n-----\n");	728	//Retrieve the continent where the country will be added
700	addAndRemoveCountry("Sky_Republic", "East_Asia", "Ocean_Republic");	729	Optional<Continent> continentFromService = mapLoaderController.getMapService().getContinentById(1);
		730	Continent continent = continentFromService.get();
		731	String continentName = continent.getName();
		732	//Name of the country to be added
		733	String countryName = "Sky_Republic";
		734	System.out.println(countryName+" "+continentName);
		735	//Get set of countries using service
		736	Set<Country> countriesFromService = mapLoaderController.getMapService().getCountries();
		737	//Get the first country in the set using iterator
		738	Iterator<Country> countryIterator = countriesFromService.iterator();
		739	//Set the country to be removed
		740	Country country = countryIterator.next();
		741	//Retrieve the country name
		742	String removedCountryName = country.getCountryName();
		743	//Call the removeCountry method with countryName as the param
		744	removeCountry(removedCountryName);
		745	System.out.println("Remove "+removedCountryName);
		746	addAndRemoveCountry(countryName, continentName, removedCountryName);
701	expectedCountrySize = initCountrySize; //Country list size should remain the same	747	expectedCountrySize = initCountrySize; //Country list size should remain the same
702	assertSame(expectedCountrySize, testMapLoader.getMapService().getCountries().size());	748	assertSame(expectedCountrySize, mapLoaderController.getMapService().getCountries().size());
703	}	749	}
704		750	
705	/**	751	/**
✱	@ -713,8 +759,25 @@ public void test008_addAndRemoveCountry() throws Exception{	759	@Test
713	@Test		

Using set and iterator types in conjunction to completely eliminate to remove any hard coded continents and countries, instead getting them from map file.

715	System.out.printf("Adding one neighbor to a country\n-----\n");	761	System.out.printf("Adding one neighbor to a country\n-----\n");
716	System.out.println("south_africa", "india");	762	//Add a country first
717	addNeighbor("Sky_Republic", "india");	763	//Retrieve the continent where the country will be added
		764	Optional<Continent> continentFromService = mapLoaderController.getMapService().getContinentById(1);
		765	Continent continent = continentFromService.get();
		766	String continentName = continent.getName();
		767	//Name of the country to be added
		768	String countryName = "Sky_Republic";
		769	System.out.println(countryName+" "+continentName);
		770	addCountry(countryName, continentName);
		771	//Retrieve the origin country
		772	Set<Country> countriesFromService = mapLoaderController.getMapService().getCountries();
		773	//Get the first country in the set using iterator
		774	Iterator<Country> countryIterator = countriesFromService.iterator();
		775	//Set the country
		776	Country country = countryIterator.next();
		777	//Retrieve the country name
		778	String originCountryName = country.getCountryName();
		779	System.out.println(originCountryName+" "+countryName);
		780	addNeighbor(originCountryName, countryName);
718	//Check if map object contains both country ID and neighbor ID	781	//Check if map object contains both country ID and neighbor ID
719	assertTrue("Country is not found", borders1.containsKey(country1.getId()));	782	assertTrue("Country is not found", borders1.containsKey(country1.getId()));
720	assertTrue("Neighbouring is not found", pair1.containsKey(neighbor1.getId()));	783	assertTrue("Neighbouring is not found", pair1.containsKey(neighbor1.getId()));
✱	@ -731,8 +794,28 @@ public void test009_addNeighbor() throws Exception{		
731	@Test	794	@Test
732	public void test010_removeNeighbor() throws Exception{	795	public void test010_removeNeighbor() throws Exception{
733	System.out.printf("Removing one neighbor from a country\n-----\n");	796	System.out.printf("Removing one neighbor from a country\n-----\n");
		797	//Get first country from the country list
		798	//Get set of countries using service
		799	Set<Country> countriesFromService = mapLoaderController.getMapService().getCountries();
		800	//Get the first country in the set using iterator
		801	Iterator<Country> countryIterator = countriesFromService.iterator();
		802	//Set the country
		803	Country originCountry = countryIterator.next();
		804	//Retrieve the country name
		805	String originCountryName = originCountry.getCountryName();
		806	//Retrieve the origin country's adjacency list
		807	Set<Integer> originCountryAdjacencyList = mapService.getAdjacencyCountries(originCountry.getId());
		808	//Get the first country in adjacency list
		809	//Retrieve an adjacent country
		810	Country neighborCountry = null;
		811	for(Integer i: originCountryAdjacencyList) {
		812	neighborCountry=mapService.getCountryById(i).get();
		813	break;
		814	}
		815	//Set adjacent country's name
		816	String neighborCountryName = neighborCountry.getCountryName();
734	//get pair of country and neighbor	817	//get pair of country and neighbor
735	removeNeighbor("Siberia", "newrick");	818	removeNeighbor(originCountryName, neighborCountryName);
736	//Check if map object contains both country ID and neighbor ID	819	//Check if map object contains both country ID and neighbor ID
737	assertTrue("Country is not found", borders1.containsKey(country1.getId()));	820	assertTrue("Country is not found", borders1.containsKey(country1.getId()));

Architecture Design of Risk Project



Architecture for Build 3 making use of different patterns