FreeCol -

FreeCol is a turn-based strategy game based on the old game Colonization, and similar to Civilization. The objective of the game is to create an independent nation.

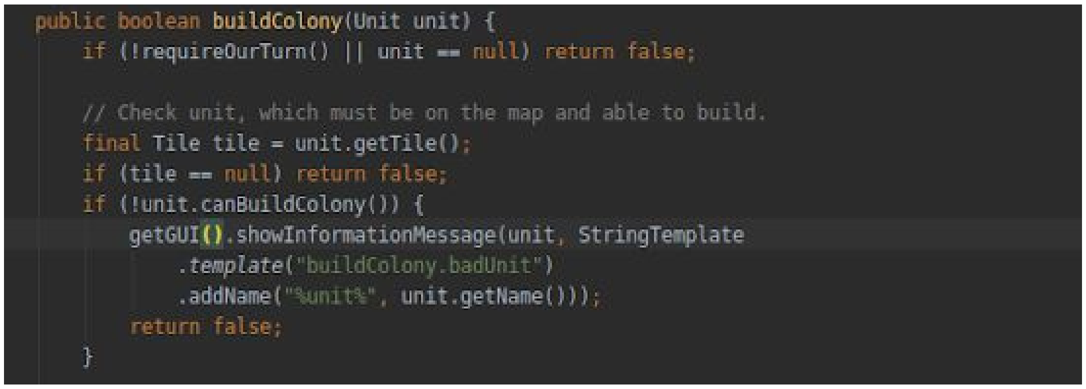
Feature 1: Movement for Ships

To find how the ship moves, we first set out to find what variable holds ship. We started our search under the Unit Class in the Common/Model Folder. We found a general method called “Set Type”. This method returns a general Unit type which led us to believe that ship might be a type of Unit. Within the same Unit class, we found another method called “isNaval”. This method returns true if the unit is naval which seemed to confirm our suspicion that ship could be of the type Unit. We then used the search bar to locate a single ship unit. This led us to method “testCanAdd”, under the UnitTest class, which returned several ship units. In order to find the starting ship, we opened up the game, which revealed the name of the ship to be “merchantman ship”. We used this new information to locate how the ship moved. The search results led us to the “specification.xml” that showed us how the game loads each ship but not how each ship moves. Using the search bar again, we found a “moveDirection” method under the inGameController class. This method gets a unit and moves/binds it towards a direction. It also revealed to us the Tile class. We inspected the Tile class and found that it is the essential component for all pieces, movement, and direction in the game. This led us back to inGameController class to a method called “moveTile”. This method takes in unit and direction and moves that couplet to an occupying tile. This seemed to confirm our suspicion that tile is indeed fundamental to movement. We decided to explore tile movement further which led us to a method called “traverseGoToPath” under the Swing GUI class. This class is a GUI wrapper that controls the tile movement using an “inGameController” (IGC) object. To further explore what the IGC object did we found the method that instantiates the object under the file FreeColClientHolder using the “inGameController” method (which was used under Swing GUI). To further understand movement related to tile we explored the tile parent class “Unit Location.” This parent class has important attributes for tile but proved to be useless for the purposes of movement. From here we wanted to understand how tile and movement is related to mouse/event presses. We were able to locate the “mouseReleased” method under the CanvasMouseListener File. This proved to be fundamental because when the mouse is released it calls the “TraverseGoToPath” method under Swing GUI which instantiates the IGC object. This method tied in the previous fundamental movement methods as it called each method of importance.

Feature 2: Building a Colony

From playing the game, we knew that you could make certain types of units build new colonies. We started by searching for “build colony,” which leads us to the *buildColony()* method in the *InGameController* class. We then went through the method (using IntelliJ’s “Find Usages” feature) to look for other classes called in the method.

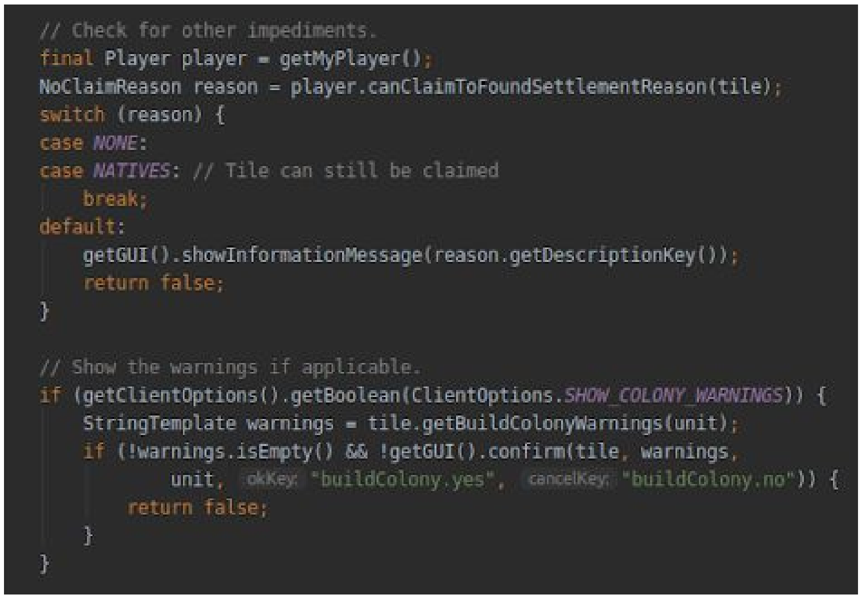
The *buildColony()* method takes in a *Unit* as an argument, which is the unit that is building the colony. *buildColony()* first checks that the unit has the ability to create a colony by calling a method (*canBuildColony()*) looking in the *Unit* and *UnitType* classes.



```java
public boolean buildColony(Unit unit) {
    if (!requireOurTurn() || unit == null) return false;

    // Check unit, which must be on the map and able to build.
    final Tile tile = unit.getTile();
    if (tile == null) return false;
    if (!unit.canBuildColony()) {
        getGUI().showInformationMessage(unit, StringTemplate
            .template("buildColony.badUnit")
            .addName("%unit%", unit.getName()));
        return false;
    }
```

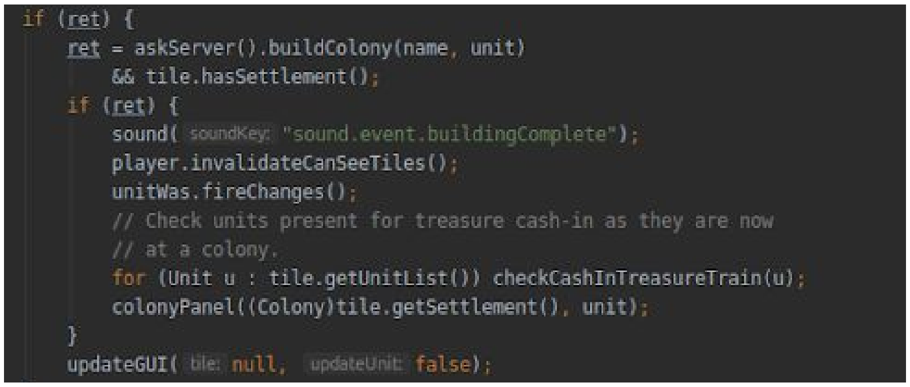
It then checks for other relevant properties and events, such as whether another player is already holding the tile, or if any warnings should be displayed. The relevant classes are *Player, ClientOptions, UnitWas, NameCache*.



```java
// Check for other impediments.
final Player player = getMyPlayer();
NoClaimReason reason = player.canClaimToFoundSettlementReason(tile);
switch (reason) {
case NONE:
case NATIVES: // Tile can still be claimed
    break;
default:
    getGUI().showInformationMessage(reason.getDescriptionKey());
    return false;
}

// Show the warnings if applicable.
if (getClientOptions().getBoolean(ClientOptions.SHOW_COLONY_WARNINGS)) {
    StringTemplate warnings = tile.getBuildColonyWarnings(unit);
    if (!warnings.isEmpty() && !getGUI().confirm(tile, warnings,
            unit, okKey: "buildColony.yes", cancelKey: "buildColony.no")) {
        return false;
    }
}
```

If there are no issues, it sends a message to the server through the *ServerAPI* class to build a colony on that tile.



```java
if (ret) {
    ret = askServer().buildColony(name, unit)
        && tile.hasSettlement();
    if (ret) {
        sound( soundKey: "sound.event.buildingComplete");
        player.invalidateCanSeeTiles();
        unitWas.fireChanges();
        // Check units present for treasure cash-in as they are now
        // at a colony.
        for (Unit u : tile.getUnitList()) checkCashInTreasureTrain(u);
        colonyPanel((Colony)tile.getSettlement(), unit);
    }
    updateGUI( tile: null,  updateUnit: false);
```

Movement for Ships (1-5)

Folder	File	Method	Why?	Priority	Notes
Common/Model	Unit	setType	Trying to find ship type (starting unit)	5	There is ship types
Common/Model	Unit	isNaval	If this is true, we find our ship	2	Too spread out to find something relevant
test/src/Model	UnitTest	testCanAdd	Ship is declared in this test	1	Not enough relevant info extracted
data/rules/classic	Specification.xml	N/A	Has individual attributes for each unit type	3	All unit types are here. Good reference
Control	InGameController	moveDirection	This feels like it controls actual unit movement	5	This indeed controls movement!
Common/Model	tile	N/A	Seems crucial to the movement action	3	Tiles can get pretty complex. Focus on actions for now
Control	IngameController	movesTile	This method does the actual moving inside a space	4	moving into a tile can trigger many other actions
client/gui	SwingGUI	traverseGoToPath	Trying to link GUI components to game logic	3	Links GUI to game movement logic
Client/gui	GUI	GUI	SwingGUI parent class	2	Too general to contain anything useful
Client/control	FreeCol-ClientHolder	igc	GUI parent class	2	Declares inGame-Controller object used by SwingGUI
common/model	UnitLocation	UnitLocation	Tile parent class, might have relevant info	1	doesn't reveal a lot of stuff about tiles
Common/model	CanvasMouseListener	mouseReleased	Might be a link from user input to game logic	5	This class ends up controlling game movement

Movement for Ships (1-5) (1-5)

Folder	File	Method	Relevant?	Relevant how?	Confidence	Notes
Commn/Model	Unit	Set Type	3	trying to find ship. This may lead us to ship unit	3	Need to find if ship is a unit
Common Model	Unit	isNaval	3	checks if unit type is naval	3	did not lead us to ship
Test/src/Model	Unit Test	test can add	3	Merchant type which is a ship	3	found a type of ship
data/rules/classic	specification.xml	N/A	3	Found the Merchant man type	3	Need to find where this is loaded
Control	In game Controller	MoveDirection	3	gets a unit + movs toward a direct	3	ties in to unit/tile direction/movement
common Model	tile	N/A	3	everything is connected to tiles	3	tiles seems essential to Movement
Control	Ingame Controller	MoveTile	4	every unit + direction occupies a tile	3	tile seems fundamental to all movement
client/GUI	Swing GUI	traverseGoToPath	4	This class is a GUI wrapper and controls tile movement using Igc	2	Calls inGameController movement
control	FreeColClientHolder	igc	3	This class instantiates inGameController used by SwingGUI	3	Instantiates inGameController
Common/model	UnitLocation	UnitLocation	2	Parent class for tile	2	Has important attributes for tile
Common/model	CanvasMouseListener	mouseReleased	5	when mouse is released, traverseGoToPath is called	5	Controls movement using mouse events

Build Colony (1-5)

Folder	File	Method	Why?	Priority	Notes
Control	Ingamecontroller	build Colony	Seems pretty straightforward and matches our objective	5	Server api builds colony
Control	FreeColClientHolder	ask server	Is called in inGame Controller	3	Must give us some idea what server does
common/ networking	Server API	build Colony	Is called in the previous method	2	Might regulate API calls
Common/model	Unit	canBuild Colony	Unit is used by build Colony	4	Lots of rules based on unit types
common/ model	UnitType	canBuildColony	Is used by the previous method	4	A wide range of rules and types in this game
Common/ model	Player	—	There is specific colony rules related to player permissions	3	Many colony rules depend on the current player
Client	ClientOptions	—	It is used for what seems to be warnings	3	Could help on discovering a set of rules
common/ model	UnitWas	fireChanges	Is called as a result of a unit action	5	Building colonies might affect the unit state
common/ i18n	Name Cache	putSettlement Names	Is called as a result of building a colony	3	might generate some interesting result

Build Colony

			(1-5)		(1-5)	
Folder	**File**	**Method**	**Relevant?**	**Relevant how?**	**Confidence**	**Notes**
Control	InGameController	buildColony	5	This actually builds the colony	5	References unit, tile, and ServerAP
common/model	Unit	CanBuildColony	4	Dictates colony building rules	4	Units determine colony building behaviour
common/model	UnitType	CanBuildColony	4	Explicit says which unit types are allowed to build	4	Units can have a lot of types
Common/model	Player	—	4	There are colony game rules that change depending on players	5	This is probably one of the m relevant files
Client	ClientOptions	—	2	Has warnings related to rules	2	Some rules can be extracted
Common/model	UnitWas	fireChanges	3	Unit actions such as colony building are described here	4	Not all that is in here is relevant
common/i18n	NameCache	putSettlementName	2	Puts a settlement name back into the pool	4	Only deals with settlement name
Control	FreeColClientHolder	askServer	2	Is called during build-Colony	3	Only returns an object
common/networking	ServerAPI	build Colony	4	Gets server query response for this action	2	API calls could be better documented