

Merge Document ES Project

DiogoOliNunes, 62333, Diogo Nunes

Antonio-Almeida-62335/Antonio-06 , 62335, António Almeida

CNR13 , 62438, Diogo Mendes

jpm-a-fernandes, 62756, João Fernandes

Basthy28, 63347, Sebastião Martins

t0mast0a, 63350, Tomás Mendes



Índice:

1. Code Metrics – Pagina 3

- 1.1. 62333's Code Metrics – Pag 3
- 1.2. 62335's Code Metrics – Pag 6
- 1.3. 62438's Code Metrics – Pag 9
- 1.4. 62756's Code Metrics – Pag 10
- 1.5. 63347's Code Metrics – Pag 12
- 1.6. 63350's Code Metrics – Pag 14

2. Code Smells – Pagina 15

- 2.1. 62333's Code Smells – Pag 15
- 2.2. 62335's Code Smells – Pag 21
- 2.3. 62438's Code Smells – Pag 24
- 2.4. 62756's Code Smells – Pag 27
- 2.5. 63347's Code Smells – Pag 29
- 2.6. 63350's Code Smells – Pag 32

3. Design Patterns – Pagina 35

- 3.1. 62333's Design Patterns – Pag 35
- 3.2. 62335's Design Patterns – Pag 37
- 3.3. 62438's Design Patterns – Pag 40
- 3.4. 62756's Design Patterns – Pag 43
- 3.5. 63347's Design Patterns – Pag 46
- 3.6. 63350's Design Patterns – Pag 49

4. Use Case Diagrams – Pagina 52

- 4.1. 62333's Use Case Diagrams – Pag 52
- 4.2. 62335's Use Case Diagrams – Pag 53
- 4.3. 62438's Use Case Diagrams – Pag 54
- 4.4. 62756's Use Case Diagrams – Pag 55
- 4.5. 63347's Use Case Diagrams – Pag 56
- 4.6. 63350's Use Case Diagrams – Pag 57

5. User Stories – Pagina 58

6. Use Case Diagram of User Stories – Pagina 59

- 6.1. User Story: Tesouros Perdidos – Pag 59
- 6.2. User Story: Cidades Arruinadas – Pag 60
- 6.3. User Story: Terrenos Abençoados – Pag 61

7. Video – Pagina 61

8. Postmortem & Conclusion – Pagina 62

Code Metrics

Lines of Code (62333):

module ^	JLOC	L(CSS)	L(Groovy)	L(HTML)	L(J)	L(JS)	L(JSP)	L(KT)	L(XML)	LOC	LOCp	LOCt	NCLOC	NCLOCp	NCLOCt
SE2324_62333_62335_62438_62756_63347_63	76 637	933	0	19 605	212 127	0	0	0	14 077	428 589	212 180	0	334 112	128 099	0
test	2 230	0	0	0	19 721	0	0	0	0	103 980	0	19 721	100 611	0	16 442
Total	78 867	933	0	19 605	231 848	0	0	0	14 077	532 569	212 180	19 721	434 723	128 099	16 442
Average	39 433.50	466.50	0.00	9 802.50	115 924.00	0.00	0.00	0.00	7 038.50	268 284.50	106 090.00	9 860.50	217 361.50	64 049.50	8 221.00

package ^	CLOC	CLOC(rec)	JLOC	JLOC(rec)	LOC	LOC(rec)	LOCp	LOCp(rec)	LOCt	LOCt(rec)	NCLOC	NCLOCp	NCLOCt	NCLOCp(rec)	NCLOCt
net.sf.freecol	0	9 817	78 867	231 848	53	231 901	53	212 180	0	19 721	53	204 735	0	18 672	0
net.sf.freecol.client	9 817	0	78 867	231 848	231 848	231 848	231 848	212 127	19 721	19 721	204 682	18 672	18 672	204 682	18 672
net.sf.freecol.client.animation	0	9 817	461	78 867	1 605	231 848	1 561	212 127	45	19 721	1 522	1 480	204 682	42	18 672
net.sf.freecol.client.dialog	0	2 372	751	21 114	1 986	68 439	1 897	68 371	89	268	1 858	1 774	66 260	84	259
net.sf.freecol.client.control	0	433	1 973	1 973	6 659	6 659	6 562	6 562	97	97	6 311	6 216	95	95	95
net.sf.freecol.client.gui	0	1 798	4 480	10 302	11 593	59 819	11 511	59 737	82	82	11 287	11 207	58 096	80	80
net.sf.freecol.client.gui.action	0	133	2 779	2 779	4 914	4 914	4 914	4 914	0	0	4 781	4 781	4 781	0	0
net.sf.freecol.client.gui.animation	0	11	212	212	494	494	494	494	0	0	487	487	487	0	0
net.sf.freecol.client.gui.images	0	171	2 178	2 178	7 442	7 442	7 442	7 442	0	0	7 290	7 290	7 290	0	0
net.sf.freecol.client.gui.label	0	31	527	527	1 271	1 271	1 271	1 271	0	0	1 245	1 245	1 245	0	0
net.sf.freecol.client.gui.mapviewer	0	173	1 236	1 236	4 444	4 444	4 444	4 444	0	0	4 288	4 288	4 288	0	0
net.sf.freecol.client.gui.menu	0	62	258	258	1 237	1 237	1 237	1 237	0	0	1 177	1 177	1 177	0	0
net.sf.freecol.client.gui.option	0	40	860	860	2 281	2 281	2 281	2 281	0	0	2 247	2 247	2 247	0	0
net.sf.freecol.client.gui.panel	0	691	4 338	5 779	15 658	22 822	15 658	22 822	0	0	15 251	15 251	22 197	0	0
net.sf.freecol.client.gui.panel.colored	0	31	553	553	2 134	2 134	2 134	2 134	0	0	2 103	2 103	2 103	0	0
net.sf.freecol.client.gui.panel.report	0	203	888	888	5 030	5 030	5 030	5 030	0	0	4 843	4 843	4 843	0	0
net.sf.freecol.client.gui.plaf	0	21	747	747	2 141	2 141	2 141	2 141	0	0	2 121	2 121	2 121	0	0
net.sf.freecol.client.gui.tooltip	0	8	65	65	334	334	334	334	0	0	327	327	327	0	0
net.sf.freecol.client.gui.video	0	48	64	64	194	194	194	194	0	0	146	146	146	0	0
net.sf.freecol.client.networking	0	1	88	88	175	175	175	175	0	0	174	174	174	0	0
net.sf.freecol.common	0	3 971	142	42 830	208	109 651	173	98 456	35	11 195	205	173	95 484	32	10 670
net.sf.freecol.common.debug	0	34	315	515	1 587	1 587	1 587	1 587	0	0	1 571	1 571	1 571	0	0
net.sf.freecol.common.I18n	0	85	966	966	3 271	3 271	2 262	2 262	1 009	1 009	3 191	2 212	2 212	979	979
net.sf.freecol.common.io	0	111	1 520	1 679	3 682	4 010	3 608	3 936	74	74	3 587	3 516	3 841	71	71
net.sf.freecol.common.io.sza	0	3	159	159	328	328	328	328	0	0	325	325	325	0	0
net.sf.freecol.common.logging	0	4	90	90	228	228	228	228	0	0	224	224	224	0	0
net.sf.freecol.common.metaserver	0	12	218	218	529	529	529	529	0	0	518	518	518	0	0
net.sf.freecol.common.model	0	3 160	24 893	25 981	68 034	70 566	58 306	60 838	9 728	9 728	65 306	56 060	58 528	9 246	9 246
net.sf.freecol.common.model.mission	0	10	460	460	810	810	810	810	0	0	800	800	800	0	0
net.sf.freecol.common.model.pathfind	0	14	434	434	1 089	1 089	1 089	1 089	0	0	1 081	1 081	1 081	0	0

class ^	CLOC	JLOC	LOC
net.sf.freecol.AllTests	6	3	22
net.sf.freecol.client.AllTests	2	0	10
net.sf.freecol.client.ClientOptions	314	254	822
net.sf.freecol.client.ClientTestHelper	3	0	30
net.sf.freecol.client.control.AllTests	2	0	9
net.sf.freecol.client.control.ClientInputHandler	15	14	31
net.sf.freecol.client.control.ConnectController	172	109	483
net.sf.freecol.client.control.FreeColClientHolder	75	75	122
net.sf.freecol.client.control.InGameController	1 638	1 297	4 806
net.sf.freecol.client.control.InGameController.MoveMode	4	4	15
net.sf.freecol.client.control.MapEditorController	86	84	263
net.sf.freecol.client.control.MapTransform	29	29	39
net.sf.freecol.client.control.MoveTest	0	0	35
net.sf.freecol.client.control.PreGameController	139	130	301
net.sf.freecol.client.control.SoundController	53	39	109
net.sf.freecol.client.FreeColClient	418	343	874
net.sf.freecol.client.gui.action.AboutAction	12	11	22
net.sf.freecol.client.gui.action.ActionManager	61	59	219
net.sf.freecol.client.gui.action.AssignTradeRouteAction	16	14	35
net.sf.freecol.client.gui.action.AttackRangedAction	13	11	35
net.sf.freecol.client.gui.action.BuildColonyAction	17	14	37
net.sf.freecol.client.gui.action.CenterAction	12	11	27
net.sf.freecol.client.gui.action.ChangeAction	24	22	75
net.sf.freecol.client.gui.action.ChangeWindowedModeAction	12	11	24
net.sf.freecol.client.gui.action.ClearOrdersAction	12	11	22
net.sf.freecol.client.gui.action.ClopediaAction	13	12	35
net.sf.freecol.client.gui.action.ClopediaAction.PanelType	1	0	9
net.sf.freecol.client.gui.action.ContinueAction	13	11	34
net.sf.freecol.client.gui.action.DebugAction	16	14	32
net.sf.freecol.client.gui.action.DeclareIndependenceAction	16	14	32
net.sf.freecol.client.gui.action.DetermineHighSeasAction	16	14	36
net.sf.freecol.client.gui.action.DisbandUnitAction	12	11	23
net.sf.freecol.client.gui.action.DisplayBordersAction	12	11	25
net.sf.freecol.client.gui.action.DisplayFogOfWarAction	16	11	38
net.sf.freecol.client.gui.action.DisplayGridAction	12	11	25

interface ^	CLOC	JLOC	LOC	NCLOC
net.sf.freecol.client.gui.animations.Animations.Procedure	4	4	7	3
net.sf.freecol.client.gui.dialog.DialogHandler	4	4	7	4
net.sf.freecol.client.gui.label.CargoLabel	18	18	24	6
net.sf.freecol.client.gui.label.Draggable	5	5	8	3
net.sf.freecol.client.gui.mapview.MapViewer.TileRenderingCallback	12	12	15	3
net.sf.freecol.client.gui.panel.DropTarget	39	38	53	15
net.sf.freecol.client.gui.panel.MiniMap.TileRenderingCallback	12	12	15	3
net.sf.freecol.client.gui.panel.report.LabourData.LocationData.G	0	0	3	3
net.sf.freecol.client.gui.video.VideoListener	6	6	9	3
net.sf.freecol.common.I18n.Selector	9	9	12	3
net.sf.freecol.common.io.sza.AnimationEvent	3	3	5	2
net.sf.freecol.common.io.sza.ImageAnimationEvent	13	13	17	4
net.sf.freecol.common.model.BaseProduction	25	25	39	14
net.sf.freecol.common.model.Constants	7	47	12	5
net.sf.freecol.common.model.Consumer	44	44	55	11
net.sf.freecol.common.model.FreeColGameObjectListener	22	22	28	6
net.sf.freecol.common.model.Locationable	33	33	40	7
net.sf.freecol.common.model.Location	133	130	166	33
net.sf.freecol.common.model.SearchHeuristic	3	3	6	3
net.sf.freecol.common.model.mission.Mission	34	47	40	6
net.sf.freecol.common.model.Movable	16	16	20	4
net.sf.freecol.common.model.Nameable	13	13	17	4
net.sf.freecol.common.model.Named	11	11	14	3
net.sf.freecol.common.model.Ownable	16	16	20	4
net.sf.freecol.common.model.Pathfinding.CostDecider	29	29	36	7
net.sf.freecol.common.model.Pathfinding.GoalDecider	44	44	49	5
net.sf.freecol.common.model.Specification.ChildReader	1	0	4	3
net.sf.freecol.common.model.TradeLocation	45	45	57	12
net.sf.freecol.common.model.TransactionListener	18	18	22	4
net.sf.freecol.common.model.MessageHandler	20	20	26	6
net.sf.freecol.common.ObjectWithId	9	9	12	3
net.sf.freecol.common.Option	50	50	62	13
net.sf.freecol.common.option.OptionContainer	142	142	206	66
net.sf.freecol.common.resources.Resource.Cleanable	7	7	10	3

interface ^	CLOC	JLOC	LOC	NCLOC
① net.sf.freecol.client.gui.animation.Animations.Procedure	4	4	7	3
① net.sf.freecol.client.gui.DialogHandler	4	4	7	4
① net.sf.freecol.client.gui.label.CargoLabel	18	18	24	6
① net.sf.freecol.client.gui.label.Draggable	5	5	8	3
① net.sf.freecol.client.gui.mapviewer.MapViewer.TileRenderingCal	12	12	15	3
① net.sf.freecol.client.gui.option.OptionUpdater	13	13	17	4
① net.sf.freecol.client.gui.panel.colopedia.ColopediaDetailPanel	16	16	20	5
① net.sf.freecol.client.gui.panel.DropTarget	39	38	53	15
① net.sf.freecol.client.gui.panel.MiniMap.TileRenderingCallback	12	12	15	3
① net.sf.freecol.client.gui.panel.report.LabourData.LocationData.G	0	0	3	3
① net.sf.freecol.client.gui.video.VideoListener	6	6	9	3
① net.sf.freecol.common.i18n.Selector	9	9	12	3
① net.sf.freecol.common.io.sza.AnimationEvent	3	3	5	2
① net.sf.freecol.common.io.sza.ImageAnimationEvent	13	13	17	4
① net.sf.freecol.common.model.BaseProduction	25	25	39	14
① net.sf.freecol.common.model.Constants	7	47	12	5
① net.sf.freecol.common.model.Consumer	44	44	55	11
① net.sf.freecol.common.model.FreeColGameObjectListener	22	22	28	6
① net.sf.freecol.common.model.Locatable	33	33	40	7
① net.sf.freecol.common.model.Location	133	130	166	33
① net.sf.freecol.common.model.Map.SearchHeuristic	3	3	6	3
① net.sf.freecol.common.model.mission.Mission	34	47	40	6
① net.sf.freecol.common.model.Movable	16	16	20	4
① net.sf.freecol.common.model.Nameable	13	13	17	4
① net.sf.freecol.common.model.Named	11	11	14	3
① net.sf.freecol.common.model.Ownable	16	16	20	4
① net.sf.freecol.common.model.pathfinding.CostDecider	29	29	36	7
① net.sf.freecol.common.model.pathfinding.GoalDecider	44	44	49	5
① net.sf.freecol.common.model.Specification.ChildReader	1	0	4	3
① net.sf.freecol.common.model.TradeLocation	45	45	57	12
① net.sf.freecol.common.model.TransactionListener	18	18	22	4
① net.sf.freecol.common.networking.MessageHandler	20	20	26	6
① net.sf.freecol.common.ObjectWithId	9	9	12	3
① net.sf.freecol.common.option.Option	50	50	62	13
① net.sf.freecol.common.option.OptionContainer	142	142	206	66
① net.sf.freecol.common.resources.Resource.Cleanable	7	7	10	3

- **CLOC:** Comment lines of code
- **CLOC(rec):** recursive CLOC
- **JLOC:** Javadoc lines of code
- **JLOC(rec):** recursive JLOC
- **LOC:** Lines of Code
- **LOC(rec):** recursive LOC
- **LOCp:** Lines of product code
- **LOCp(rec):** recursive LOCp
- **LOCt:** Lines of test code
- **LOCt(rec):** recursive LOCt

- **NCLOC:** Non-comment lines of code
- **NCLOCp:** product NCLOC
- **NCLOCp(rec):** recursive product NCLOC
- **NCLOCT:** test NCLOC
- **NCLOCT(rec):** recursive NCLOCT
- **RLOC:** Relative lines of code

No Class metrics existem classes com LOC muito elevado (cerca de quase cinco mil linhas), podendo levar a problemas no código extensos e difíceis de compreender.

Deste tipo de problemas poderia derivar Code Smells como o Long Method, que vemos à frente ser bastante comum ao longo do código do jogo.

Dependency (62335):

package	Cyclic	PDcy	PDpt	PDpt*
net.sf.freecol	43	14	26	47
net.sf.freecol.client	43	16	17	47
net.sf.freecol.client.control	43	19	12	47
net.sf.freecol.client.gui	43	26	19	47
net.sf.freecol.client.gui.action	43	13	8	47
net.sf.freecol.client.gui.animation	43	5	1	47
net.sf.freecol.client.gui.dialog	43	19	3	47
net.sf.freecol.client.gui.images	43	4	1	47
net.sf.freecol.client.gui.label	43	11	4	47
net.sf.freecol.client.gui.mapviewer	43	13	1	47
net.sf.freecol.client.gui.menu	43	11	1	47
net.sf.freecol.client.gui.option	43	12	2	47
net.sf.freecol.client.gui.panel	43	20	13	47
net.sf.freecol.client.gui.panel.colopedi	43	7	2	47
net.sf.freecol.client.gui.panel.report	43	11	2	47
net.sf.freecol.client.gui.plaf	43	8	4	47
net.sf.freecol.client.gui.tooltip	43	10	1	47

interface	Cyclic	Dcy	Dcy*	Dpt	Dpt*	PDcy	PDpt
net.sf.freecol.client.gui.animation.Animations.Procedure	0	0	0	4	922	0	2
net.sf.freecol.client.gui.DialogHandler	0	0	0	12	922	0	5
net.sf.freecol.client.gui.label.CargoLabel	804	2	990	4	921	2	2
net.sf.freecol.client.gui.label.Draggable	0	0	0	4	922	0	2
net.sf.freecol.client.gui.mapviewer.TileRenderingCa	804	1	990	1	921	1	1
net.sf.freecol.client.gui.option.OptionUpdater	0	0	0	2	922	0	1
net.sf.freecol.client.gui.panel.colopedia.ColopediaDetailPanel	0	0	0	4	923	0	1
net.sf.freecol.client.gui.panel.DropTarget	804	3	990	10	921	1	1
net.sf.freecol.client.gui.panel.MiniMap.TileRenderingCallback	804	1	990	1	921	1	1
net.sf.freecol.client.gui.panel.report.LabourData.LocationData.C	804	2	990	2	921	1	1
net.sf.freecol.client.gui.video.VideoListener	0	0	0	0	0	0	0
net.sf.freecol.common.i18n.Selector	0	0	0	2	928	0	1
net.sf.freecol.common.io.sza.AnimationEvent	0	0	0	4	926	0	2
net.sf.freecol.common.io.sza.ImageAnimationEvent	0	1	1	3	925	1	2
net.sf.freecol.common.model.BaseProduction	804	4	990	7	921	1	2
net.sf.freecol.common.model.Constants	0	0	0	43	922	0	11
net.sf.freecol.common.model.Consumer	804	2	990	8	921	1	1

class	Cyclic	Dcy	Dcy*	Dpt	Dpt*	PDcy	PDpt
net.sf.freecol.AllTests	0	5	1 084	0	0	5	0
net.sf.freecol.client.AllTests	0	2	1 003	0	0	2	0
net.sf.freecol.client.ClientOptions	804	26	990	39	921	6	12
net.sf.freecol.client.ClientTestHelper	0	9	991	1	4	7	1
net.sf.freecol.client.control.AllTests	0	1	1 000	2	2	1	2
net.sf.freecol.client.control.ClientInputHandler	804	6	990	1	921	4	1
net.sf.freecol.client.control.ConnectController	804	26	990	15	921	11	6
net.sf.freecol.client.control.FreeColClientHolder	804	12	990	16	921	6	4
net.sf.freecol.client.control.InGameController	804	109	990	93	921	16	11
net.sf.freecol.client.control.InGameController.MoveMode	0	0	0	1	922	0	1
net.sf.freecol.client.control.MapEditorController	804	23	990	10	921	11	5
net.sf.freecol.client.control.MapTransform	804	1	990	11	921	1	3
net.sf.freecol.client.control.MoveTest	0	16	999	1	3	6	1
net.sf.freecol.client.control.PreGameController	804	29	990	20	921	12	4
net.sf.freecol.client.control.SoundController	804	8	990	4	921	5	3
net.sf.freecol.client.FreeColClient	804	37	990	294	921	15	16
net.sf.freecol.client.gui.action.AboutAction	804	3	990	3	921	3	3

- **Cyclic:** Number of cyclic dependencies
- **Dcy:** Number of dependencies
- **Dcy^{*}:** Number of transitive dependencies
- **Dpt:** Number of dependents
- **Dpt^{*}:** Number of transitive dependents
- **PDcy:** Number of package dependencies
- **PDpt:** Number of dependent packages

Observando as metrics apresentadas (Class Metrics e Interface Metrics), podemos chegar à conclusão que há valores extremos relativos a Dcy^{*} e a Dpt^{*}. Estas metrics então podem indicar que alterações num módulo têm um amplo impacto no sistema e assim afetam grande parte do código.

- Code Metric Chidamber-Kemerer (62438)

Class metrics	CBO	DIT	LCOM	NOC	RFC	WMC
class ^						
net.sf.freecol.tools.DesktopEntry	1	1	1	0	21	12
net.sf.freecol.tools.FlagTest	8	6	3	0	59	21
net.sf.freecol.tools.ForestMaker	2	1	1	0	44	22
net.sf.freecol.tools.ForestMaker.ImageLocation	1	1	1	0	6	6
net.sf.freecol.tools.FSGConverter	2	1	1	0	30	23
net.sf.freecol.tools.GenerateDocumentation	4	1	3	0	44	20
net.sf.freecol.tools.InstallerTranslations	2	1	1	0	27	17
net.sf.freecol.tools.MapConverter	7	1	1	0	16	4
net.sf.freecol.tools.MergeTranslations	3	1	1	0	32	14
net.sf.freecol.tools.RiverMaker	1	1	1	0	29	16
net.sf.freecol.tools.SaveGameValidator	2	1	1	0	23	6
net.sf.freecol.tools.Texture2Tile	2	1	1	0	38	18
net.sf.freecol.tools.TranslationReport	2	1	1	0	42	39
net.sf.freecol.tools.TranslationReport.LanguageStatsRecord	1	1	0	0	0	0
net.sf.freecol.util.test.FreeColTestCase	108	3	8	69	107	52
net.sf.freecol.util.test.FreeColTestCase.IndianSettlementBuilder	30	1	1	0	29	22
net.sf.freecol.util.test.FreeColTestCase.MapBuilder	10	1	1	0	24	24
net.sf.freecol.util.test.FreeColTestUtils	15	1	2	0	10	4
net.sf.freecol.util.test.FreeColTestUtils.ColonyBuilder	22	1	1	0	36	21
net.sf.freecol.util.test.MockMapGenerator	13	1	1	0	12	8
net.sf.freecol.util.test.MockPseudoRandom	4	2	1	0	16	13
Total						26 634
Average	20,98	3,53	2,65	0,64	39,80	24,48

CBO(Coupling between objects): média 21

DIT(Depth of Inheritance tree): média 4

LCOM(Lack of Cohesion of Methods): média 3

NOC(Number of Children): média 1

RFC(Response for Class): média 40

WMC(Weighted Method Complexity): média 24

Estas métricas mostram-nos que pode existir problemas com a qualidade do código, pois como estas indicam existe um grande nível de coupling entre objetos, algo que nunca é desejado, pois retira flexibilidade do código criando uma maior dependência entre objetos. Também que o nível de complexidade dos métodos é bastante elevado dificultando a leitura e compreensão do código.

Complexity (62756)

class		OCavg	OCmax	WMC
net.sf.freecol.AllTests		1.00	1	2
net.sf.freecol.client.AllTests		1.00	1	1
net.sf.freecol.client.ClientOptions		2.26	8	52
net.sf.freecol.client.ClientTestHelper		1.00	1	2
net.sf.freecol.client.control.AllTests		1.00	1	1
Public unit test class	ontrol.ClientInputHandler	1.00	1	3
ontrol.ConnectController		5.00	10	60
net.sf.freecol.client.control.FreeColClientHolder		1.14	2	16
net.sf.freecol.client.control.InGameController		4.62	45	786
net.sf.freecol.client.control.InGameController.MoveMode		2.00	2	4
net.sf.freecol.client.control.MapEditorController		1.60	3	24
net.sf.freecol.client.control.MapTransform		1.00	1	2
net.sf.freecol.client.control.MoveTest		1.50	2	3
net.sf.freecol.client.control.PreGameController		1.62	6	39
net.sf.freecol.client.control.SoundController		2.29	5	16
net.sf.freecol.client.FreeColClient		1.73	10	95
net.sf.freecol.client.gui.action.AboutAction		1.00	1	2
net.sf.freecol.client.gui.action.ActionManager		2.88	5	23
net.sf.freecol.client.gui.action.AssignTradeRouteAction		1.67	2	5
net.sf.freecol.client.gui.action.AttackRangedAction		1.67	2	5
net.sf.freecol.client.gui.action.BuildColonyAction		1.33	2	4
net.sf.freecol.client.gui.action.CenterAction		2.00	3	4
net.sf.freecol.client.gui.action.ChangeAction		3.25	7	13
net.sf.freecol.client.gui.action.ChangeWindowedModeAction		1.50	2	3
net.sf.freecol.client.gui.action.ChatAction		1.00	1	3
net.sf.freecol.client.gui.action.ClearOrdersAction		1.00	1	2
net.sf.freecol.client.gui.action.ClopediaAction		1.00	1	2
net.sf.freecol.client.gui.action.ClopediaAction.PanelType		1.00	1	1
net.sf.freecol.client.gui.action.ContinueAction		2.50	4	5
net.sf.freecol.client.gui.action.DebugAction		1.33	2	4
net.sf.freecol.client.gui.action.DedareIndependenceAction		1.00	1	3
net.sf.freecol.client.gui.action.DetermineHighSeasAction		1.33	2	4
net.sf.freecol.client.gui.action.DisbandUnitAction		1.00	1	2
net.sf.freecol.client.gui.action.DisplayBordersAction		1.00	1	2
net.sf.freecol.client.gui.action.DisplayFogOfWarAction		1.33	2	4
net.sf.freecol.client.gui.action.DisplayGridAction		1.00	1	2
net.sf.freecol.client.gui.action.DisplayTileTextAction		1.33	2	4
net.sf.freecol.client.gui.action.DisplayTileTextAction.DisplayText		1.00	1	1
net.sf.freecol.client.gui.action.EndTurnAction		1.00	1	2
net.sf.freecol.common	ui	1.10	11	11
net.sf.freecol.common.metaserver		2.18	61	
net.sf.freecol.common.model		2.25	9171	
net.sf.freecol.common.model.mission		1.38	73	
net.sf.freecol.common.model.pathfind		4.10	160	
net.sf.freecol.common.model.product		3.40	85	
net.sf.freecol.common.networking		1.82	1797	
net.sf.freecol.common.option		1.75	493	
net.sf.freecol.common.resources		2.37	313	
net.sf.freecol.common.sound		2.44	122	
net.sf.freecol.common.util		2.05	583	
net.sf.freecol.common.utils		1.50	6	
net.sf.freecol.meta	server	1.92	50	
net.sf.freecol.meta.server		1.99	161	
net.sf.freecol.server.ai		3.95	2483	
net.sf.freecol.server.ai.military		2.23	138	
net.sf.freecol.server.ai.mission		3.27	1408	
net.sf.freecol.server.control		3.51	541	
net.sf.freecol.server.generator		6.01	463	
net.sf.freecol.server.model		4.52	1577	
net.sf.freecol.server.networking		1.71	36	
net.sf.freecol.tools		4.23	292	
net.sf.freecol.util.test		2.30	161	
Total			30204	
Average		2.58	642.64	

method	CogC	ev(G)	iv(G)	v(G)
net.sf.freecol.client.ClientOptions.getColonyComparatorInternal(int)	3	5	2	7
net.sf.freecol.client.ClientOptions.getDisplayTileText()	0	1	1	1
net.sf.freecol.client.ClientOptions.getModelMessageComparator(Game)	1	2	1	3
net.sf.freecol.client.ClientOptions.getPanelPosition(String)	2	1	3	3
net.sf.freecol.client.ClientOptions.getPanelSizeString()	2	1	2	3
net.sf.freecol.client.ClientOptions.getSpecialOptions()	2	1	2	3
net.sf.freecol.client.ClientOptions.getXMLTagName()	0	1	1	1
net.sf.freecol.client.ClientOptions.isRiverAnimationEnabled()	1	1	2	2
net.sf.freecol.client.ClientOptions.isTerrainAnimationsEnabled()	1	1	2	2
net.sf.freecol.client.ClientOptions.load(FreeColSavegameFile)	3	2	2	4
net.sf.freecol.client.ClientOptions.merge(File)	1	2	1	2
net.sf.freecol.client.ClientOptions.merge(FreeColSavegameFile)	1	2	1	2
net.sf.freecol.client.ClientOptions.regroup(String, String)	1	1	2	2
net.sf.freecol.client.ClientTestHelper.startClient(FreeColServer, Specification)	0	1	1	1
net.sf.freecol.client.ClientTestHelper.stopClient(FreeColClient)	0	1	1	1
net.sf.freecol.client.control.AllTests.suite()	0	1	1	1
net.sf.freecol.client.control.ClientInputHandler.ClientInputHandler(FreeColClient)	0	1	1	1
net.sf.freecol.client.control.ClientInputHandler.handle(Connection, Message)	0	1	1	1
net.sf.freecol.client.control.ClientInputHandler.read(Connection)	0	1	1	1
net.sf.freecol.client.control.ConnectController.connect(String, int)	4	2	3	4
net.sf.freecol.client.control.ConnectController.connectController(FreeColClient)	0	1	1	1
net.sf.freecol.client.control.ConnectController.joinMultiplayerGame(String, String)	10	4	4	9
net.sf.freecol.client.control.ConnectController.login(ServerState, Game, String)	10	2	7	9
net.sf.freecol.client.control.ConnectController.logout(LogoutReason)	9	1	4	8
net.sf.freecol.client.control.ConnectController.mainTitle()	6	2	5	6
net.sf.freecol.client.control.ConnectController.newGame()	4	3	4	5
net.sf.freecol.client.control.ConnectController.requestLogin(String, String, String)	4	4	2	4
net.sf.freecol.client.control.ConnectController.requestLogout(LogoutReason)	8	2	5	6
net.sf.freecol.client.control.ConnectController.startMultiplayerGame(Specification)	3	3	2	4
net.sf.freecol.client.control.ConnectController.startSavedGame(File)	15	4	8	12
net.sf.freecol.client.control.ConnectController.startSinglePlayerGame(Specification)	3	2	3	4
net.sf.freecol.client.control.FreeColClientHolder.askServer()	0	1	1	1
net.sf.freecol.client.control.FreeColClientHolder.currentPlayerIsMyPlayer()	0	1	1	1
net.sf.freecol.client.control.FreeColClientHolder.FreeColClientHolder(FreeColClient)	0	1	1	1
net.sf.freecol.client.control.FreeColClientHolder.getClientOptions()	0	1	1	1
net.sf.freecol.client.control.FreeColClientHolder.getConnectController()	0	1	1	1
net.sf.freecol.client.control.FreeColClientHolder.getFreeColClient()	0	1	1	1
net.sf.freecol.client.control.FreeColClientHolder.getFreeColServer()	0	1	1	1
net.sf.freecol.client.control.FreeColClientHolder.getGame()	0	1	1	1

- **CogC:** Cognitive complexity
- **ev(G):** Essential cyclomatic complexity
- **iv(G):** Design complexity
- **v(G):** Cyclomatic complexity
- **OCavg:** Average operation complexity
- **OCmax:** Maximum operation
- **WMc:** Weighted method complexity
- **v(G)avg:** average cyclomatic complexity
- **v(G)tot:** total cyclomatic complexity

Estas métricas dão-nos a conhecer a complexidade do código.

CogC é uma medida de quão difícil é para um ser humano compreender uma porção de código, por exemplo a classe ConnectController demonstra ser a mais complexa para um uma pessoa ler e compreender.

A complexidade ciclomática é uma métrica de software que quantifica a complexidade de um programa ao contar o número de caminhos independentes através do código. É frequentemente usada para identificar código que pode ser difícil de entender, testar ou manter.

A complexidade ponderada dos métodos é uma métrica que atribui diferentes pesos a diferentes elementos de código com base em sua complexidade. É frequentemente usado para priorizar esforços de refatoração de código.

Martin packaging Metrics (63347)

Abstractness (A): Abstractness mede a proporção de classes/interfaces abstratas em um pacote em relação ao número total de classes, indicando o quão abstratas ou concretas são as classes do pacote.

Afferent Couplings (Ca): Afferent couplings contam o número de pacotes externos que dependem (referenciam) o pacote em questão, refletindo quantos outros pacotes utilizam as classes deste pacote.

Efferent Couplings (Ce): Efferent couplings contam o número de pacotes externos dos quais o pacote em questão depende (faz referência), indicando quantos outros pacotes as classes deste pacote utilizam.

Distância da Sequência Principal (D): A distância da sequência principal quantifica o equilíbrio entre abstração e instabilidade, calculando a diferença absoluta entre a abstração do pacote e sua instabilidade, visando um valor menor mais próximo da "sequência principal".

Instabilidade (I): A instabilidade avalia as dependências do pacote em relação a outros pacotes versus as dependências de outros pacotes em relação a ele, indicando o quão estável ou volátil o pacote é em termos de suas relações com outros pacotes.

package	A	Ca	Ce	D	I
net.sf.freecol.server	0.00	1,104	483	0.70	0.31
net.sf.freecol.server.ai	0.17	1,251	5,977	0.01	0.83
net.sf.freecol.server.ai.military	0.00	15	380	0.04	0.96
net.sf.freecol.server.ai.mission	0.04	435	3,505	0.07	0.89
net.sf.freecol.server.control	0.12	189	5,328	0.09	0.97
net.sf.freecol.server.generator	0.23	11	1,758	0.22	0.99
net.sf.freecol.server.model	0.12	1,159	8,509	0.00	0.88
net.sf.freecol.server.networking	0.00	28	47	0.37	0.63
net.sf.freecol.tools	0.00	0	159	0.00	1.00
net.sf.freecol.util.test	0.00	2,218	278	0.89	0.08
Total					
Average	0.08	1,231.21	1,231.21	0.29	0.51

Ter um valor muito alto de Afferent Couplings e Efferent Couplings pode indicar que o pacote é altamente dependente e influenciado por outros pacotes, o que pode levar a complexidade e fragilidade.

Os valores baixos de Abstractness significam que este tem classes concretas facilitando o seu entendimento.

Um valor médio de Distância da sequência principal significa que esta não possui um equilíbrio mau entre instabilidade e abstração.

Um valor próximo de 0 na I (instability) , ter um valor baixo significa que é altamente estável.

MOOD Metrics (63350):

Project metrics						
project	AHF	AIF	CF	MHF	MIF	PF
project	73,09%	81,12%	3,06%	25,33%	72,83%	9,01%

- **AHF:** Attribuite hiding factor
- **AIF:** Attribuite inheritance factor
- **CF:** Coupling Factor
- **MHF:** Method hiding factor
- **MIF:** Method inheritance factor
- **PF:** Polymorphism factor

Como podemos verificar, existir uma relativa alta percentagem nas métricas de AHF, AIF e MIF. A alta percentagem de AIF e MIF pode levar a uma codebase mais difícil de manter, pois essas percentagens indicam que alterações a certos métodos/atributos irão afetar diversas classes, o que dificulta a manutenção. Por outro lado, as percentagens de CF e PF são relacionáveis, pois enquanto é preferível ter um CF baixo e um PF alto, como CF é relacionado à dependência entre classes e PF à flexibilidade de classes, é difícil encontrar um balanço entre ambas. Preferir CF baixo a PF alto é uma boa solução para assegurar a independência dos métodos de cada classe.

Code Smells

- Data Class, FreeColClientHolder (62333)

```
1 > /...
19
20 package net.sf.freecol.client.control;
21
22 > import ...
23
24
25 v /**
26 * This base class provides access to a
27 * {@link net.sf.freecol.client.FreeColClient} for several subclasses.
28 */
29 16 inheritors ▲ Mike Pope +1
30 e| public class FreeColClientHolder {
31
32     /**
33      * The main client object.
34      */
35     private final FreeColClient freeColClient;
36
37     /**
38      * Simple constructor.
39      */
40
41     /**
42      * @param freeColClient The {@code FreeColClient} to hold.
43      */
44
45     /**
46      * @param freeColClient The {@code FreeColClient} to hold.
47      */
48     1▲ Mike Pope
49     protected FreeColClientHolder(FreeColClient freeColClient) { this.freeColClient = freeColClient; }
50
51
52     /**
53      * Meaningfully named access to the server API.
54      */
55
56     /**
57      * @return The {@code ServerAPI}.
58      */
59     91 usages ▲ Mike Pope
60     public ServerAPI askServer() { return this.freeColClient.askServer(); }
61
62     /**
63      * Check if the current player is the client player.
64      */
65
66     /**
67      * @return True if the client player is current.
68      */
69
```

```
▲ Mike Pope
protected boolean currentPlayerIsMyPlayer() { return this.freeColClient.currentPlayerIsMyPlayer(); }

/**
 * Get the client options.
 *
 * @return The {@code ClientOptions} held by the client.
 */
▲ Mike Pope
protected ClientOptions getClientOptions() { return this.freeColClient.getClientOptions(); }

/**
 * Get the connect controller.
 *
 * @return The {@code ConnectController} held by the client.
 */
▲ Mike Pope
protected ConnectController getConnectController() { return this.freeColClient.getConnectController(); }

/**
 * Get the main client object.
 *
 * @return The {@code FreeColClient} held by this object.
 */
▲ Mike Pope
protected FreeColClient getFreeColClient() { return this.freeColClient; }

/**
 * Get the server.
 *
 * @return The {@code FreeColServer} held by the client.
 */
▲ Mike Pope
protected FreeColServer getFreeColServer() { return this.freeColClient.getFreeColServer(); }

/**
 * Get the game.
 *
 * @return The {@code Game} held by the client.
 */
```

```

/*
 * Mike Pope
protected Game getGame() { return this.freeColClient.getGame(); }

/**
 * Get the Map.
 *
 * @return The game {@code Map}.
 */
* Mike Pope
protected Map getMap() {
    Game game = getGame();
    return (game == null) ? null : game.getMap();
}

/**
 * Get the GUI.
 *
 * @return The {@code GUI} held by the client.
 */
* Mike Pope
protected GUI getGUI() { return this.freeColClient.getGUI(); }

/**
 * Get the client player.
 *
 * @return The {@code Player} associated with the client.
 */
* Mike Pope
protected Player getMyPlayer() { return this.freeColClient.getMyPlayer(); }

/**
 * Get the specification.
 *
 * @return The {@code Specification} held by the game.
 */
* Mike Pope
protected Specification getSpecification() {
    Game game = getGame();
    return (game == null) ? null : game.getSpecification();
}

```

```

/**
 * Get the in-game controller.
 *
 * @return The {@code InGameController} for the client.
 */
* Mike Pope
protected InGameController igc() { return this.freeColClient.getInGameController(); }

/**
 * Get the pre-game controller.
 *
 * @return The {@code PreGameController} for the client.
 */
* Mike Pope
protected PreGameController pgc() { return this.freeColClient.getPreGameController(); }
}

```

Esta classe só tem getters e setters e é pequena, definindo-se como uma **Data Class**.

● Data Class, MapTransform (62333)

```
Linux

package net.sf.freecol.client.control;

import ...

/**
 * Represents a transformation that can be applied to
 * a {@code Tile}.
 *
 * @see #transform(Tile)
 */
22 usages  6 inheritors  ▲ Mike Pope
public abstract class MapTransform {

    /**
     * A panel with information about this transformation.
     */
    2 usages
    private JPanel descriptionPanel = null;

    /**
     * Applies this transformation to the given tile.
     *
     * @param t The {@code Tile} to be transformed,
     */
    6 implementations  ▲ Mike Pope
    public abstract void transform(Tile t);

    /**
     * A panel with information about this transformation. This panel
     * is currently displayed on the
     * {@link net.sf.freecol.client.gui.panel.InfoPanel} when selected,
     * but might be used elsewhere as well.
     *
     * @return The panel or {@code null} if no panel
     *         has been set.
     */
    1 usage  ▲ Mike Pope
    public JPanel getDescriptionPanel() { return descriptionPanel; }
}
```

```
/*
 * Sets a panel that can be used for describing this
 * transformation to the user.
 *
 * @param descriptionPanel The panel.
 */
1 usage  ▲ Mike Pope
public void setDescriptionPanel(JPanel descriptionPanel) { this.descriptionPanel = descriptionPanel; }
}
```

Esta classe só tem getters e setters da classe JPanel e é bastante curta.

● Long Method, startSavedGame() (62333)

```
/*
 * Loads and starts a game from the given file.
 *
 * @param file The saved game.
 * @return True if the game starts successfully.
 */
3 usages ▲ Mike Pope +5
public boolean startSavedGame(File file) {
    final FreeColClient fcc = getFreeColClient();
    final GUI gui = getGUI();
    fcc.setMapEditor(false);

    // Get suggestions for player name, single/multiplayer and
    // public server state from the file, and update the client
    // options if possible.
    final ClientOptions options = getClientOptions();
    final boolean defaultSinglePlayer;
    final boolean defaultPublicServer;
    FreeColSavegameFile fis = null;
    try {
        fis = new FreeColSavegameFile(file);
    } catch (FileNotFoundException fnfe) {
        logger.log(Level.WARNING, msg: "Can not find file: " + file.getName(),
        | fnfe);
        gui.showErrorPanel(fnfe,
            | FreeCol.badFile( messageId: "error.couldNotFind", file));
        return false;
    } catch (IOException ioe) {
        logger.log(Level.WARNING, msg: "Could not load file: " + file.getName(),
        | ioe);
        gui.showErrorPanel(ioe,
            | FreeCol.badFile( messageId: "error.couldNotLoad", file));
        return false;
    }
    options.merge(fis);
    options.fixClientOptions();
}
```

```
List<String> values = null;
try {
    values = fis.peekAttributes(savedKeys);
} catch (Exception ex) {
    logger.log(Level.WARNING, msg: "Could not read from: " + file.getName(),
    | ex);
    gui.showErrorPanel(ex,
        | FreeCol.badFile( messageId: "error.couldNotLoad", file));
    return false;
}
if (values != null && values.size() == savedKeys.size()) {
    String str = values.get(0);
    if (str != null) FreeCol.setName(str);
    defaultSinglePlayer = Boolean.parseBoolean(values.get(1));
    defaultPublicServer = Boolean.parseBoolean(values.get(2));
} else {
    defaultSinglePlayer = true;
    defaultPublicServer = false;
}
```

```

// Reload the client options saved with this game.
final boolean publicServer = defaultPublicServer;
final boolean singlePlayer;
final String serverName;
final InetSocketAddress address;
final int port;
final int sgo = options.getInteger(ClientOptions.SHOW_SAVEGAME_SETTINGS);
boolean show = sgo == ClientOptions.SHOW_SAVEGAME_SETTINGS_ALWAYS
    || (!defaultSinglePlayer
        && sgo == ClientOptions.SHOW_SAVEGAME_SETTINGS_MULTIPLAYER);
if (show) {
    LoadingSavegameInfo lsi = getGUI()
        .showLoadingSavegameDialog(defaultPublicServer,
                                    defaultSinglePlayer);
    if (lsi == null) return false;
    singlePlayer = lsi.isSinglePlayer();
    serverName = lsi.getServerName();
    address = lsi.getAddress();
    port = lsi.getPort();
} else {
    singlePlayer = defaultSinglePlayer;
    serverName = null;
    address = null;
    port = -1;
}
Messages.loadActiveModMessageBundle(options.getActiveMods(),
    FreeCol.getLocale());

FreeColServer fcs = fcc.startServer(publicServer, singlePlayer, file,
    address, port, serverName);
if (fcs == null) return false;

getGUI().refreshGuiUsingClientOptions();

fcc.setFreeColServer(fcs);
fcc.setSinglePlayer(true);
return requestLogin(FreeCol.getName(), nationid: null, fcs.getHost(), fcs.getPort());
}

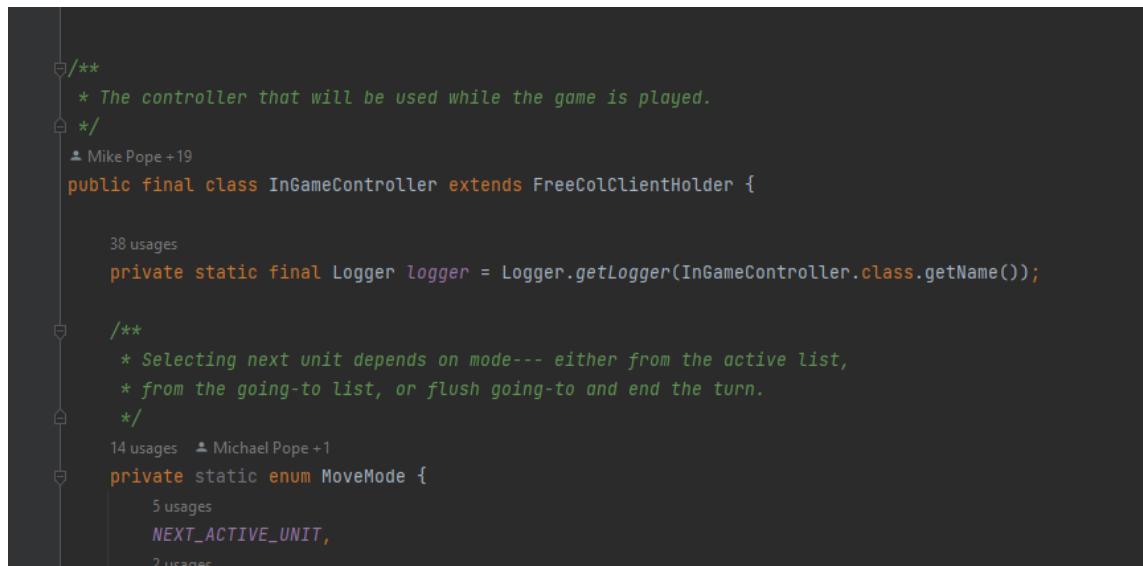
```

O método encontra-se nas linhas 361-448 da classe ConnectController.

Este método desempenha demasiadas funções, duas que são bem distintas e podem ser distinguidas pelos comentários iniciais.

Poderíamos criar um método reloadSavedClientOptions() (com os parâmetros necessários) que iria carregar o “status” do jogo guardado do cliente.

- Large Class, InGameController (62335)



```
/**  
 * The controller that will be used while the game is played.  
 */  
▲ Mike Pope +19  
public final class InGameController extends FreeColClientHolder {  
  
    38 usages  
    private static final Logger logger = Logger.getLogger(InGameController.class.getName());  
  
    /**  
     * Selecting next unit depends on mode--- either from the active list,  
     * from the going-to list, or flush going-to and end the turn.  
     */  
    14 usages ▲ Michael Pope +1  
    private static enum MoveMode {  
        5 usages  
        NEXT_ACTIVE_UNIT,  
        2 usages  
    }  
}
```

A classe *InGameController* como um todo representa um **Large Class** com 5385 linhas.

A classe é responsável por uma grande parte do funcionamento do jogo, tornando-se difícil de manter e compreender.

Como solução poderíamos dividir as funcionalidades desta classe em classes menores especificá-las, cada uma com uma responsabilidade única, facilitando a sua compreensão e manutenção.

- Feature Envy, updateGUI() (62335)

```

573     private void updateGUI(final Tile tile, boolean updateUnit) {
574         displayModelMessages( allMessages: false, endOfTurn: false);
575
576         /* This seems dubious, so I am just displaying the message and
577          * continuing even when there are additional model messages.
578         *
579         * There are multiple cases where we must update the GUI to
580         * avoid errors -- for example when the unit has been removed.
581         *
582         if (displayModelMessages(false, false)) {
583             ; // If messages are displayed they probably refer to the
584             // current unit, so do not update it.
585         } else {
586             /*
587
588             final GUI gui = getGUI();
589             // Update the unit if asked to, or none present, or the
590             // current one is out of moves (but not in Europe or newly
591             // bought/trained units get immediately deselected), or has
592             // been captured.
593             Unit active = gui.getActiveUnit();
594             final boolean update = updateUnit || active == null
595                 || !active.isCandidateForNextActiveUnit()
596                 || !getMyPlayer().owns(active);
597             // Tile is displayed if no new active unit is found,
598             // which is useful when the last unit might have died
599             invokeLater(() -> {
600                 if (update) updateActiveUnit(tile);
601                 gui.updateMapControls();
602                 gui.updateMenuBar();
603             });
604         }
605     }

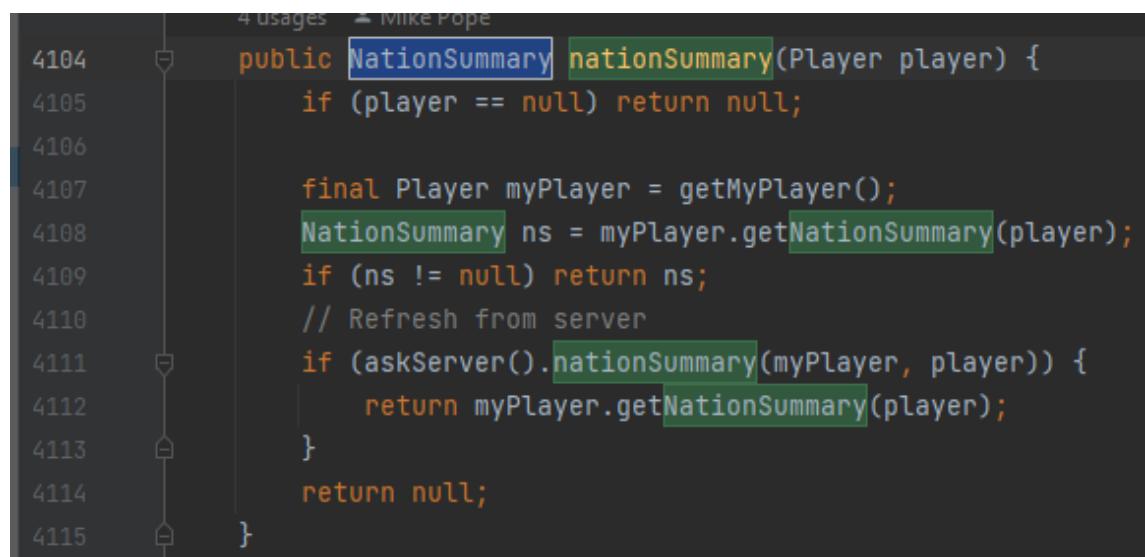
```

O método updateGUI() faz muitas chamadas e manipulações a objetos do tipo GUI.

O método está nas linhas 573-604 da classe InGameController e depende da classe GUI dado que realiza várias chamadas para atualizar os elementos específicos da interface do usuário, podendo indicar um acoplamento forte e perturbar a baixa dependência entre classes.

Uma solução seria reduzir a dependência do método em relação à classe GUI ao mover algumas responsabilidades relacionadas à interface do usuário para métodos da própria classe GUI ou criando novas classes para partilhar as suas responsabilidades.

- Feature Envy, nationSummary() (62335)



```
4104     public NationSummary nationSummary(Player player) {  
4105         if (player == null) return null;  
4106  
4107         final Player myPlayer = getMyPlayer();  
4108         NationSummary ns = myPlayer.getNationSummary(player);  
4109         if (ns != null) return ns;  
4110         // Refresh from server  
4111         if (askServer().nationSummary(myPlayer, player)) {  
4112             return myPlayer.getNationSummary(player);  
4113         }  
4114     }  
4115 }
```

Linhas 4104-4115 da classe InGameController.

O método usa grande parte dos dados de objetos do tipo NationSummary e Player, causando **Feature Envy** por depender demasiado da classe.

Seria uma boa resposta para este problema mover o código relacionado com o NationSummary para dentro da classe Player ou oferecer um método mais geral sem pesar numa outra classe.

Client/Control/InGameController/moveTrade() linhas 2063 até 2084 (62438)

```
/**  
 * Arrive at a settlement with a laden carrier following a move of  
 * MoveType.ENTER_SETTLEMENT_WITH_CARRIER_AND_GOODS.  
 *  
 * @param unit The carrier.  
 * @param direction The direction to the settlement.  
 * @return True if automatic movement of the unit can proceed (never).  
 */  
1 usage  ↳ Mike Pope  
private boolean moveTrade(Unit unit, Direction direction) {  
    if (!askClearGotoOrders(unit)) return false;  
  
    Settlement settlement = getSettlementAt(unit.getTile(), direction);  
    if (settlement instanceof Colony) {  
        final Game game = getGame();  
        final Player player = unit.getOwner();  
        DiplomaticTrade agreement  
            = new DiplomaticTrade(game, TradeContext.TRADE,  
                                  player, settlement.getOwner(), items: null, version: 0);  
        showNegotiationDialog(unit, settlement, agreement,  
                              agreement.getSendMessage(player, settlement), direction);  
    } else if (settlement instanceof IndianSettlement) {  
        askServer().newNativeTradeSession(unit, (IndianSettlement)settlement);  
        changeView(unit, force: false); // Will be deselected on losing moves  
    } else {  
        throw new RuntimeException("Bogus settlement: "  
                               + settlement.getId());  
    }  
    return false;  
}
```

É um speculative generality code smell porque é criada uma função boolean que retorna sempre falso, dando a entender que o valor de bool desta função foi criado para algo futuro porém que nunca foi feito, a função podia ser facilmente ser alterada para void fazendo com que não houvesse necessidade de lidar com o valor bool retornado.

Client/Control//InGameController/moveDirection() linhas 1293-1495 Duplicated code (62438)

```
1283 /**
1284 * Move a unit in a given direction.
1285 *
1286 * Public for the test suite.
1287 *
1288 * @param unit The {@code Unit} to move.
1289 * @param direction The {@code Direction} to move in.
1290 * @param interactive Interactive mode: play sounds and emit errors.
1291 * @return True if automatic movement of the unit can proceed.
1292 */
1293 @
1294 public boolean moveDirection(Unit unit, Direction direction,
1295                               boolean interactive) {
1296     // Is the unit on the brink of reaching the destination with
1297     // this move?
1298     final Location destination = unit.getDestination();
1299     final Tile oldTile = unit.getTile();
1300     boolean destinationImminent = destination != null
1301         && oldTile != null
1302         && Map.isSameLocation(oldTile.getNeighbourOrNull(direction),
1303                               destination);
1304
1305     // Consider all the move types.
1306     final Unit.MoveType mt = unit.getMoveType(direction);
1307     boolean result = mt.isLegal();
1308     switch (mt) {
```

O método é longo e contém uma série de instruções condicionais que lidam com diferentes tipos de movimentos de unidades.

Este método é extenso e recorre à repetição de muitas linhas onde apenas difere em uma palavra para funcionar, o ideal seria criar funções auxiliares para evitar tanta repetição, facilitaria também a compreensão do código.

Server/Control/FreeColServerHolder Data Class (62438)

```
26  /**
27  * This base class provides thread-safe access to a
28  * {@link net.sf.freecol.server.FreeColServer} for several subclasses.
29  */
30  public class FreeColServerHolder {
31
32      /** The main server object. */
33
34      private final FreeColServer freeColServer;
35
36      /**
37      * Constructor.
38      *
39      * @param server The initial value for the server.
40      */
41      protected FreeColServerHolder(FreeColServer server) { this.freeColServer = server; }
42
43      /**
44      * Returns the main server object.
45      *
46      * @return The main server object.
47      */
48
49 }
```

Esta é uma classe muito pequena que possui apenas dois métodos de get, fazendo assim com que ela seja considerada uma data class

- Message Chains, setAvailable() (62756)

```

156  |  /**
157   * Sets a nation's state.
158   *
159   * @param nation The {@code Nation} to set.
160   * @param state The {@code NationState} value to set.
161   */
162  |  public void setAvailable(Nation nation, NationState state) {
163      getGame().getNationOptions().getNations().put(nation, state);
164
165      askServer().setAvailable(nation, state);    Pope, 16/03/2016 09:23
166  }
167

```

linhas 162-166 da classe PreGameController.
“getGame().getNationOptions().getNations().put(nation, state)” é uma “corrente” muito grande de métodos.

- Speculative Generality, startGameInternal() (62756)

```

284  |  private boolean startGameInternal() { (new) 7 minutes ago  Uncommitted changes
285      final FreeCoClient fcc = getFreeCoClient();
286      final Player player = getMyPlayer();
287      final GUI gui = getGUI();
288      // Clear the main display
289      gui.closeMainPanel();
290      gui.closeMenus();
291      gui.closeStatusPanel();
292
293      // Stop the long introduction sound and play the player intro
294      gui.playSound(null);
295      gui.playSound("sound.intro." + player.getNationId());
296
297      final AudioResource defaultPlaylist = ResourceManager.getAudioResource(key: "sound.music.playlist.default", warn: true);
298      if (defaultPlaylist != null) {
299          getFreeCoClient().getSoundController().setDefaultPlaylist(defaultPlaylist.getAllAudio());
300      }
301
302      // Switch to InGame mode
303      fcc.changeClientState(inGame: true);
304      fcc.restoreGUI(player);
305
306      // Clean up autosaves
307      final ClientOptions co = getClientOptions();
308      if (player.isAdmin() && co.getBoolean(ClientOptions.AUTO_SAVE_DELETE)) {
309          String logMe = FreeCoDirectories
310              .removeAutosaves(co.getText(ClientOptions.AUTO_SAVE_PREFIX));
311          if (logMe != null) logger.info(logMe);
312      }
313
314      // Check for debug skipping
315      if (FreeCoDebugger.isDebugEnabled(FreeCoDebugger.DebugMode.MENUS)
316          && FreeCoDebugger.getDebugRunTurns() > 0) {
317          fcc.skipTurns(FreeCoDebugger.getDebugRunTurns());
318          return false;
319      }
320
321      // Starting message if needed
322      if (getGame().getTurn().getNumber() == 1) {
323          player.addStartGameMessage();
324      }
325      igc().nextModelMessage();
326      return true;
327  }

```

linhas 284-328 da classe PreGameController. startGameIntenal() é um método do tipo boolean no entanto os seus valores de return nunca são utilizados para nada, o que nos leva a concluir que o tipo boolean foi atribuído na esperança de que num futuro estes valores viessem a ser utilizados, tal coisa nunca chegou a acontecer. Portanto poderíamos fazer com que o método fosse um void retirando também as linhas de return.

- Duplicate Code, logout() (62756)

```
146 @ ...
147     public boolean logout(LogoutReason reason) {
148         final FreeColClient fcc = getFreeColClient();
149         final Player player = fcc.getMyPlayer();
150         logger.info( msg: "Logout end for client " + player.getName()
151                     | + ": " + reason);
152
153         askServer().disconnect();
154
155         switch (reason) {
156             case DEFEATED: case QUIT:
157                 fcc.Logout( inGame: false );
158                 SwingUtilities.invokeLater( () -> {
159                     mainTitle();
160                 });
161                 break;
162             case LOGIN: // Occurs when you Open a new game during a current one
163                 fcc.logout( inGame: false );
164                 break;
165             case MAIN_TITLE: // All the way back to the MainPanel
166                 fcc.logout( inGame: false );
167                 /*
168                  * This should not be needed as it was mainTitle that
169                  * initiated the logout.
170                  *
171                  * mainTitle();
172                  */
173                 break;
174             case NEW_GAME: // Back to the NewPanel
175                 fcc.logout( inGame: false );
176                 /*
177                  * This should not be needed as it was mainTitle that
178                  * initiated the logout.
179                  *
180                  * newGame();
181                  */
182                 break;
183         }
184     }
```

Está presente nas linhas 146-207 da classe ConnectController. logout() tem chamadas de si mesmo duplicadas, que poderiam estar “fundidas” como é feito na linha 155.

Com esta solução ficaria “case LOGIN:case MAIN_TITLE: case NEW_GAME: fcc.logout(false)”.

- Long Method, csKill() (63347)

Este método csKill(linhas 629-710, src/net/sf/freecol/server/model/ServerPlayer) é um método grande com repetitividade da verificação se uma variável é null ou não. Isto pode ser simplificado de forma a chamar um único método que faça esse trabalho, em vez de repetir as mesmas linhas de código várias e várias vezes.

```

629  @
630    private void csKill(ChangeSet cs) {
631      final Game game = getGame();
632
633      setDead(true);
634      cs.addPartial(See.all(), fcgo: this, ...fields: "dead", Boolean.TRUE.to
635      cs.add(See.all(), new SetDeadMessage( player: this));
636
637      // Clean up missions and remove tension/alarm/stance.
638      for (Player other : game.getLivePlayerList( ...exclude: this)) {
639          if (isEuropean() && other.isIndian()) {
640              for (IndianSettlement is : other.getIndianSettlementList())
641                  ServerIndianSettlement sis = (ServerIndianSettlement)is;
642                  if (is.hasMissionary( player: this)) sis.csKillMissionary(
643                      is.getTile().cacheUnseen()); //+til
644                  sis.removeAlarm( player: this); //+til
645              }
646              other.removeTension( player: this);
647              other.setStance( player: this, newStance: null);
648          }
649
650          // All other units must disappear
651          Set<Unit> tiles = new HashSet<>( initialCapacity: 64);
652          for (FreeColGameObject fcgo : game.getFreeColGameObjectList()) {
653              if (fcgo instanceof Unit) {
654                  Unit u = (Unit)fcgo;
655                  if (u.hasTile() && !this.owns(u)) {
656                      Tile t = u.getTile();
657                      cs.addRemove(See.only(this), t, u);
658                      tiles.add(t);
659                  }
660              }
672
673      }
674
675      tiles.addAll(game.getMap().getTileList(monitorKey.equals( key: this, tile::getOwner)));
676      for (Tile t : tiles) {
677          t.cacheUnseen(); //+til
678          t.changeOwnership( player: null, settlement: null); //+til
679      }
680
681      // Remove units
682      for (Unit u : getUnitSet()) {
683          if (u.hasTile()) tiles.add(u.getTile());
684          ((ServerUnit)u).csRemove(See.perhaps().always(this),
685                                  u.getLocation(), cs); //+vis(this)
686      }
687
688      cs.add(See.perhaps().always(this), tiles);
689
690      // Remove European stuff
691      if (market != null) {
692          market.dispose();
693          market = null;
694      }
695      if (monarch != null) {
696          monarch.dispose();
697          monarch = null;
698      }
699      if (europe != null) {
700          europe.dispose();
701          europe = null;
702      }
703      currentFather = null;
704      if (foundingFathers != null) foundingFathers.clear();
705      if (offeredFathers != null) offeredFathers.clear();
706      // FIXME: stance and tension?
707      if (tradeRoutes != null) tradeRoutes.clear();
708      // Retaining model messages for now
709      // Retaining history for now
710

```

• Speculative Generality, startGameInternal() (63347)

Class ServerPlayer (src/net/sf/freecol/server/model/ServerPlayer).

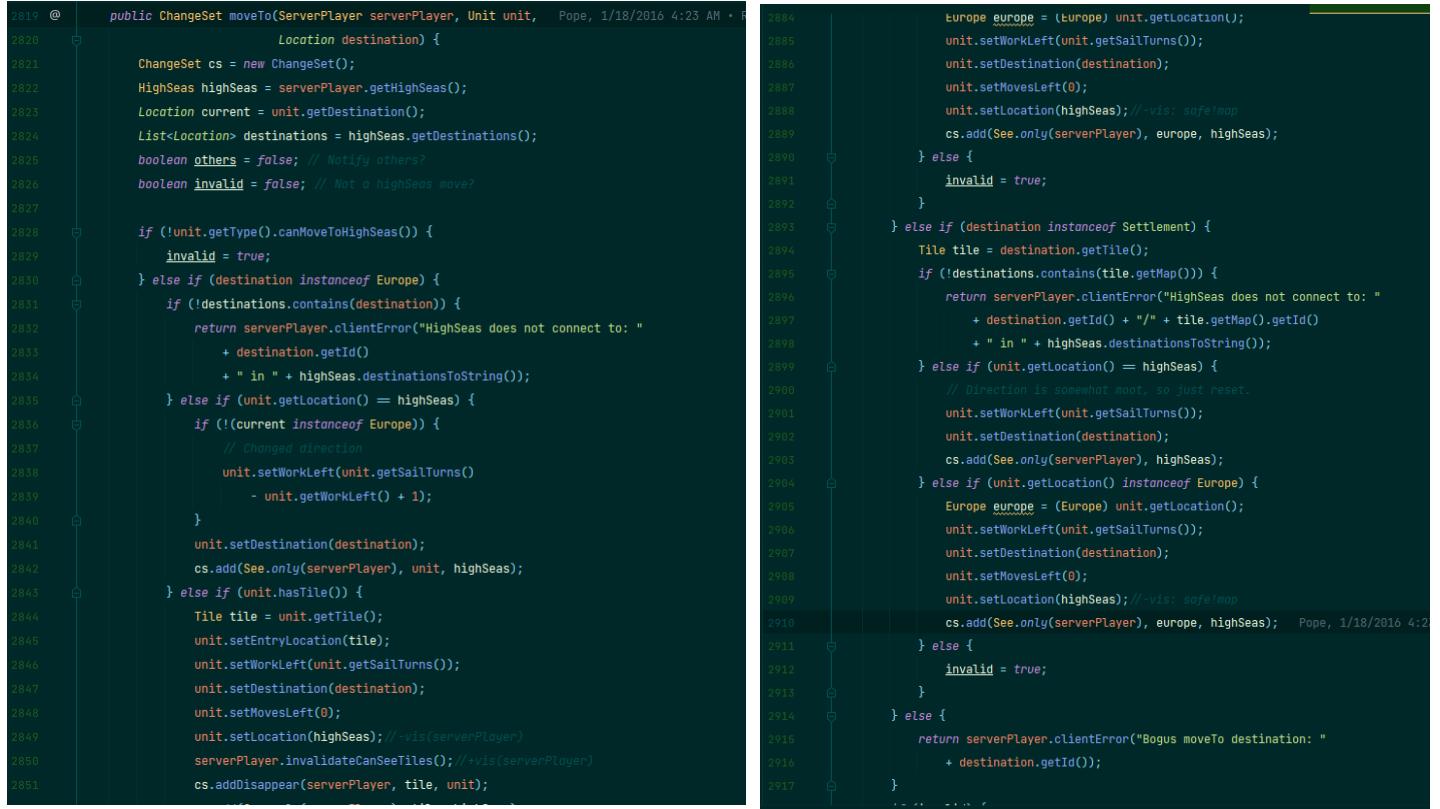
Em específico na linha 157, onde é criada uma constante com o intuito de ser utilizada no decorrer do desenvolvimento do código, porém acabou por nunca ter tido utilidade e por conseguinte poderia ser simplesmente eliminado de forma a deixar o código mais simples.

```
142     private static final Logger logger = Logger.getLogger(ServerPlayer.class.getName());  
143  
144     // FIXME: move to options or spec?  
145     public static final int ALARM_RADIUS = 2;  
146     public static final int ALARM_TILE_IN_USE = 2;  
147  
148     // checkForDeath result type  
149     public static enum DeadCheck {  
150         IS_DEAD,  
151         IS_DEFEATED,  
152         IS_AUTORECRUIT,  
153         IS_ALIVE  
154     };  
155  
156     // Penalty for destroying a settlement (Col1)  
157     public static final int SCORE_SETTLEMENT_DESTROYED = -5;  
158  
159     // Penalty for destroying a nation (FreeCol extension)  
160     public static final int SCORE_NATION_DESTROYED = -50;  
161  
162     // Gold converts to score at 1 pt per 1000 gp (Col1)  
163     public static final double SCORE_GOLD = 0.001;  
164  
165     // Score bonus for each founding father (Col1)  
166     public static final int SCORE_FOUNDING_FATHER = 5;  
167  
168     // Percentage bonuses for being the 1st,2nd and 3rd player to  
169     // achieve independence. (Col1)  
170     public static final int SCORE_INDEPENDENCE_BONUS_FIRST = 100;  
171     public static final int SCORE_INDEPENDENCE_BONUS_SECOND = 50;  
172     public static final int SCORE_INDEPENDENCE_BONUS_THIRD = 25;  
173
```

Duplicated code (63347)

server/control/InGameController/moveTo,
linhas 2819, 2926

Na classe moveTo existem muitas porções de código repetidas que poderiam ser simplificadas.



```
2819 @Override
2820 public ChangeSet moveTo(ServerPlayer serverPlayer, Unit unit, Location destination) {
2821     ChangeSet cs = new ChangeSet();
2822     HighSeas highSeas = serverPlayer.getHighSeas();
2823     Location current = unit.getDestination();
2824     List<Location> destinations = highSeas.getDestinations();
2825     boolean others = false; // Notify others?
2826     boolean invalid = false; // Not a highSeas move?
2827
2828     if (!unit.getType().canMoveToHighSeas()) {
2829         invalid = true;
2830     } else if (destination instanceof Europe) {
2831         if (!destinations.contains(destination)) {
2832             return serverPlayer.clientError("HighSeas does not connect to: "
2833                     + destination.getId()
2834                     + " in " + highSeas.destinationsToString());
2835         } else if (unit.getLocation() == highSeas) {
2836             if (!(current instanceof Europe)) {
2837                 // Changed direction
2838                 unit.setWorkLeft(unit.getSailTurns()
2839                             - unit.getWorkLeft() + 1);
2840             }
2841             unit.setDestination(destination);
2842             cs.add(See.only(serverPlayer), unit, highSeas);
2843         } else if (unit.hasTile()) {
2844             Tile tile = unit.getTile();
2845             unit.setEntryLocation(tile);
2846             unit.setWorkLeft(unit.getSailTurns());
2847             unit.setDestination(destination);
2848             unit.setMovesLeft(0);
2849             unit.setLocation(highSeas); //vis(serverPlayer)
2850             serverPlayer.invalidateCanSeeTiles(); //vis(serverPlayer)
2851             cs.addDisappear(serverPlayer, tile, unit);
2852         }
2853     }
2854
2855     if (invalid) {
2856         return serverPlayer.clientError("Bogus moveTo destination: "
2857             + destination.getId());
2858     }
2859
2860     if (highSeas != null) {
2861         highSeas.setLastMoveTime();
2862     }
2863
2864     if (others) {
2865         cs.add(See.only(serverPlayer), highSeas);
2866     }
2867
2868     if (invalid) {
2869         return serverPlayer.clientError("Bogus moveTo destination: "
2870             + destination.getId());
2871     }
2872
2873     if (highSeas != null) {
2874         highSeas.setLastMoveTime();
2875     }
2876
2877     if (invalid) {
2878         return serverPlayer.clientError("Bogus moveTo destination: "
2879             + destination.getId());
2880     }
2881
2882     if (highSeas != null) {
2883         highSeas.setLastMoveTime();
2884     }
2885
2886     if (invalid) {
2887         return serverPlayer.clientError("Bogus moveTo destination: "
2888             + destination.getId());
2889     }
2890
2891     if (invalid) {
2892         return serverPlayer.clientError("Bogus moveTo destination: "
2893             + destination.getId());
2894     }
2895
2896     if (invalid) {
2897         return serverPlayer.clientError("Bogus moveTo destination: "
2898             + destination.getId());
2899     }
2900
2901     if (invalid) {
2902         return serverPlayer.clientError("Bogus moveTo destination: "
2903             + destination.getId());
2904     }
2905
2906     if (invalid) {
2907         return serverPlayer.clientError("Bogus moveTo destination: "
2908             + destination.getId());
2909     }
2910
2911     if (invalid) {
2912         return serverPlayer.clientError("Bogus moveTo destination: "
2913             + destination.getId());
2914     }
2915
2916     if (invalid) {
2917         return serverPlayer.clientError("Bogus moveTo destination: "
2918             + destination.getId());
2919     }
2920
2921     if (invalid) {
2922         return serverPlayer.clientError("Bogus moveTo destination: "
2923             + destination.getId());
2924     }
2925
2926     if (invalid) {
2927         return serverPlayer.clientError("Bogus moveTo destination: "
2928             + destination.getId());
2929     }
2930
2931     if (invalid) {
2932         return serverPlayer.clientError("Bogus moveTo destination: "
2933             + destination.getId());
2934     }
2935
2936     if (invalid) {
2937         return serverPlayer.clientError("Bogus moveTo destination: "
2938             + destination.getId());
2939     }
2940
2941     if (invalid) {
2942         return serverPlayer.clientError("Bogus moveTo destination: "
2943             + destination.getId());
2944     }
2945
2946     if (invalid) {
2947         return serverPlayer.clientError("Bogus moveTo destination: "
2948             + destination.getId());
2949     }
2950
2951     if (invalid) {
2952         return serverPlayer.clientError("Bogus moveTo destination: "
2953             + destination.getId());
2954     }
2955
2956     if (invalid) {
2957         return serverPlayer.clientError("Bogus moveTo destination: "
2958             + destination.getId());
2959     }
2960
2961     if (invalid) {
2962         return serverPlayer.clientError("Bogus moveTo destination: "
2963             + destination.getId());
2964     }
2965
2966     if (invalid) {
2967         return serverPlayer.clientError("Bogus moveTo destination: "
2968             + destination.getId());
2969     }
2970
2971     if (invalid) {
2972         return serverPlayer.clientError("Bogus moveTo destination: "
2973             + destination.getId());
2974     }
2975
2976     if (invalid) {
2977         return serverPlayer.clientError("Bogus moveTo destination: "
2978             + destination.getId());
2979     }
2980
2981     if (invalid) {
2982         return serverPlayer.clientError("Bogus moveTo destination: "
2983             + destination.getId());
2984     }
2985
2986     if (invalid) {
2987         return serverPlayer.clientError("Bogus moveTo destination: "
2988             + destination.getId());
2989     }
2990
2991     if (invalid) {
2992         return serverPlayer.clientError("Bogus moveTo destination: "
2993             + destination.getId());
2994     }
2995
2996     if (invalid) {
2997         return serverPlayer.clientError("Bogus moveTo destination: "
2998             + destination.getId());
2999     }
2999 }
```

Long Method - rearrangeColony() (63350)

```

public Set<AIUnit> rearrangeColony(LogBuilder lb) {
    final AIMain aiMain = getAIMain();
    Set<AIUnit> result = new HashSet<>();

    // First check if it is collapsing.
    if (colony.getUnitCount() <= 0) {
        if (!avertAutoDestruction()) return result;
    }

    // Skip this colony if it does not yet need rearranging.
    final int turn = getGame().getTurn().getNumber();
    if (rearrangeTurn.getNumber() > turn) {
        if (colony.getCurrentlyBuilding() == null
            && colonyPlan != null
            && colonyPlan.getBestBuildableType() != null) {
            logger.warning(colony.getName() + " could be building but"
                + " is asleep until turn: " + rearrangeTurn.getNumber()
                + "( > " + turn + ")");
        } else {
            return result;
        }
    }

    final Tile tile = colony.getTile();
    final Player player = colony.getOwner();
    final Specification spec = getSpecification();
    lb.add("\n ", colony.getName());

    // For now, cap the rearrangement horizon, because confidence
    // that we are triggering on all relevant changes is low.
    int nextRearrange = 15;

    // See if there are neighbouring LCRs to explore, or tiles
    // to steal, or just unclaimed tiles (a neighbouring settlement
    // might have disappeared or relinquished a tile).
    // This needs to be done early so that new tiles can be
    // included in any new colony plan.
    exploreLCRs();
    stealTiles(lb);
    for (Tile t : transform(tile.getSurroundingTiles(1,1),
        t2 -> !player.owns(t2) && player.canClaimForSettlement(t2)))
        for (Tile t : transform(tile.getSurroundingTiles(1,1),
            t2 -> !player.owns(t2) && player.canClaimForSettlement(t2))) {
            AIMessage.askClaimLand(t, this, 0);
            if (player.owns(t)) lb.add(", claimed tile ", t);
        }

        // Update the colony plan.
        if (colonyPlan == null) colonyPlan = new ColonyPlan(aiMain, colony);
        colonyPlan.update();

        // Now that we know what raw materials are available in the
        // colony plan, set the current buildable, first backing out
        // of anything currently being built that is now impossible.
        // If a buildable is chosen, refine the worker allocation in
        // the colony plan in case the required building materials
        // have changed.
        BuildableType oldBuild = colony.getCurrentlyBuilding();
        BuildableType build = colonyPlan.getBestBuildableType();
        if (build != oldBuild) {
            List<BuildableType> queue = new ArrayList<>(1);
            if (build != null) queue.add(build);
            AIMessage.askSetBuildQueue(this, queue);
            build = colony.getCurrentlyBuilding();
        }
        colonyPlan.refine(build, lb);

        // Collect all potential workers from the colony and from the
        // being careful not to disturb existing non-colony missions.
        // Note the special case of a unit aiming to build a colony on
        // tile, which happens regularly with the initial AI colony.
        // Remember where the units came from.
        List<Unit> workers = colony.getUnitList();
        List<UnitWas> was = transform(workers, alwaysTrue(),
            u -> new UnitWas(u));
        final Predicate<Unit> workerPred = u -> {
            AIUnit validAIU;
            return u.isPerson() && !u.hasAbility(Ability.REF_UNIT)
        };

        // Argh. We may have chosen to build something we can no
        // longer build due to some limitation. Try to find a
        // replacement, but do not re-refine as that process is
        // sufficiently complex that we can not be confident that this
        // will not loop indefinitely. The compromise is to just
        // rearrange next turn until we get out of this state.
        if (build != null && !colony.canBuild(build)) {
            BuildableType newBuild = colonyPlan.getBestBuildableType();
            lb.add(", renamed building ", build.getSuffix(),
                " (", colony.getNoBuildReason(build, null), ")");
            List<BuildableType> queue = new ArrayList<>();
            if (newBuild != null) queue.add(newBuild);
            AIMessage.askSetBuildQueue(this, queue);
            nextRearrange = 1;
        }

        // Now that all production has been stabilized, plan to
        // rearrange when the warehouse hits a limit.
        if (colony.getNetProductionOf(spec.getPrimaryFoodType()) < 0) {
            GoodsType food = spec.getPrimaryFoodType();
            int net = colony.getNetProductionOf(food);
            int when = colony.getGoodsCount(food) - net;
            nextRearrange = Math.max(0, Math.min(nextRearrange, when-1));
        }

        int warehouse = colony.getWarehouseCapacity();
        for (GoodsType g : transform(spec.getStorableGoodsTypeList(),
            gt -> !gt.isFoodType())) {
            int have = colony.getGoodsCount(g);
            int net = colony.getAdjustedNetProductionOf(g);
            if (net >= 0 && (have > warehouse || g.limitIgnored())) continue;
            int when = (net < 0) ? (have / -net - 1)
                : Integer.MAX_VALUE;
            nextRearrange = Math.max(1, Math.min(nextRearrange, when));
        }

        // Return any units with the wrong mission
        for (Unit u : colony.getUnitList()) {
            final AIUnit aiu = getAIUnit(u);
            if (!aiu.tryWorkInsideColonyMission(this, lb)) result.add(aiu);
        }

        final Predicate<Unit> workerPred = u -> {
            AIUnit validAIU;
            return u.isPerson() && !u.hasAbility(Ability.REF_UNIT)
        };

        // Assign the workers according to the colony plan.
        // ATM we just accept this assignment unless it failed, in
        // which case restore original state.
        AIPlayer aiPlayer = getAIOwner();
        LogBuilder aw = new LogBuilder(256);
        boolean preferScouts = aiPlayer.scoutsNeeded() > 0;
        Colony scratch = colonyPlan.assignWorkers(new ArrayList<>(workers)
            preferScouts, aw);

        if (scratch == null) {
            lb.add(", failed to assign workers.");
            rearrangeTurn = new Turn(turn + 1);
            return result;
        }
        lb.add(", assigned ", workers.size(), " workers");

        // Apply the arrangement, and give suitable missions to all units.
        AIMessage.askRearrangeColony(this, workers, scratch);

        // Emergency recovery if something broke and the colony is empty.
        if (colony.getUnitCount() <= 0) {
            lb.add(", autodestruct detected");
            StringBuilder sb = new StringBuilder(64);
            sb.append("Autodestruct at ").append(colony.getName())
                .append(" in ").append(turn).append(':');
            for (UnitWas uw : was) sb.append('\n').append(uw);
            logger.warning(sb.toString());
            if (!avertAutoDestruction()) return result;
        }

        // Argh. We may have chosen to build something we can no
        for (Unit u : colony.getUnitList()) {
            final AIUnit aiu = getAIUnit(u);
            if (!aiu.tryWorkInsideColonyMission(this, lb)) result.add(aiu);
        }

        // Allocate pioneers if possible.
        int tipSize = tileImprovementPlans.size();
        if (tipSize > 0) {
            List<Unit> pioneers
                = transform(tile.getUnits(), u -> u.getPioneerScore() > 0,
                    Function.<Unit>identity(), pioneerComparator);
            for (Unit u : pioneers) {
                final AIUnit aiu = getAIUnit(u);
                if (aiu.tryPioneeringMission(lb)) {
                    if (-tipSize <= 0) break;
                }
            }
        }

        for (Unit u : tile.getUnitList()) {
            final AIUnit aiu = getAIUnit(u);
            if (!aiu.trySomeUsefulMission(colony, lb)) result.add(aiu);
        }

        // Log the changes.
        build = colony.getCurrentlyBuilding();
        String buildStr = (build != null) ? build.toString()
            : ((build = colonyPlan.getBestBuildableType()) != null
            ? "unexpected-null(" + build + ")"
            : "expected-null");
        lb.add(", building ", buildStr, ", population ", colony.getUnitCount(),
            ", rearrange ", nextRearrange, "\n");
        lb.shrink("\n");
        for (UnitWas uw : was) lb.add("\n ", uw);

        // Set the next rearrangement turn.
        rearrangeTurn = new Turn(turn + nextRearrange);

        return result;
    }
}

```

O método encontra-se nas linhas 237 a 426 na classe AIColony

O método em questão usa demasiadas funções, umas que podem ser separadas entre diversas outras funções

Para simplificar o método, era possível separar as funções das 2 1^{as} imagens num método separado “needsArranging()” para 1º identificar se alguma colónia e os seus workers tem condição para dar rearrange

Feature Envy, startMapEditor() (63350)

```
public void startMapEditor() {
    final FreeColClient fcc = getFreeColClient();
    final GUI gui = getGUI();
    try {
        final Specification specification = getDefaultSpecification();

        final List<FreeColModFile> mods = getClientOptions().getActiveMods();
        final boolean specificationChanges = mods.stream().anyMatch(m -> m.hasSpecification());
        specification.loadMods(mods);

        fcc.setMapEditor(true);
        final FreeColServer freeColServer
            = new FreeColServer(false, false, specification, null, 0, null);
        fcc.setFreeColServer(freeColServer);
        ServerGame serverGame = freeColServer.getGame();
        requireNativeNations(serverGame);
        fcc.setGame(serverGame);
        fcc.setMyPlayer(null);
        gui.playSound(null);
        gui.closeMainPanel();
        gui.closeMenus();
        //fcc.changeClientState(true);
        //gui.changeView((Tile)null);
        gui.startMapEditorGUI();

        if (specificationChanges) {
            gui.showInformationPanel("mapEditor.loadedWithMods");
        }
    } catch (IOException e) {
        gui.showErrorPanel(StringTemplate
            .template("server.initialize"));
        return;
    }
}
```

O método startMapEditor() faz muitas chamadas e manipulações a objetos dos tipos GUI e FCC.

Encontra-se nas linhas 143 a 176 da classe MapEditorController usando classes GUI e FCC causando **Feature Envy** por depender demasiado destas

Seria uma boa resposta para este problema mover o código relacionado com o método para dentro da classe referida

Message Chain, determineStances() (63350)

```
/*
 * Determines the stances towards each player.
 * That is: should we declare war?
 *
 * @param lb A {@code LogBuilder} to log to.
 */
private void determineStances(LogBuilder lb) {
    final Player player = getPlayer();
    lb.mark();

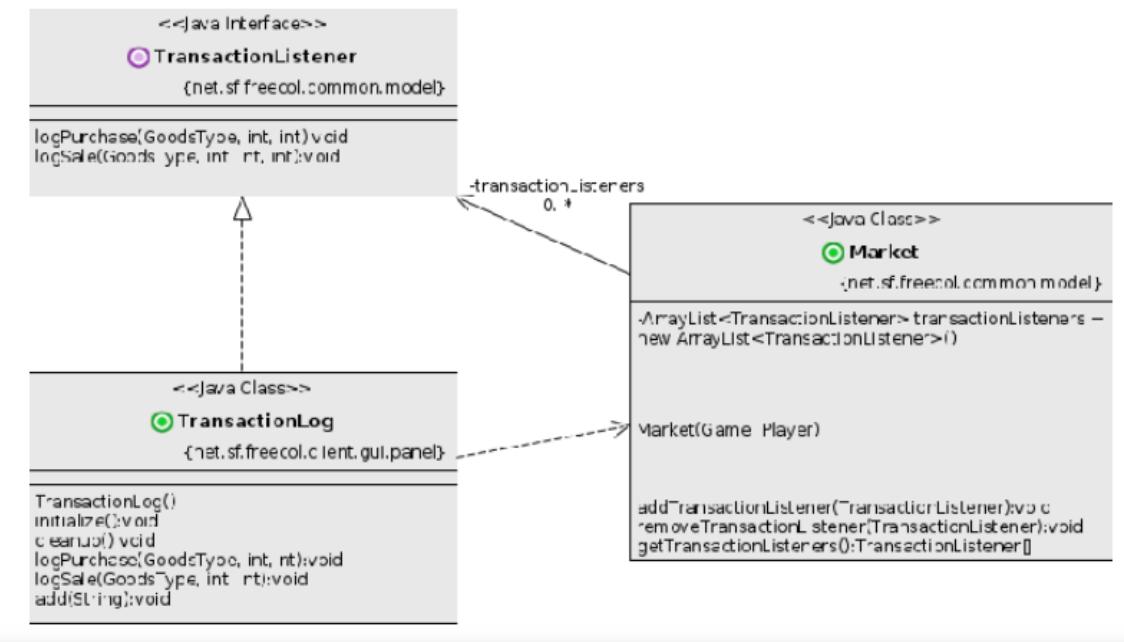
    for (Player p : getGame().getLivePlayerList(player)) {
        Stance newStance = determineStance(p);
        if (newStance != player.getStance(p)) {
            getAIMain().getFreeColServer().getInGameController()
                .changeStance(player, newStance, p, true);
            lb.add(" ", p.getDebugName(), "->", newStance, " ");
        }
    }
    if (lb.grew("\n  Stance changes:")) lb.shrink(", ");
}
```

linhas 164-177 da classe NativeAIPlayer

getAIMain().getFreeColServer().getInGameController().changeStance(player,newStance, p, TRUE) é uma linha com uma quantidade excessiva de métodos, sendo útil a separação dessa tal linha em uma função que obtenha os 3 aspectos que chama e que executa o código em “changeStance()”

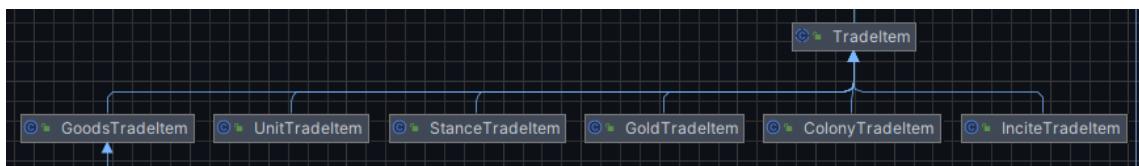
Design Patterns

- Observer Pattern (62333)



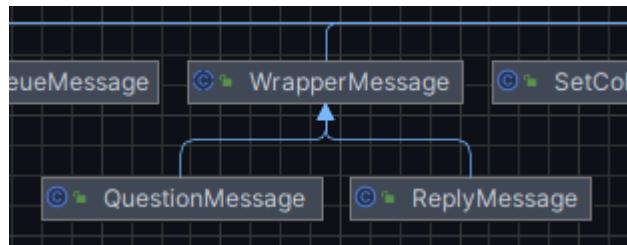
TransactionListener notifica quando existe uma mudança no Mercado do jogo.

- Template Pattern (62333)



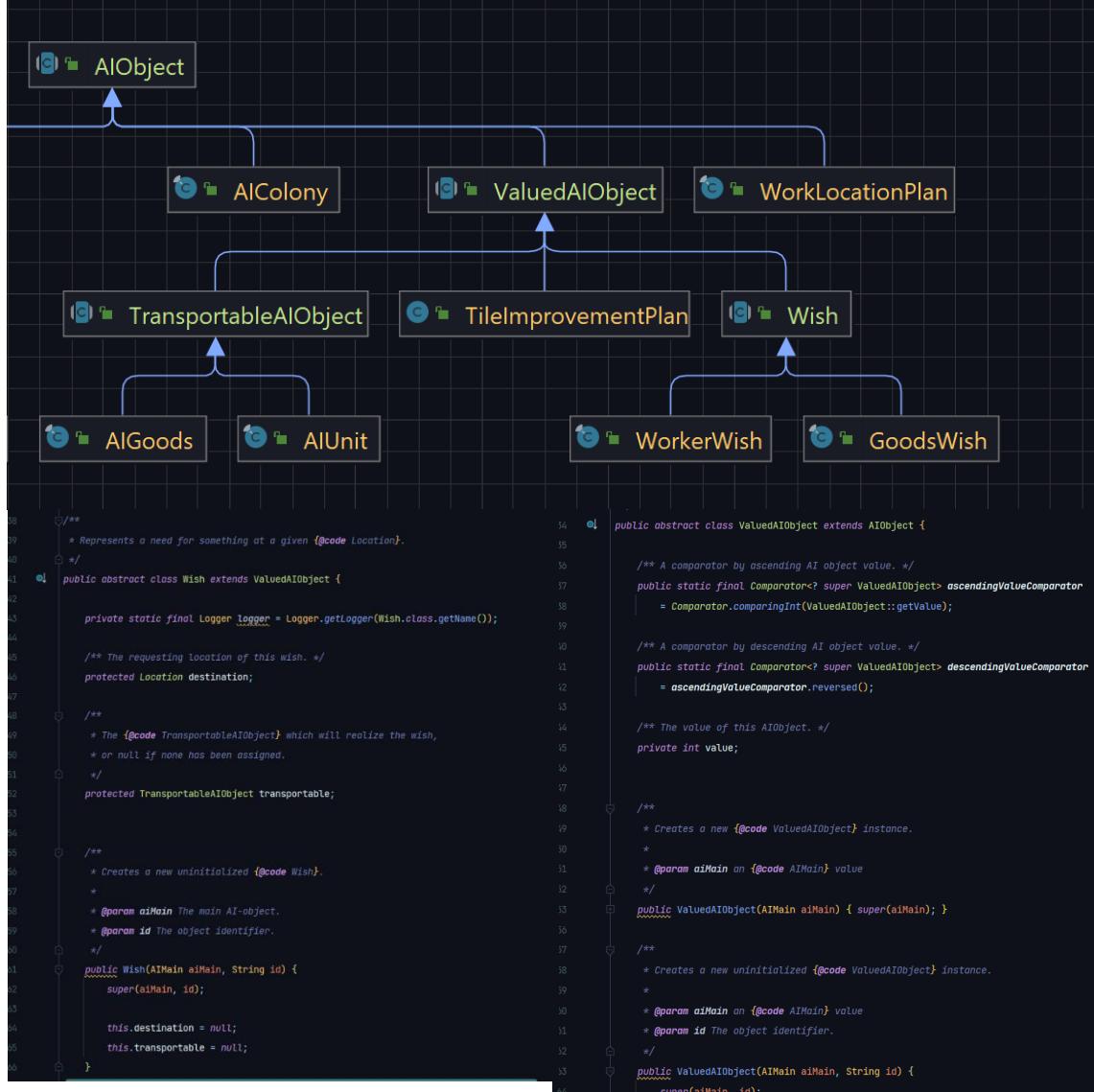
Define o algoritmo de transação de itens.

- Template Pattern (62333)



WrapperMessage é a classe abstrata e as QuestionMessage e ReplyMessage são as classes concretas, que definem o algoritmo das Mensagens.

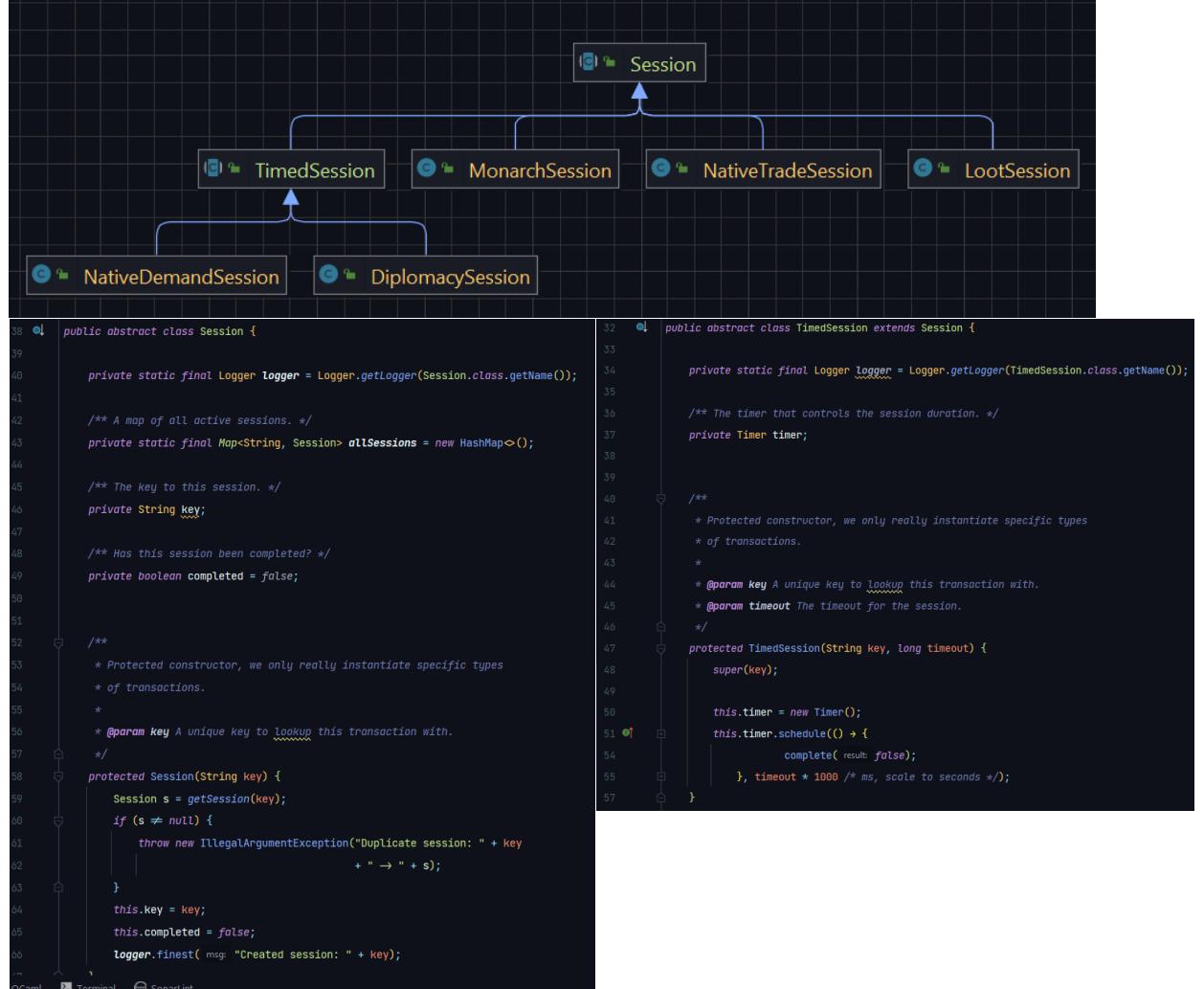
Decorator (62335)



Localização: src/net/sf/freecol/server/ai/AIObject.java

O Decorator é um design pattern estrutural que permite adicionar novos comportamentos aos objetos, colocando esses objetos dentro de objetos especiais que contêm os comportamentos. Como são estendidas classes a outras classes que por sua vez têm funcionalidades específicas, então podemos considerar isto um decorator.

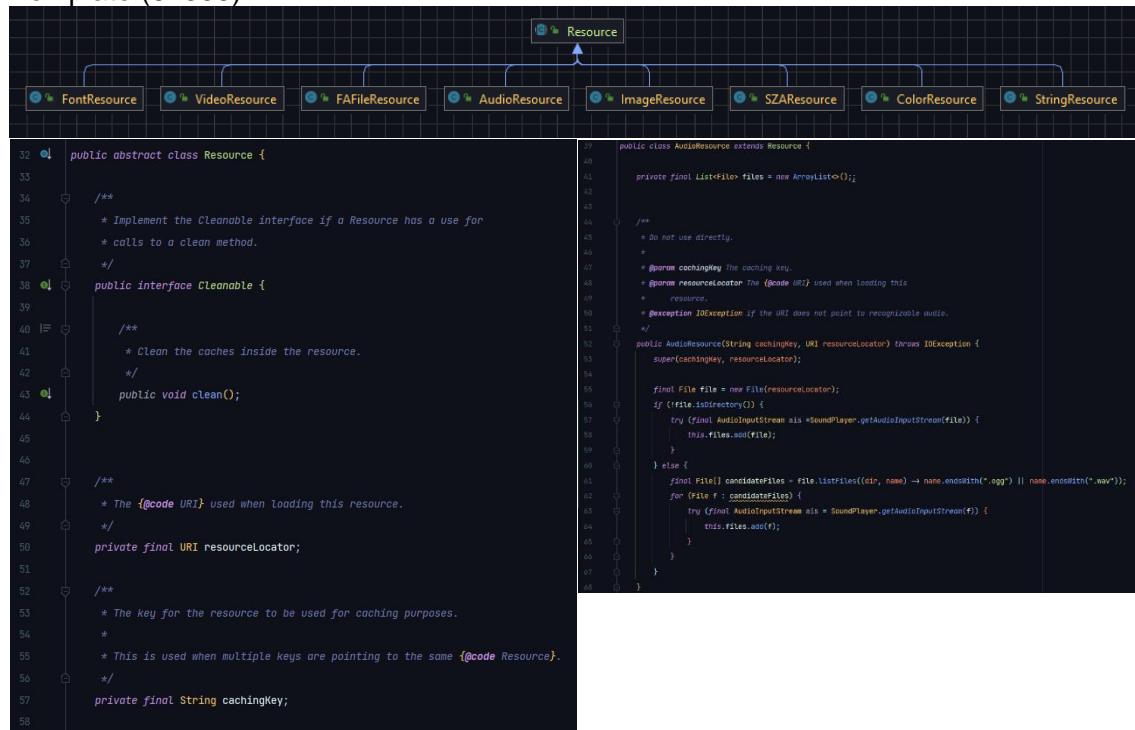
Template (62335)



Localização: src/net/sf/freecol/server/model/Session.java

O Template Method é um design pattern que define a estrutura de um algoritmo na superclasse, mas permite que subclasses substituam etapas específicas do algoritmo sem alterar a sua estrutura. Tendo em conta as extensões de classes abstratas podemos concluir que é um template.

Template (62335)



Localização: src/net/sf/freecol/common/resources/Resource.java

O Template Method é um design pattern que define a estrutura de um algoritmo na superclasse, mas permite que subclasses substituam etapas específicas do algoritmo sem alterar a sua estrutura. Tendo em conta as extensões de classes abstratas podemos concluir que é um template

src/net/sf/freecol/common/model/mission/AbstractMission.java

Template Pattern (62438)



```
31     /**
32      * The AbstractMission provides basic methods for building Missions.
33      */
34  @ 3 usages 3 inheritors  Michael Vehrs +4
35  public abstract class AbstractMission extends FreeColGameObject implements Mission {
36
37      /**
38       * The Unit this mission was assigned to. Must not be null.
39       */
40      private Unit unit;
41
42      /**
43       * The number of times this mission should be repeated. Defaults
44       * to 1.
45       */
46      private int repeatCount = 1;
47
48      /**
49       * The number of turns this mission will take to carry out. In
50       * most cases, this will be zero, since most missions do not take
51       * a fixed number of turns to carry out. Building TileImprovements
52       * and learning in school would be exceptions, however.
53       */
```

É possível identificar a class AbstractMission como Template Pattern pois ela é usada para definir um esqueleto das classes que a estendem, isto é útil porque permite evitar repetição de código e estabelecer uma base para o comportamento de certas classes.

src/net/sf/freecol/server/FreeColServer.java

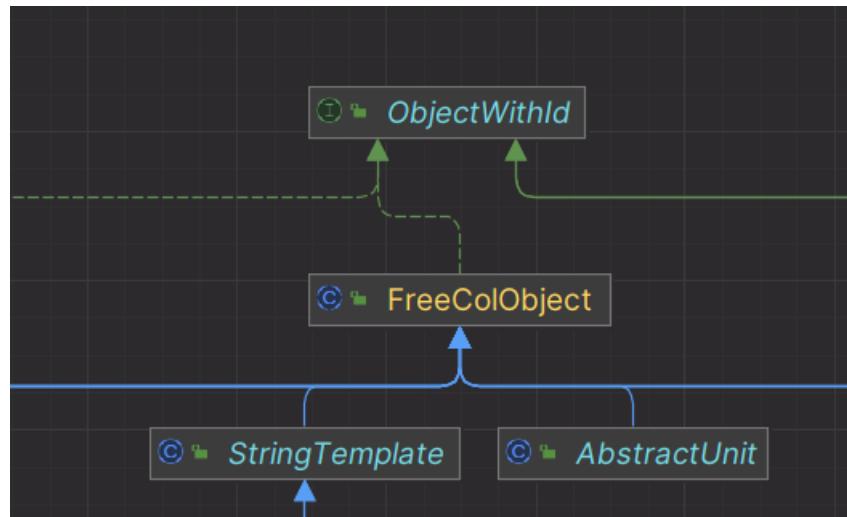
Facade (62438)

```
/*
 * Base constructor common to the following new-game and
 * saved-game constructors.
 *
 * @param name An optional name for the server.
 * @param address The address to use for the public socket.
 * @param port The TCP port to use for the public socket.
 * @exception IOException on failure to open the port.
 */
2 usages + Mike Pope +1
private FreeColServer(String name, InetAddress address, int port) throws IOException {
    this.name = name;
    this.server = createServer(address, port); // Throws IOException
    this.server.start();
    this.userConnectionHandler = new UserConnectionHandler( freeColServer: this);
    this.preGameController = new PreGameController( freeColServer: this);
    this.inGameController = new InGameController( freeColServer: this);
    this.inputHandler = new ServerInputHandler( freeColServer: this);
}
```

Esta classe é responsável por lidar com os pedidos de outras classes a certas classes servindo assim como um intermediário entre a comunicação de classes exteriores a um subsistema e esse subsistema, o que simplifica o uso deste subsistema por classes exteriores ao mesmo, simplificando o código em geral.

src/net/sf/freecol/common/model/FreeColObject.java

Decorator Pattern (62438)



```
149 inheritors ▾ Mike Pope +8
public abstract class FreeColObject
    implements Comparable<FreeColObject>, ObjectWithId {

    protected static final Logger logger = Logger.getLogger(FreeColObject.class.getName());

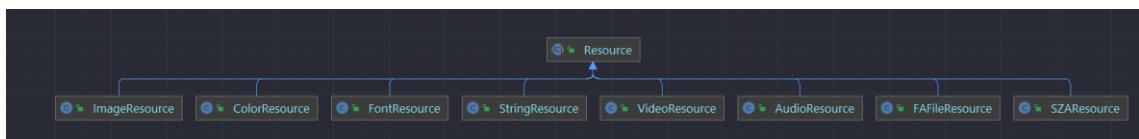
    /** Comparator by FCO identifier. */
    4 usages
    public static final Comparator<? super FreeColObject> fcoComparator
        ▾ Mike Pope
        = new Comparator<FreeColObject>() {
            ▾ Mike Pope
            public int compare(FreeColObject fco1, FreeColObject fco2) {
                return FreeColObject.compareIds(fco1, fco2);
            }
        };
}
```

Nesta imagem estamos presentes de um decorator pattern pois a classe abstrata FreeColObject implementa uma interface e é estendida por duas outras classes que são versões ligeiramente modificadas da classe original dando-lhe propriedades diferentes, este pattern permite-nos obter estes objetos ligeiramente diferentes sem ter de recorrer a uma grande repetição de código mantendo o código mais simples.

Factory Pattern (62756)

```
1  / ...
19 package net.sf.freecol.common.resources;
21
22 import ...
23
24
30 /**
31 * A factory class for creating {@code Resource} instances.
32 * @see Resource
33 */
34 public class ResourceFactory { Grenborgen, 06/05/2008 21:04 · Adding classes for handling resources.
35
36     private static final Logger logger = Logger.getLogger(ResourceFactory.class.getName());
37
38
39     /**
40      * Ensures that only one {@code Resource} is created given the same {@code URI}.
41      */
42     private final Map<URI, Resource> resources = new HashMap<>();
43
44
45     /**
46      * Returns an instance of {@code Resource} with the
47      * given {@code URI} as the parameter.
48      *
49      * @param key The key part of the resource mapping.
50      * @param cachingKey The caching key.
51      * @param uri The {@code URI} used when creating the instance.
52      * @return The <code>Resource</code> if created.
53      */
54     public Resource createResource(String key, String cachingKey, URI uri) {
55         final Resource r = resources.get(uri);
56         if (r != null) {
57             return r;
58         }
59
60         final String pathPart;
61         if (uri.getPath() != null) {
62             pathPart = uri.getPath();
63         } else if (uri.toString().indexOf("/") >= 0) {
64             pathPart = uri.toString().substring(uri.toString().indexOf("/") + 2);
65         } else {
66             pathPart = null;
67         }
68     }

```



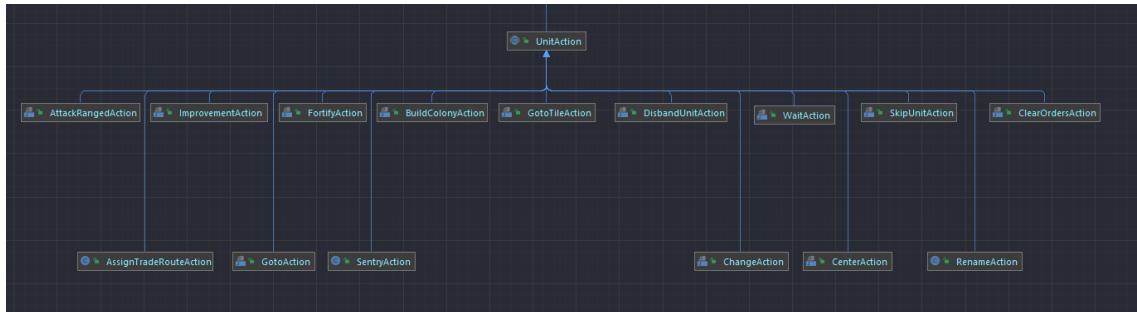
Esta class é utilizada como uma factory class para produzir resources como os que são demonstrados no diagrama acima.

Facade – Widgets (62756)

```
1  /* ... */
19
20  package net.sf.freecol.client.gui;
21
22  import ...
137
138
139  /**
140  * Container for all the higher level dialogs and panels.
141  * Moved here so that Canvas is more manageable.
142  */
143  public final class Widgets {
144
145      private static final Logger logger = Logger.getLogger(Widgets.class.getName());
146
147      /** The game client. */
148      private final FreeColClient freeColClient;
149
150      /** The canvas to write to. */
151      private final Canvas canvas;
152
153
154      /** A wrapper class for non-modal dialogs. */
155      private class DialogCallback<T> implements Runnable {
156
157          /** The dialog to show. */
158          private final FreeColDialog<T> fcd;
159
160          /** A rough position for the dialog. */
161          private final PopupPosition pos;
162
163          /** The handler for the dialog response. */
164          private final DialogHandler<T> handler;    Pope, 14/06/2020 08:45 · Split out simple panel
165
166
167          /**
168           * Constructor.
169           *
170           * @param fcd The parent {@code FreeColDialog}.
171           * @param pos A {@code PopupPosition} for the dialog.
172           * @param handler The {@code DialogHandler} to call when run.
173           */
174          public DialogCallback(FreeColDialog<T> fcd, PopupPosition pos,
175                               DialogHandler<T> handler) {
176              this.fcd = fcd;
```

Widgets age como uma típica facade class, ela é a class que dá o acesso às classes de fora a todos os diálogos de alto nível e panels.

Template (62756)



```
package net.sf.freecol.client.gui.action;

import ...

/**
 * Super class for all actions that should be disabled when no unit is
 * selected.
 */
public abstract class UnitAction extends MapboardAction { Vehrs, 08/05/2010 15:11 · Spring clean a

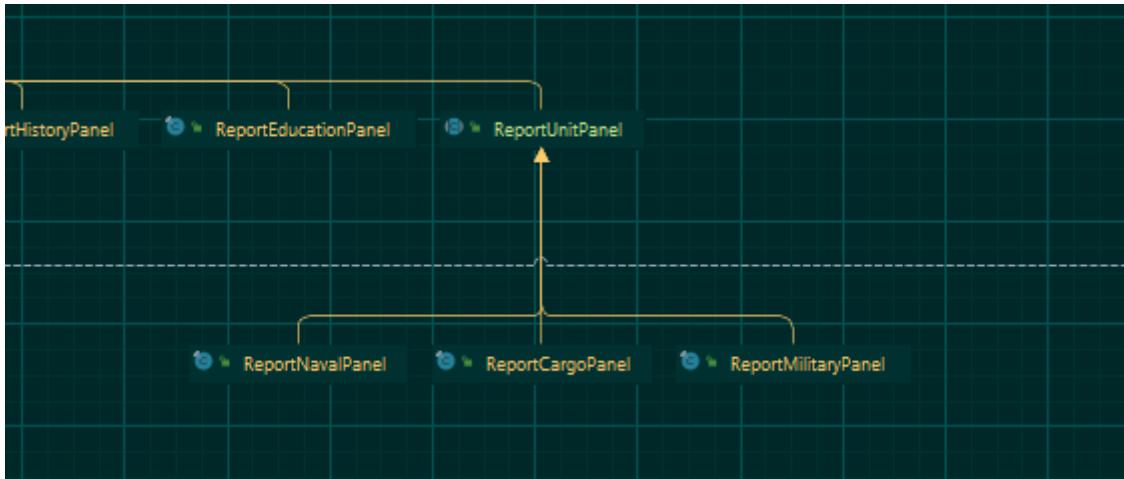
    /**
     * Creates a new {@code UnitAction}.
     *
     * @param freeColClient The {@code FreeColClient} for the game.
     * @param id The object identifier.
     */
    protected UnitAction(FreeColClient freeColClient, String id) { super(freeColClient, id); }

    // Override FreeColAction

    /**
     * {@inheritDoc}
     */
    @Override
    protected boolean shouldBeEnabled() {
        final GUI gui = getGUI();
        Player player;
        Unit unit;
        return gui != null
            && (player = getFreeColClient().getMyPlayer()) != null
            && (unit = gui.getActiveUnit()) != null
            && player.owns(unit)
            && super.shouldBeEnabled();
    }
}
```

Unit Action é a classe de ação abstrata as suas subclasses vão ter implementações mais especializadas e direcionadas no desempenho de uma função.

● Template Pattern (63347)



```

15 /**
16  * Is this unit from the player's units reportable?
17  *
18  * @param unit The {@code Unit} to check.
19  * @return True if the unit should appear in this panel.
20  */
21 protected abstract boolean isReportable(Unit unit);
22
23 /**
24  * Is a unit type and role reportable as a combination usable by
25  * the player?
26  *
27  * @param unitType The {@code UnitType} to check.
28  * @param role The {@code Role} to check.
29  * @return True if the unit type and role should appear in this panel.
30  */
31 protected abstract boolean isReportable(UnitType unitType, Role role);
32
33 /**
34  * Is a REF unit reportable?
35  *
36  * @param au The {@code AbstractUnit} to check.
37  * @return True if REF unit should appear in this panel.
38  */
39 protected abstract boolean isReportableREF(AbstractUnit au);
40
41 /**
42  * Is this unit from the player's units reportable?
43  *
44  * @param unit The {@code Unit} to check.
45  * @return True if the unit should appear in this panel.
46  */
47 protected boolean isReportable(Unit unit) { return unit.isNaval(); }
48
49 /**
50  * Is a unit type and role reportable as a combination usable by
51  * the player?
52  *
53  * @param unitType The {@code UnitType} to check.
54  * @param role The {@code Role} to check.
55  * @return True if the unit type and role should appear in this panel.
56  */
57 protected boolean isReportable(UnitType unitType, Role role) {
58     return unitType.isAvailableTo(getMyPlayer()) && unitType.isNaval()
59         && Specification.DEFAULT_ROLE_ID.equals(role.getId());
60 }
61
62 /**
63  * Is this unit from the player's units reportable?
64  *
65  * @param au The {@code AbstractUnit} to check.
66  * @return True if REF unit should appear in this panel.
67  */
68 protected boolean isReportableREF(AbstractUnit au) { return au.getType(getSpecification()).isNaval(); }
69
70 }

```

A classe ReportUnitPanel é uma classe abstrata em que os seus métodos são utilizadas por outras classes como ReportNavalPanel.

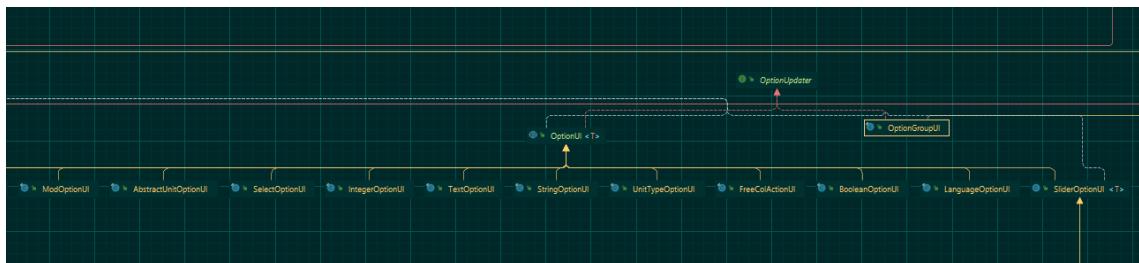
Isso promove a reutilização de código e a personalização de partes do algoritmo. Em resumo, ajuda a criar algoritmos flexíveis e reutilizáveis.

● Command Pattern unload() (63347)

```
|  Pope, 11/7/2011 8:09 AM * Simplify, add comments and group related functionality in client land
5204  public boolean unload(Unit unit) {
5205      if (unit == null || !unit.isCarrier()
5206          || !requireOurTurn()) return false;
5207
5208      boolean ret = true;
5209      Colony colony = unit.getColony();
5210      if (colony != null) { // In colony, unload units and goods.
5211          for (Unit u : unit.getUnitList()) {
5212              if (!leaveShip(u)) ret = false;
5213          }
5214          for (Goods goods : unit.getGoodsList()) {
5215              if (!unloadCargo(goods, dump: false)) ret = false;
5216          }
5217      } else if (unit.isInEurope()) { // In Europe, unload non-boycotted goods
5218          final Player player = getMyPlayer();
5219          for (Goods goods : unit.getCompactGoodsList()) {
5220              if (player.canTrade(goods.getType())) {
5221                  if (!sellGoods(goods)) ret = false;
5222              }
5223          }
5224          if (unit.hasGoodsCargo()) { // Goods left here must be dumped.
5225              getGUI().showDumpCargoDialog(unit,
5226                  (List<Goods> goodsList) -> {
5227                      for (Goods g : goodsList) unloadCargo(g, dump: true);
5228                  });
5229              return false;
5230          }
5231      } else { // Dump goods, units dislike jumping overboard
5232          for (Goods goods : unit.getGoodsList()) {
5233              if (!unloadCargo(goods, dump: false)) ret = false;
5234          }
5235      }
5236  }
```

O método unload() encapsula um comando que pode ser executado para descarregar unidades e bens de uma unidade, característico do padrão Command Pattern.

• Decorator Pattern (63347)



Neste padrão Decorator, são criadas duas classes que implementam a mesma interface que o objeto original a ser estendido. A classe de decorador (OptionUI) envolve o objeto base e adiciona funcionalidades específicas. Isso permite adicionar ou remover funcionalidades de maneira flexível, permite estender objetos existentes de forma dinâmica, adicionando funcionalidades sem modificar a estrutura do objeto original.

- Factory Method, ActionManager() (63350)

```

public void initializeActions(InGameController inGameController,
                             ConnectController connectController) {
    /**
     * Please note: Actions should only be created and not initialized
     * with images etc. The reason being that initialization of actions
     * are needed for the client options ... and the client options
     * should be loaded before images are preloaded (the reason being that
     * mods might change the images).
    */

    /**
     * Possible FIXME: should we put some of these, especially the
     * move and tile improvement actions, into OptionGroups of
     * their own? This would simplify the MapControls slightly.
    */

    // keep this list alphabetized.
    add(new AboutAction(freeColClient));
    add(new AssignTradeRouteAction(freeColClient));
    add(new BuildColonyAction(freeColClient));
    add(new CenterAction(freeColClient));
    add(new ChangeAction(freeColClient));
    add(new ChangeWindowedModeAction(freeColClient));
    add(new ChatAction(freeColClient));
    add(new ClearOrdersAction(freeColClient));
    for (PanelType panelType : PanelType.values()) {
        add(new ColopediaAction(freeColClient, panelType));
    }
    add(new ContinueAction(freeColClient));
    add(new DebugAction(freeColClient));
}

```

```

add(new EndTurnAction(freeColClient));
add(new EuropeAction(freeColClient));
add(new ExecuteGotoOrdersAction(freeColClient));
add(new FindSettlementAction(freeColClient));
add(new FortifyAction(freeColClient));
add(new GotoAction(freeColClient));
add(new GotoTileAction(freeColClient));
add(new LoadAction(freeColClient));
add(new MapControlsAction(freeColClient));
add(new MapEditorAction(freeColClient));
add(new MiniMapToggleViewAction(freeColClient));
add(new MiniMapToggleViewAction(freeColClient, true));
add(new MiniMapToggleFogOfWarAction(freeColClient));
add(new MiniMapToggleFogOfWarAction(freeColClient, true));
add(new MiniMapZoomInAction(freeColClient));
add(new MiniMapZoomInAction(freeColClient, true));
add(new MiniMapZoomOutAction(freeColClient));
add(new MiniMapZoomOutAction(freeColClient, true));
for (Direction d : Direction.values()) {
    add(new MoveAction(freeColClient, d));
    add(new MoveAction(freeColClient, d, true));
}
add(new NewAction(freeColClient));
add(new NewEmptyMapAction(freeColClient));
add(new OpenAction(freeColClient));
add(new PreferencesAction(freeColClient));
add(new SaveAndQuitAction(freeColClient));
add(new QuitAction(freeColClient));
add(new AttackRangedAction(freeColClient));
add(new ReconnectAction(freeColClient));
add(new RenameAction(freeColClient));

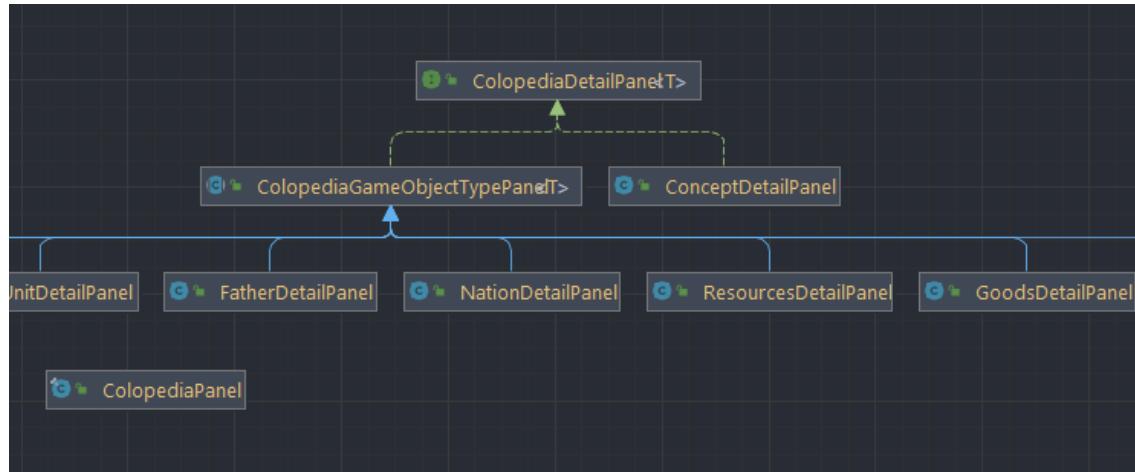
```

Existe um factory method na classe ActionManager, das linhas 73 á 177

O método em questão tem a responsabilidade de criar todas as Actions no próprio, sendo este método chamado em diversos outros métodos

Tal como o comentário sugere, a possibilidade de remover alguns destes Actions e os colocar em grupos próprios (tal como OptionGroups) seria algo que facilitaria a manutenção desse código

- Decorator, ColopediaGameObjectTypePanel() (63350)

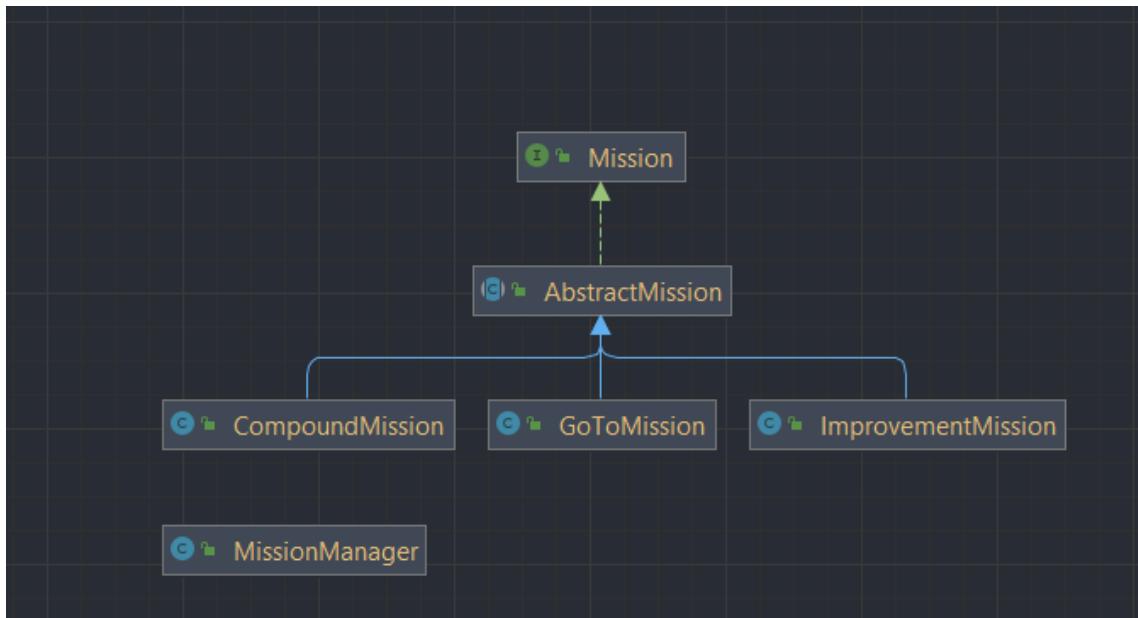


ColopediaGameObjectTypePanel é uma classe abstrata que é utilizado em diversas classes (algumas dessas designadas na imagem, tal como “FatherDeatilPanel”, “GoodsDetailPanel”, etc) e ligada á interface ColopediaDetailPanel, sendo que a classe abstrata é usada nas subclasses para completar o código usado.

A classe em questão, ColopediaGameObjectTypePanel, permite uma composição com outras classes componentes e ao mesmo nível (ConceptDetailPanel) e também permite ser uma superclasse para as subclasses já mencionadas, impedindo assim a repetição de código

Devido a essas características, podemos indicar a classe abstrata como um Decorator

- Template, AbstractMission() (63350)



AbstractMission é uma classe abstrata ligada à interface Mission e que conecta às subclasses “CompoundMission”, “GoToMission” e “Improvement Mission”, sendo uma classe que ajusta métodos usados nas 3 classes

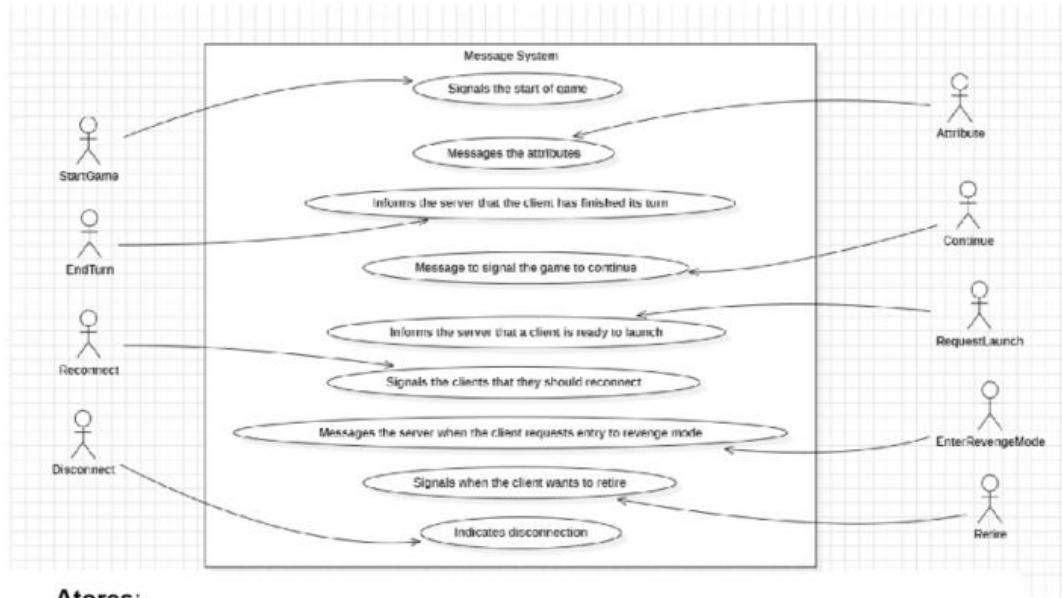
A classe em questão, AbstractMission, permite uma composição com outras classes componentes do tipo Mission e também permite ser uma superclasse para as subclasses já mencionadas, impedindo assim a repetição de código

Assim, AbstractMission é um template

Use Case Diagrams

- Message System (62333)

User Case Diagram



Atores:

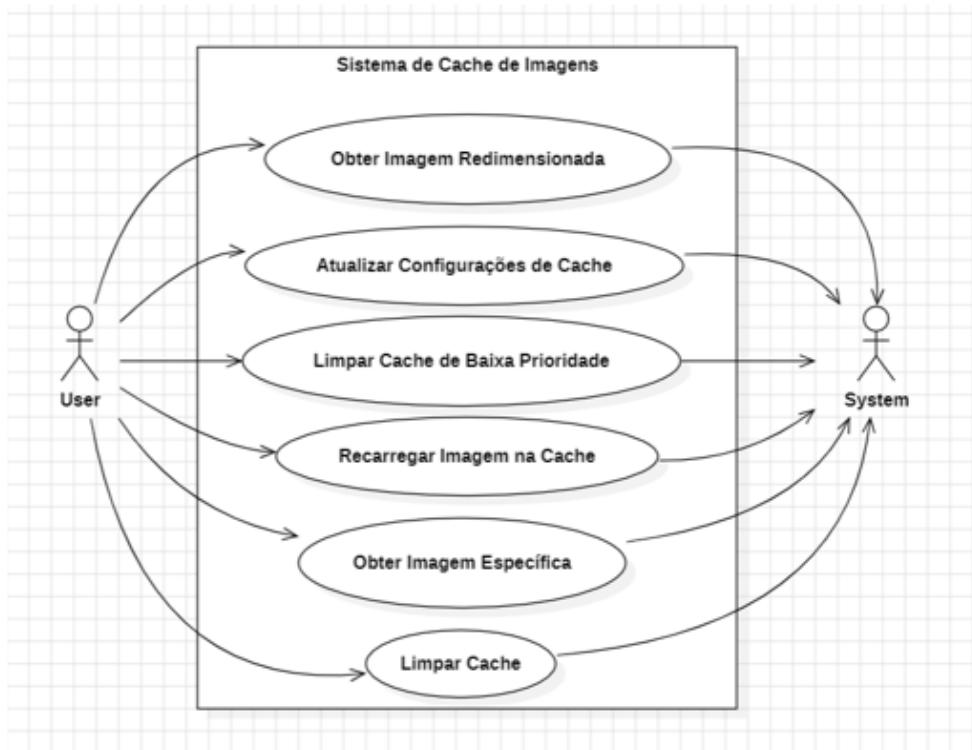
- Continue
- Attribute
- Reconnect
- RequestLaunch
- EndTurn
- Disconnect
- EnterRevengeMode
- StartGame
- Retire

Use Cases:

- Dá continuação ao jogo (Continue)
- Mostra os atributos (Attribute)
- Informa os clientes para se reconectarem (Reconnect)
- Informa que um cliente está pronto para executar (RequestLaunch)
- Informa quando o cliente se quer retirar (Retire)
- Informa o início do jogo (StartGame)
- Informa quando o cliente quer entrar em *revenge mode* (EnterRevengeMode)
- Indica desconexão (Disconnect)
- Informa o server que o client acabou a ronda (EndTurn)

- Cache Image System (62335)

- ImageCache System



Atores:

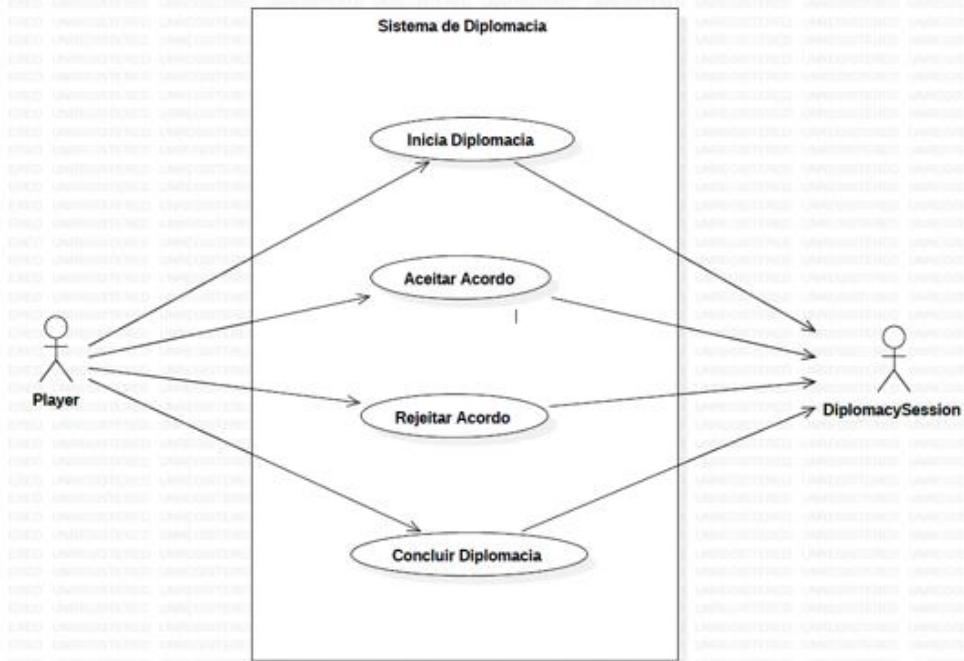
- User
- System

Use Cases:

- É solicitada uma imagem redimensionada ao sistema
- É solicitada uma imagem específica ao sistema
- É solicitada a limpeza da cache de imagens ao sistema
- É solicitada a limpeza da cache de imagens de baixa qualidade ao sistema
- É solicitado o recarregamento de uma imagem específica da cache ao sistema
- É solicitada a atualização de configurações da cache ao sistema

- Diplomacy System (62438)

Use Case Diagrams



Atores:

Player: Utilizador do jogo

DiplomacySession: Objeto dedicado ao controlo das diplomacias do jogo

Use Case:

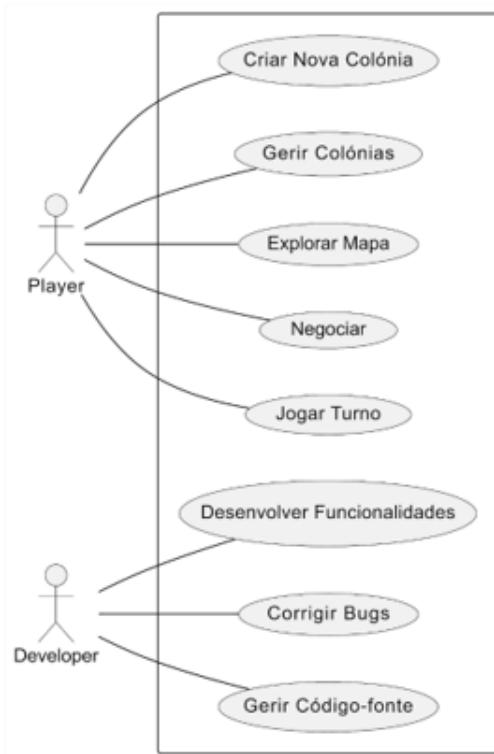
Iniciar Diplomacia: cria uma diplomacia

aceitar acordo: aceita um acordo para uma nova diplomacia

rejeitar acordo: receita o acordo para uma nova diplomacia

concluir diplomacia: termina uma dada diplomacia

- Player Turn System (62756)



Atores:

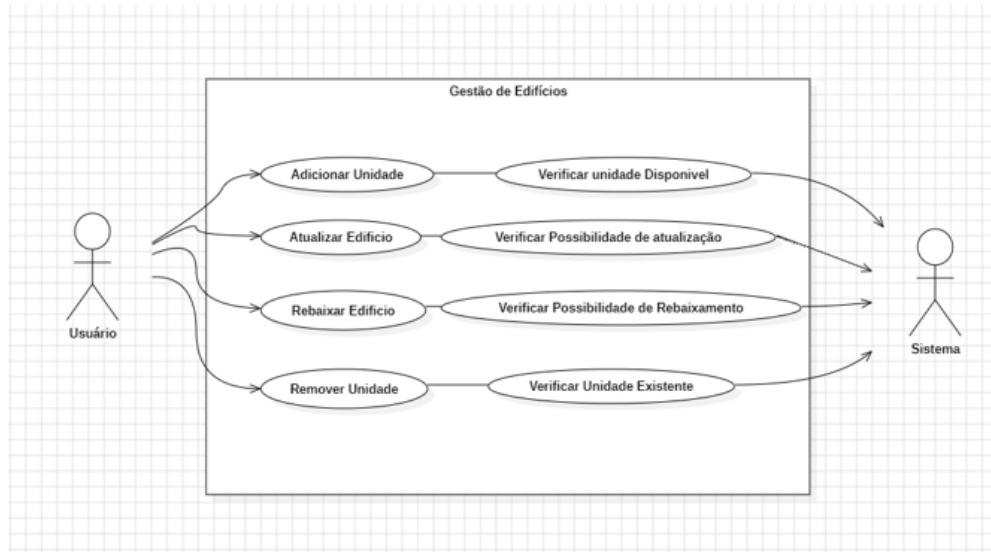
- Player (Usuário do jogo)
- Developer (desenvolvedor do jogo)

Use Cases:

- Criar Nova Colónia: o jogador pede ao jogo para que ele crie uma nova colónia no local desejado.
- Gerir Colónias: o jogador inicia uma funcionalidade que o permite gerir as suas colónias.
- Explorar Mapa: O jogador tem a possibilidade de pedir ao jogo para explorar o mapa
- Negociar: O jogador tem como negociar no jogo
- Jogar Turno: O jogador inicia a sua jogada em dado turno
- Desenvolver Funcionalidades: o developer pode criar novas funcionalidades para o seu jogo e implementá-las
- Corrigir Bugs: o desenvolvedor pode corrigir problemas do código de forma a corrigir erros no funcionamento do jogo.
- Gerir Código-fonte: o desenvolvedor pode gerir o código do jogo melhorando e facilitando a sua leitura.

- Building System (63347)

- Building System



Atores:

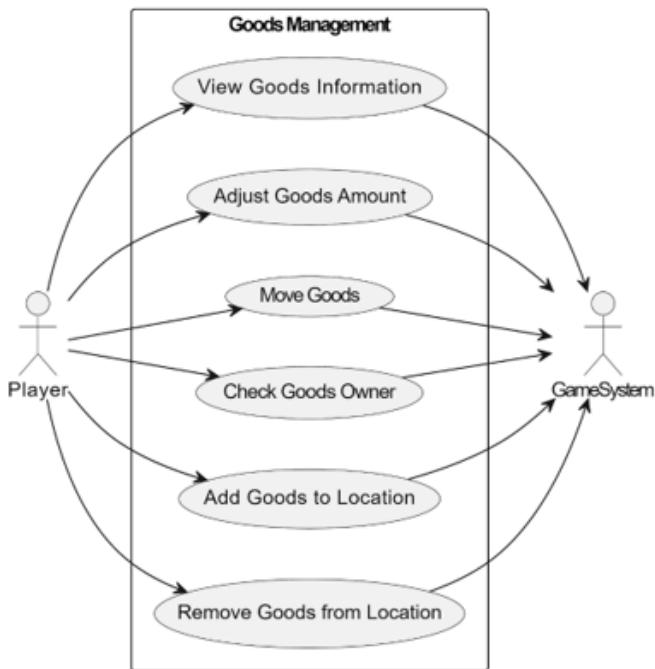
- Usuário
- Sistema

Use Cases:

- Adicionar Unidade: usuário pede ao sistema que adicione uma nova unidade a um edifício. Antes de adicionar a unidade, o sistema verifica se a unidade está disponível.
- Atualizar Edifício: usuário pede ao sistema que atualize as informações de um edifício existente. Antes de realizar a atualização, o sistema verifica a possibilidade de efetuar a atualização.
- Rebaixar Edifício: usuário pede ao sistema que rebaixe um edifício, o que pode envolver a mudança do status ou classificação do edifício. Antes de rebaixar o edifício, o sistema verifica se essa ação é possível.
- Remover Unidade: usuário pede ao sistema que remova uma unidade de um edifício. Antes de realizar a remoção, o sistema verifica se a unidade a ser removida existe.

- Goods Management System (63350)

Use Case Diagram



Atores:

- Usuário (Player)
- Sistema (Game System)

Use Cases:

- View Goods Information - O user pede ao sistema informação sobre um bem específico e o sistema determina esse bem e busca informação do tal
- Adjust Goods Amount - O user pede para usar diferente quantidade de bens e o sistema verifica se é possível aumentar/diminuir para a quantidade desejada
- Move Goods - O user tenta mover os goods e o sistema verifica se é possível executar essa ação (por exemplo, se os goods são do user)
- Check Goods Owner - O user pede informação sobre o dono de um tal bem, fazendo com que o sistema verifique o bem e o dono do tal
- Add Goods To Location - O user tenta colocar bens num certo local e é responsabilidade do sistema verificar se essa ação é possível
- Remove Goods from Location - Semelhante à de cima, o sistema verifica as condições para estabelecer se o pedido do user para remover o bem é possível ou não

User Stories

- Tesouros:

Perspetiva Desenvolvedor:

- Eu como desenvolvedor quero gerar tesouros aleatoriamente no mapa para incentivar os jogadores a explorar o mesmo

Perspetiva Jogador:

- Eu como jogador quero saber como obter tais tesouros e quero que estes sejam valiosos para que valha a pena explorar o oceano com os navios

- Estátuas:

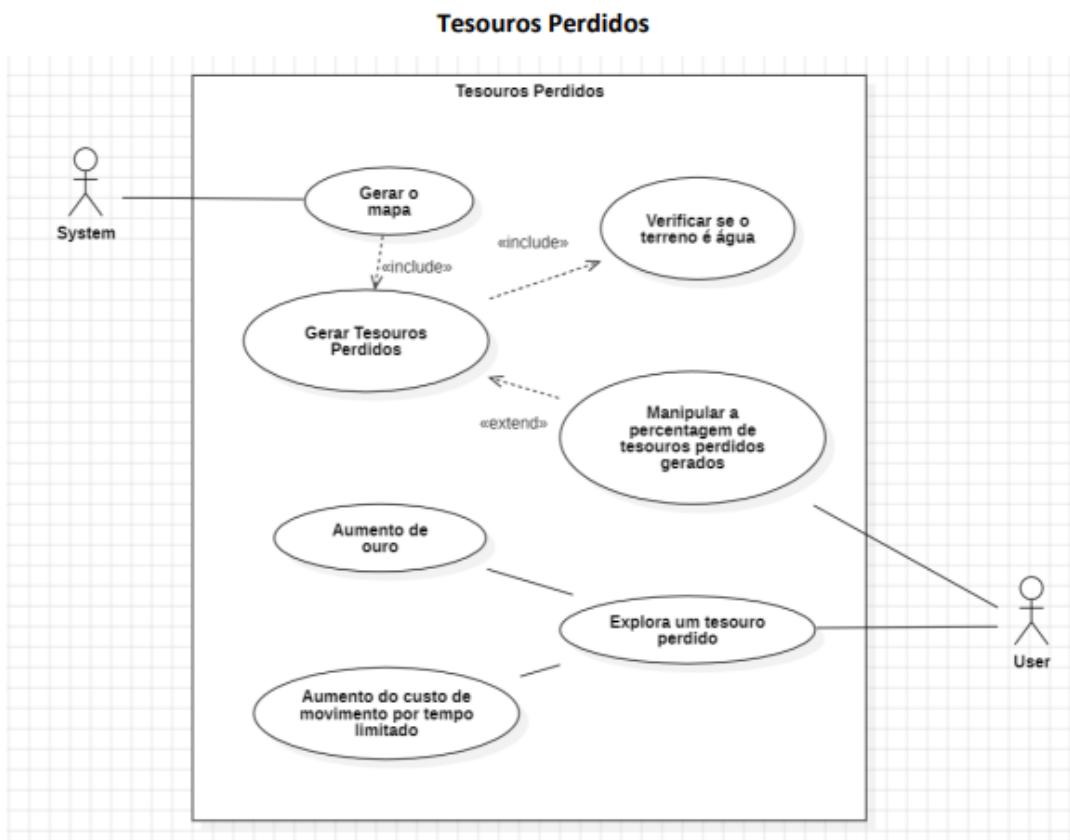
- Eu como jogador quero poder ver uma nova função/utilidades para as estátuas encontradas no mapa para que valha a pena o deslocamento para as mesmas

- Produção:

- Eu como jogador quero que haja novos métodos para aumentar a produção de recursos numa determinada terra

Use Case Diagrams para as User Stories

- Tesouros Perdidos



Atores:

- System
- User

Use cases:

Gerar o mapa: o sistema solicita a geração do mapa.

Gerar tesouros perdidos: na geração do mapa existe uma possibilidade que o mar gerado tenha um recurso de Tesouro perdido.

Verificar se o terreno é água: É verificado se a Tile em questão é do tipo Ocean.

Aumento do custo de movimento por tempo limitado: O sistema aumenta o custo de movimento da embarcação por um número limitado de rondas.

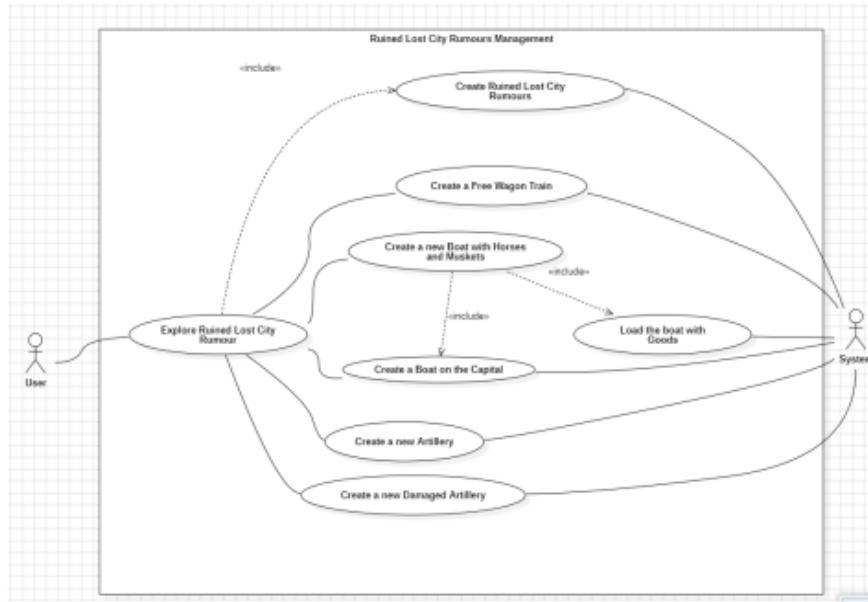
Manipular a percentagem de tesouros perdidos gerados: O utilizador pode alterar a quantidade de tesouros perdidos gerados, manualmente antes da geração do mapa.

Aumento de ouro: O sistema atribui ouro adicional ao utilizador por explorar um tesouro perdido.

Explora um tesouro perdido: O User explora um tesouro perdido gerado pelo mapa.

- Ruined Lost City Rumours (Cidades Arruinadas)

Ruined Lost City Rumours



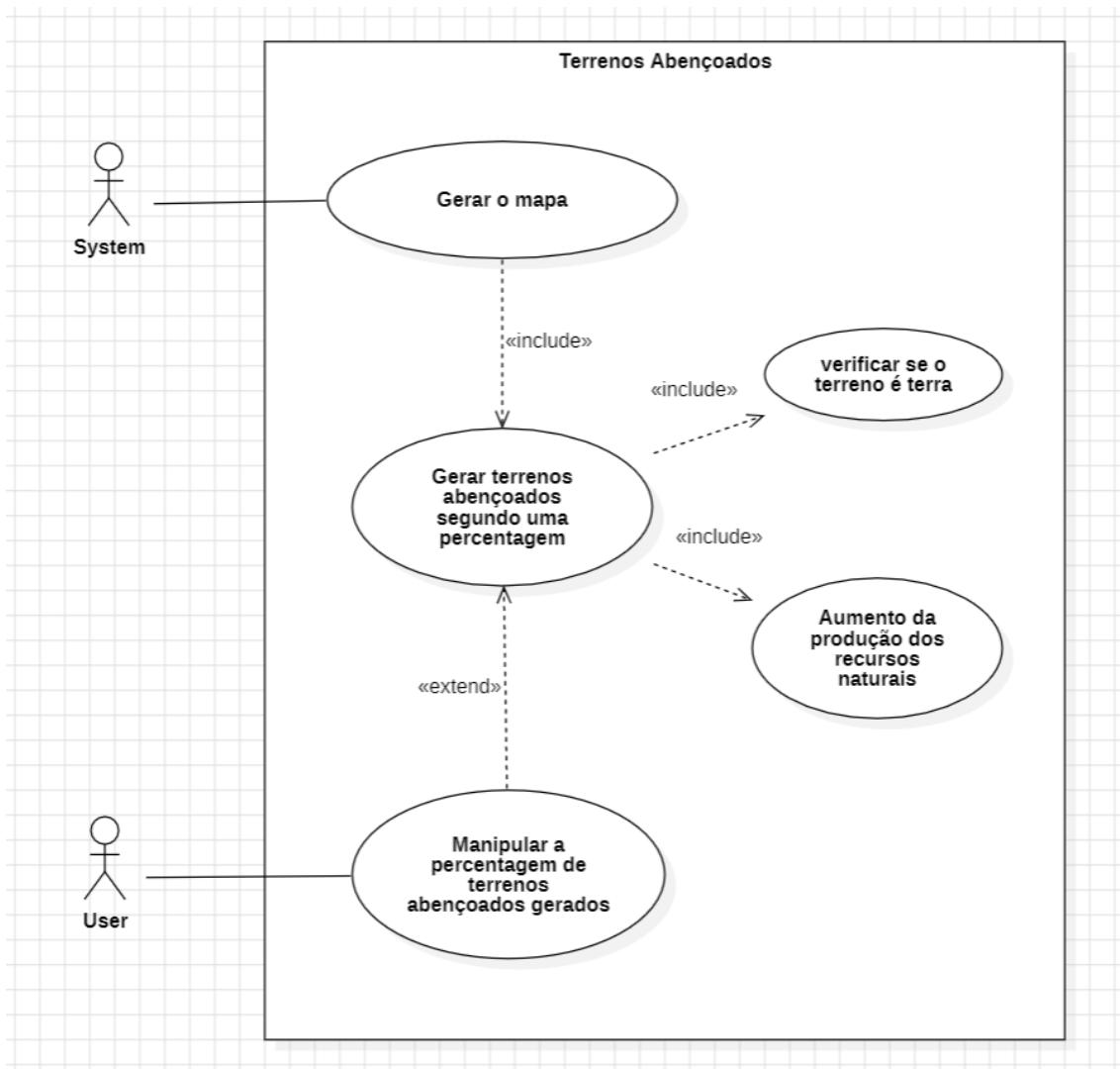
Atores:

- Usuário
- Sistema

Use Cases:

- Explore Ruined Lost City Rumour: Usuário pede ao Sistema para explorar, e baseado em uma probabilidade escolhe o Rumor, isto inclui já ter sido criado um Ruined Lost City Rumour.
- Create Ruined Lost City Rumours: Sistema cria previamente os Rumores, com um tipo aleatório.
- Create a Free Wagon Train: Usuário ao explorar o rumor pede ao sistema para lhe fornecer um novo Wagon Train.
- Create a new Boat with Horses and Muskets: Usuário ao explorar pede ao sistema para lhe fornecer um barco armado, isto inclui o sistema fornecer-lhe o barco e carregar o barco com bens.
- Create a new Boat on the Capital: Sistema cria um barco na capital aleatoriamente, devolvendo-o ao Utilizador.
- Load the boat with Goods: Sistema carrega o barco com os bens ganhos pelo Utilizador.
- Create a new Artillery: Utilizador pede ao Sistema para lhe tentar fornecer uma nova unidade de artilharia
- Create a new Damaged Artillery: Utilizador pede ao Sistema para lhe fornecer uma nova unidade de artilharia danificada.

- Terrenos Abençoados



Vídeo a demonstrar as funcionalidades do jogo:

Link- <https://www.youtube.com/watch?v=FPG7Udt5ym4>

Postmortem e Conclusão:

Após estas 8 semanas de projeto, podemos concluir que um trabalho destes foi uma experiência muito importante para um engenheiro informático. Aprender a trabalhar em um projeto consideravelmente extensivo como o jogo do FreeCol foi realmente algo muito diferente de qualquer outro projeto que alguma vez tenhamos feito.

Como em qualquer outro projeto, nem tudo sempre corre bem. Um dos problemas terá sido a nossa comunicação ao longo projeto que nem sempre foi a melhor (problemas como informação desatualizada, versões diferentes entre nós e no github, entre outros). Foi melhorando ao longo do projeto, no entanto. Um problema específico que aconteceu foi o facto de um dos elementos do grupo ter dado commit dos burndown charts e scrum boards de outros elementos devido ao facto de este ter entendido que o scrum master da semana é que dava commit em vez do scrum master da semana de cada scrum board quando estes foram criados. Ao pedido dos professores, mantivemos os commits.

Mas referindo certos aspectos positivos, acreditamos que as nossas implementações no FreeCol sejam eficientes e, tal como o código do FreeCol, extensíveis. Após diversas horas de resolução de erros, conseguimos criar funcionalidades que passam a todos os testes próprios do FreeCol como também passam a todos os testes customizados especificadamente às nossas implementações.

Passar tempo em equipa a ponderar possibilidades para funcionalidades para o jogo, analisar detalhadamente o código já existente para entender como vários aspectos funcionam, cooperar entre nós para facilitar ao máximo a implementação das nossas funcionalidades e manter a legibilidade e bom funcionamento do código foi certamente trabalhoso mas no fundo, uma experiência necessária e útil (pois a experiência de trabalhar em grupos grandes em código é algo difícil de acontecer em projetos de faculdade mas é algo que certamente será muito mais comum futuramente).