Tim Henderson
Steve Johnson
Leigh Praskac

Diplomacloud

The original game of international intrigue taken to the cloud
February 12, 2009
Progress Report Part 1

1. Introduction: Overview

Diplomacloud will provide an online, computer-judged form of the original Diplomacy board game. Unlike similar strategy games such as Risk, Diplomacy is a game determined by skill and inter-player alliances, *not luck*. Seven players compete for the conquest of a map divided into territories. Each player represents one country. During negotiations, players write orders for each of their units and submit them to the board master. When all of the orders have been turned in, the board master reads all of the orders and pieces are moved. Conflicts are resolved using a rule set, which is attached to this report. This order-and-resolution mechanism makes Diplomacy an ideal game for the online database format required for this project. Players will create games, view the current game map, and submit orders online. The program handles conflict resolution and map views automatically.

The basic system is already implemented on a Linux server (masran.case.edu/diplomacy/main.py). The server is running the Apache2 web server to interface with the web. The application is written in the Python 2.5 programming language. The database management system used is MySQL. Python interfaces with MySQL using the MySQLdb API python module. Python interfaces with Apache2 using CGI. Apache is configured to execute python on the specified scripts, rather than using mod_python.

All HTML pages in Diplomacy will be dynamically generated. A templating engine called YAPTU (Yet Another Python Templating Utility) is used to create the HTML templates. Each web page seen by the user will have two parts on the back end: the HTML template, and the python script that interfaces with the database and generates the page based on the template.

Most of the queries run against the database will be in the form of MySQL stored procedures, which allow the programmers to create a logical separation between database interaction and the game logic. However, MySQL's implementation of stored procedures is not entirely adequate for all situations. For instance, stored procedures do not provide an efficient method for inserting large datasets into the database. In these instances, dynamic SQL will be generated by Python scripts and executed on the database.

To protect our database from being compromised, we will employ several measures. The first will be a special user account with very limited permissions just for the application. Next, each web-submitted value that needs to be captured by the database will be validated and escaped to ensure that it does not result in a SQL injection. Using stored procedures also enhances security by having most of our queries parameterized. Although this method is not fool proof, by carefully auditing our procedures we can ensure that they are safe from injection. Finally, our application already has in place an extensive logging capability so we can capture auditing and debugging information in text files. These data will not be captured in the database, so if database fails, the operators will be able to find out why.

2. Application Requirements Specifications

Every game play webpage will include several items: the standard header, footer, and navigation bar in addition to a map showing current unit positions, a season/year counter, and a countdown clock to order submission deadline. The game will be played on a map generated by a random procedural algorithm that we have already written. Players will start the game owning a small number of territories determined by the algorithm. An example map is shown in the attached images. To avoid problems with text label overlap, territory names will be abbreviated to three-letter acronyms. A random name generator will provide a long name and an acronym for each territory.

The game map is arguably the most important part of the game play interface. It shows units as colored shapes representing country (color) and type of unit (army or fleet). The season/year counter displays the current turn of the game. Turns represent six-month periods that rotate through the spring and fall seasons. To be more explicit, games start in the spring of 1901, then progress into the fall of 1901, the spring of 1902, the fall of 1902, and so forth. Orders for each season must be submitted within a given period of time, as kept track of by the countdown clock. Once the clock expires, order resolutions are processed and the season advances.

Each season consists of four stages: 1) movement order writing 2) order resolution 3) retreat/disbandment order writing 4) retreat/disbandment order resolution. Stages 1 and 3 require separate pages. Stages 2 and 4 will be executed internally. Fall seasons include a fifth and sixth stage: 5) gaining/losing units order writing 6) gaining/losing units order resolution. Stage 5 requires a page; stage 6 will be executed internally. Two additional pages display current supply center count for all countries and the final map at the end of the game. In the following paragraphs, the requirements for each page type will be discussed. Example pages of each type are attached to this report.

Games will be started by single users who will add other users to the game. The game creator will choose a map from a paged list of named thumbnails, set a world name, and specify a maximum turn length. When the game creator is ready, the server will randomly assign countries to players and start the game clock.

<u>(Stage 1) Movement Order Page</u>

For each unit of a given country, four types of orders are possible: Hold, Move, Support, and Convoy. Players give orders to each piece individually. If no order is specified, then the unit defaults to Hold. This page therefore needs to provide a way for the player to specify orders for each of their units.

For the player logged in, a list of their units appears next to the current game map. Next to each unit is the specified order for that unit (all units start at Hold). Clicking on the unit name brings an option to change the order. Move orders specify where the unit should move to; Support orders specify which unit to support; and Convoy orders specify which unit to convoy.

When the player is finished specifying orders, the save button is pressed and orders are saved to the database. Players may modify their moves at any time before the countdown clock expires. Once the clock expires, the most recently saved order set is sent for order

resolution. If the player did not save orders before the clock expired, all units default to hold.

(Stage 3) Retreat/disbandment Order Page

When an enemy succeeds in taking over a territory, the defeated unit must vacate the territory. This defeated unit has two options: Retreat or Disband. The default order is to disband. This page, then, provides a way for players to specify Retreat/Disband orders for their defeated units.

Similar to the Stage 3 Page, this page displays all of the player's defeated units next to the current game map. Next to each unit is the specified order for that unit (all units start at Disband). Clicking on the unit name brings up an option to change the order. Retreat orders specify which territory to retreat to. Note that players without defeated units are shown a page displaying the current map only.

Orders are processed in the same way described in the Stage 3 Page section.

(Stage 5) Gaining/Losing Units Order Page

At the end of each Fall turn, supply centers are reassigned. Countries with supply center counts greater than the number of units they currently have on the board are allowed to build new units on their home centers, while countries with supply center counts less than the number of units they currently have on the board must disband units. This page provides a way for players to specify Build and Disband orders.

This page shows either the number of units a player is allowed to build or the number of units a player must disband. If building is the option, the player selects which home center(s) will build what unit type(s). The default is to build no units. If disbanding units is an option, the player selects which units to disband. In the case of no selection, a random unit is selected for disbandment (the unit is selected from among units non-home territories if possible). Note that players not allowed to build or disband are shown a page displaying the current map only.

Orders are processed in the same way described in the Stage 3 Page section.

Supply Center Count Page

This page is common to all players. It consists of a table of all supply centers on the map listed with the current owner of that territory. A second table shows the current supply center count of each country. No user interaction is involved.

End of Game- Map View (and statistics?) Page

When a game is finished, the final map will be displayed on this page. No user interaction is involved.

3. Database Requirement Specification

Entities and Relationships

Most of the model is represented in a partial entity-relationship diagram, attached to this report. We made the diagram using an automated system called GraphViz so that we could simply specify entities and relationships in text without having to draw arrows and rearrange graphical elements. The generated arrangement is not perfect, but it is moderately easy to understand with the accompanying notes.

We have already set up a server to work with and created all of the tables. The MySQL code to create the tables is attached to this document. The code should clear up any ambiguities within the explanations.

Game(gam_id:int, map_id:int,pic: string, season:int, gameyear:year, turn_start:datetime, turn_length:time, turn_stage:int, ended:int)

Game contains an auto-incrementing ID, a map, and its current season and year. *Game* does not relate directly to *User* because each user can only play one country per game, and to relate *Game* to *User* would introduce redundancy. To get a list of users, we can simply look at *Countries*.

The *Game-Uses-Map* relation is implemented with a simple *map_id* foreign key, since a game can have only one map. Since the same map is shown to all players in a game but different forms of the same map are shown to different players in different games, the *Game* entity also has a *pic* attribute, which is the name of the image file that will be shown to the players.

The current state of the game is preserved in the *season* and *gameyear* attributes. The next-turn countdown information is stored in *turn_start* and *turn_length*. The *turn_start* attribute is set to the current time whenever a turn begins, and *turn_length* determines the maximum amount of time a turn can take.

There are multiple stages in a turn, as explained in the previous section. The online interface will display different options based on the current stage. The *turn_stage* attribute keeps track of the current turn stage. It is a foreign key to a *TurnStage* entity, which is a simple integer-string pair.

Map(map_id:int, world_name:string, pic:string)

Map contains basic data about the game world. *World_name* is simply the map's given name so that users can distinguish between maps they have already played. *Pic* is the name of an image file (minus the extension) containing the currently shown version of the map.

The *Map-Has-Country* relation is implemented with a *map_id* foreign key in *Countries*.

Territory(ter_id:int map_id:int, name:string, abbrev:string, piece_pos:int, label_pos:int, supply:int, coastal:int, ter_type:int)

Territories belong to *Map* and have a full name and abbreviation. The attributes *label_x* and *label_y* are used to display the name in the correct position. To display a piece that occupies the territory *piece_x* and *piece_y* are used. The *map_*id foreign key implements the *Map-Has-Territory* relation. The *supply* attribute determines whether the territory is a supply center, and *coastal* is set if the territory is land and borders a sea. The *ter_type* attribute determines whether the territory is land or sea.

Adjacent(ter_id:int, adj_ter_id:int)

Territory adjacencies make up a many-to-many relationship, so an extra table is necessary to store adjacencies. Both attributes are foreign keys to *Territory*, and the primary key of *Adjacent* is the pair of both keys, since no adjacency pair should ever appear in the table twice. Adjacencies will be stored both ways (a-b and b-a) to make queries simpler and faster.

Line(ln_id:int x1:int, y1:int, x2:int, y2:int)

Visual map data must be stored in the database, which means that we have to define a number of primitives. The *Line* entity represents a simple graphical line. Lines can be part of one or more territories, so the *Line-In-Territory* relation is not implemented in the *Line* entity itself.

LineInTerritory(ter_id:int, ln_id:int)

Lines can be part of more than one territory, since a line determines the boundary between two territories. As such, the *LineInTerritory* entity simply has foreign keys to *Territory* and *Line*. The pair of both forms the primary key.

Triangle(tri_id:int, ter_id:int, pos1:int, pos2:int, pos3:int)

Territories are graphically composed of one or more triangles Triangles are always part of exactly one territory, so the *Triangle-In-Land* relation is implemented with the *ter_id* foreign key to *Territory*.

Country(cty_id:int, usr_id:string, name:string, color:string)

Each *country* is controlled by a single user and exists on a single map. Players name their own countries. Colors are specified in 7-byte hex strings, e.g. "#ff0000" for opaque red. The *User-Controls-Country* relation is implemented with a foreign key *usr_id*. The *Map-Has-Country* relation is implemented with the *map_id* foreign key to *Map*.

GamePiece(pce_id:int, cty_id:int, ter_id:int, pce_type:int)

Game pieces belong to individual countries, only one at a time. They can also occupy only one territory at a time. A game piece can be either a fleet or an army, hence the *type* field. The *Country-Has-GamePiece* relation and the *GamePiece-Occupies-Territory* relation are implemented with foreign keys.

Supplier(ter_id:int, cty_id:int)

Territories are assigned supplier status at the map generation stage, but supply centers are owned by different countries in different games depending on how the game plays out. To deal with multiple concurrent games being played on the same map, we need a separate *Supplier* entity to keep track of the current game state. *Supplier* contains foreign keys to *Territory* and *Country*, and the pair forms the primary key. Multiple games reference the same territories, territories can supply one country per game, and multiple games can run at the same time, so neither foreign key alone is sufficient to be a primary key for *Supplier*. However, the pair is sufficient, since a territory can only supply one country at a time in one game.

The foreign keys implement the *Supplier-Represents-Territory* and *Supplier-BelongsTo-Country* relations.

Order(ord_id:int, cty_id:int, pce_id:int, season:enum, gameyear:year, order_type:int, destination:int, executed:int)

Orders are the most important part of the game play, and they are somewhat sophisticated. A simple order operates on a single piece, specified by the foreign key *pce_id*. The *pce_id* attribute implements the *Order-Commands-GamePiece* relation. Orders also have a season and year, allowing for a page showing a summary of past orders. The *order_type* attribute specifies which order is being given, e.g. hold, attack, etc. It is a foreign key to an *OrderType* table that simply stores type-string pairs. Almost every order has exactly one destination with the exception of Hold and Move-Via-Convoy, so there is a *destination* attribute for convenience. The *executed* attribute is a flag to determine whether or not the order was carried out. (If two orders conflict, they will not be carried out, but they will be preserved in the game history.) Orders are related to countries via the *Country-Gives-Order* relationship because all actions in the game are carried out via countries, and we may display order history.

Orders occasionally have arbitrary numbers of arguments, i.e. Move-Via-Convoy. This case is handled by the *Operand* entity.

Operand(opr_id:int, ord_id:int, ter_id:int)

The *Operand* entity allows the program to supply multiple arguments to an order. Operands contain their own ID, the ID of their owner, and the territory argument. The order of the arguments does not matter, so we do not need to insert them in a consistent order. The *ord_id* foreign key implements the *Operand-RefersTo-Order* relation, and the *ter_id* foreign key implements the *Territory-Participates-Operand* relation.

User(usr_id:string, name:string, email:string, screen_name:string, pass_hash:string, salt:string, last_login:datetime, creation:datetime, status:string)

When a person creates an account, they enter their own name, email address, screen name, and password. The system sets *usr_id* by generating a *32*-byte SHA-2 hash based on random seed data provided by /dev/urandom. The password salt is generated the same way from different seed data. The account creation date is stored in *creation*, and *last_login* is updated whenever they log into the system. The *status* attribute holds a status message to display to the user at the bottom of the screen.

Session(session_id:string, sig_id:string, msg_sig:string, usr_id:string, last_update:datetime)

Whenever a user logs in, a *Session* is created. The *session_id* attribute is a SHA-2 string. The *sig_id* and *msg_sig* attributes are session spoofing prevention measures that ensure the validity of the user's session. The *last_update* attribute is updated whenever the user loads a new page in the system.

GameMembership(usr_id:string, gam_id:int, orders_given:int)

Users can play more than one game at once, and their past games are saved in the database. This setup constitutes a many-to-many relationship which requires a separate entity, *GameMembership*. This entity stores foreign keys to *User* and *Game*. In addition to representing the relationship, the *GameMembership* entity keeps track of whether players have clicked the "Issue Orders" button indicating that they are finished choosing orders. If the *orders_given* attribute is set for all players and the game click has not run out, then the turn will roll over anyway in order to avoid wasting time.

Message(msg_id:int, from_usr:string, to_usr:string, time_sent:datetime, subject:string, msg:string, read:int)

Since discussion is an integral part of Diplomacy, and therefore Diplomacloud, the ability to communicate is essential. The *Message* entity is the main entity in the messaging system. Each message has an ID, attributes for the sender and recipient, the time sent, a subject line, a message body, and a flag to determine if the message has been read by the recipient. We do not plan to implement messages with multiple recipients; instead, we will create copies for each individual.

Queries

Generating the map:

- Create new instances of *Line, Triangle, Territory, Adjacent, LineInTerritory,* and *Map* based on the data generated by the map generator program

Drawing the map:

- Select all lines associated with the chosen map
- Select all triangles associated with the map
- Select all territories associated with the map
- Join territories with countries to obtain their colors
- Update the *Game.pic* attribute to point to the current location of the map

These data will be used to draw and color the triangles and then inform the database of where the new image is saved.

Starting the game:

- Get a list of all users
- Get a list of users that are members of the current game
- Get the *pic* attribute of the chosen map
- Update the *Game.map_id* attribute
- Add GameMembership entries
- Create new instances of *Country*, *GamePiece*, and *Supplier*
- Update the *Game.turn_start, turn_length, turn_stage, season,* and *year* attributes

Turn stage 1:

- Select all possible Stage 1 orders from the order_types table
- Select all sea territories associated with the current map
- Select all territories adjacent to a specified territory
- Create new instances of *Order* and *Operands*

Turn stage 2:

- Select all orders in this turn
- Select all orders in this turn with the same destination
- Update *Order.executed* attributes for orders that did not conflict
- Update instances of *GamePiece* referenced by orders with *executed* attributes set
- Select all supply centers that belong to each player one at a time

Turn stage 3:

- Select all territories adjacent to a specified territory
- Select a specified *GamePiece*

Turn stage 4:

- Select all orders in this turn
- Update instances of *GamePiece* referenced by orders

Turn stage 5:

- Select all supply centers belonging to a country
- Remove and create instances of *GamePiece* based on orders

End game/old game view:

- Select *Game* to display statistics
- Select all supply centers belonging to a country

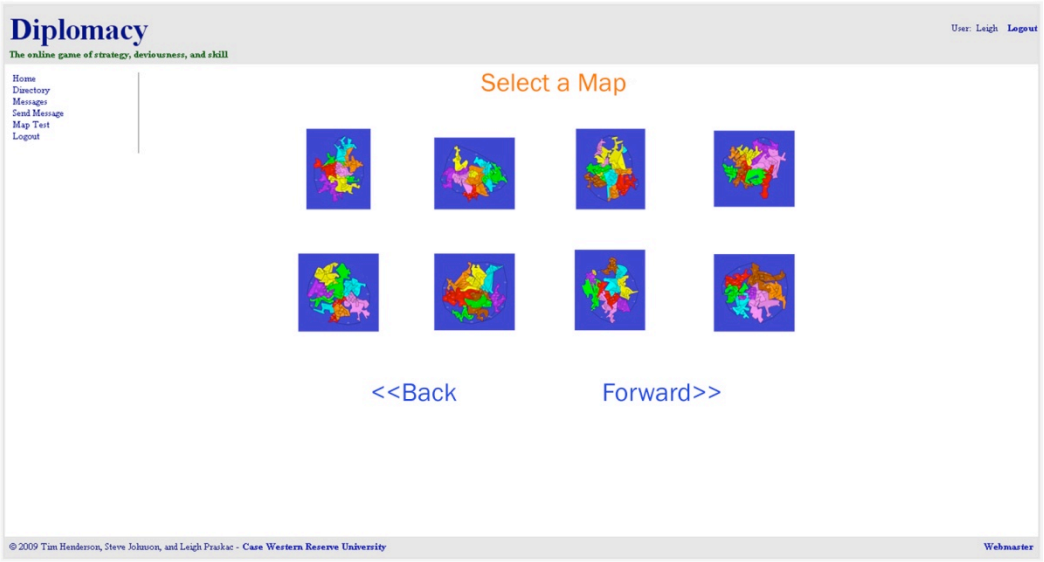
User activities:

- Create, update, and delete instances of *Session*
- Log out of a *Session*
- Create a new instance of *User*
- Update the *last_login* attribute of a *User* instance
- Select all messages with a specific *User* as the recipient

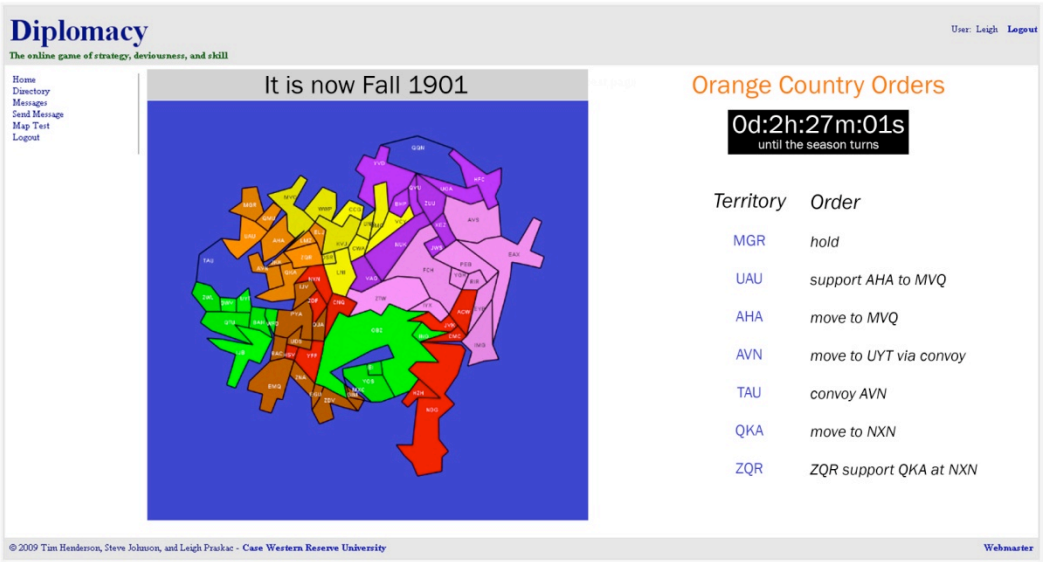
Message activities:

- Create and delete instances of *Message*

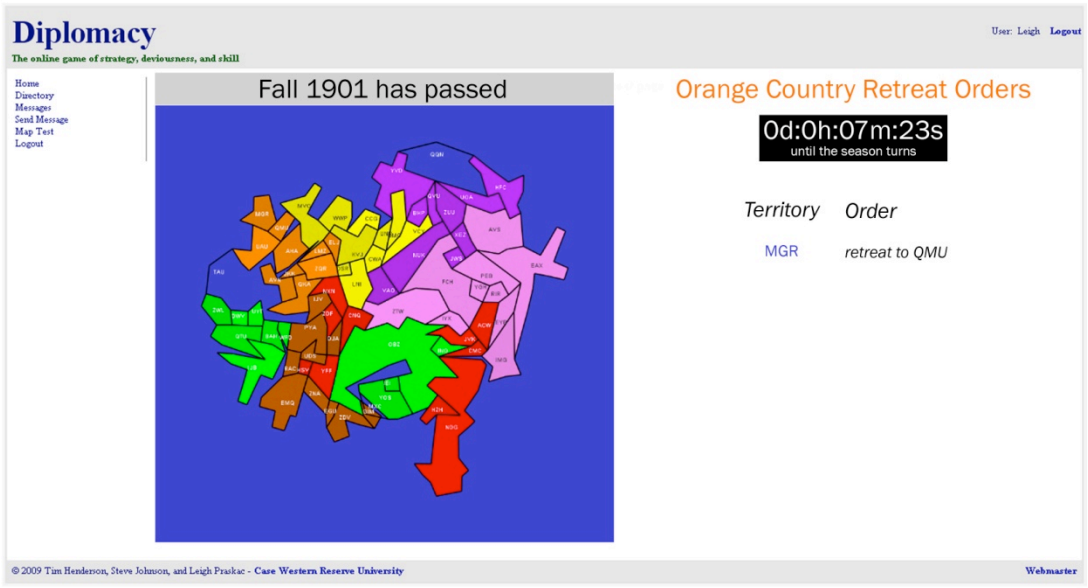
Mock
Game Start: Select Map



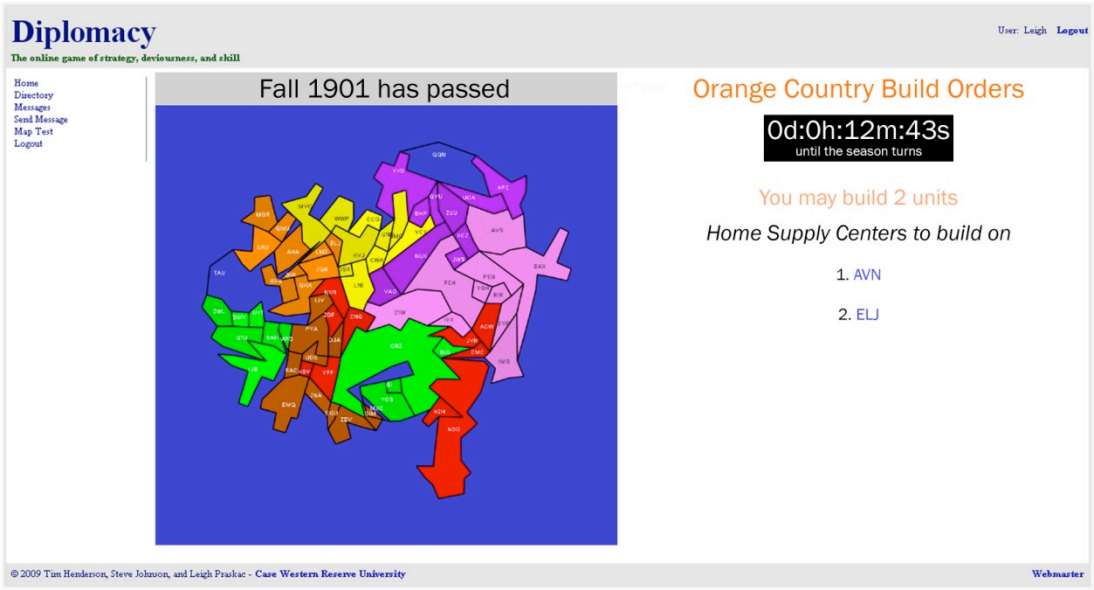
Mock
(Stage 1) Movement Order Page



Mock
(Stage 3) Retreat/Disbandment Order Page



Mock
(Stage 5) Gaining/Losing Order Page



Mock
Supply Center Count Page

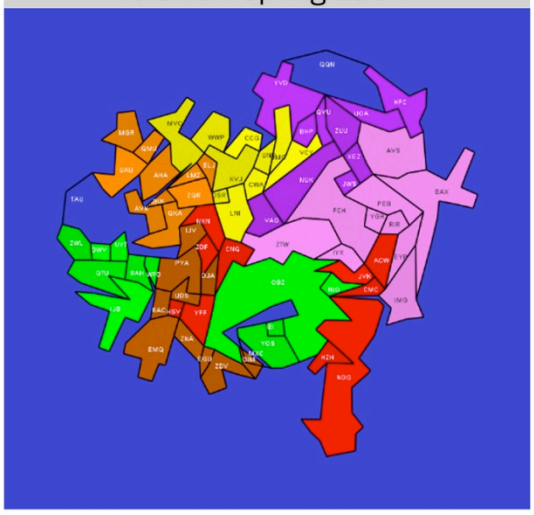
Diplomacy

The online game of strategy, deviousness, and skill

User: Leigh **Logout**

Home
Directory
Messages
Send Message
Map Test
Logout

It is now Spring 1904



Current Rankings

Rank	*Player*	*Country*	*Supply Center Count*
1	Joe	Zland	12
2	Freddy	United Jive	8
3	Elisa	Sumatrail	7
4	Melanie	Milky Way	6
5	James	J Kingdom	4
6	Fisk	Mexicate	2
7	Tootsie	Silva	2

Supply Center Ownership

Supply Center	Owner
YVD	Zland
ZUU	Zland
AVS	Tootsie
JVK	-
OBZ	Fisk
...	...

Webmaster

4. ER Data Model Design

Attributes

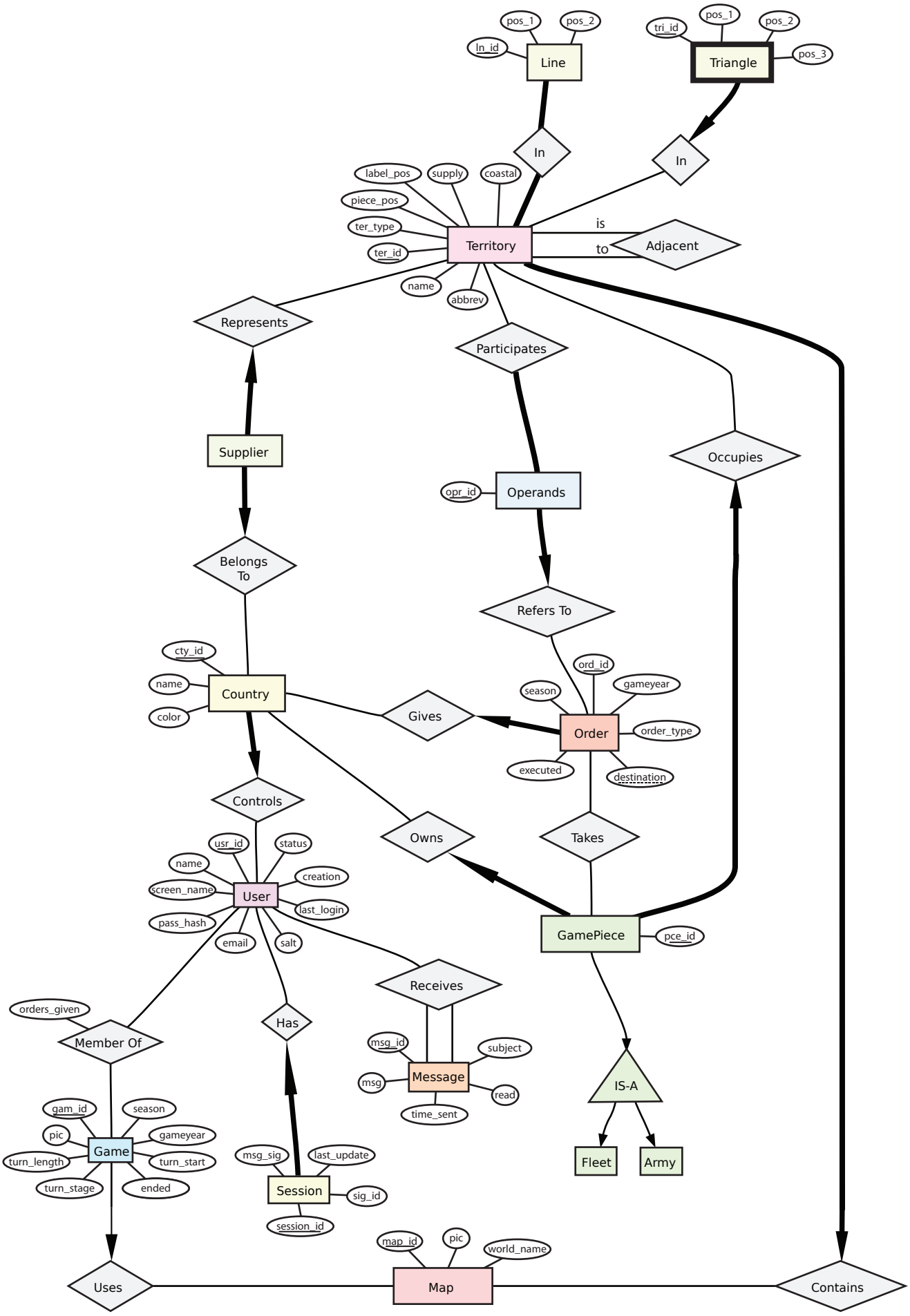
The following table lists all of the entities that have attributes and specifies properties for each attribute. All attributes are single-valued and not derived. The only weak entity in the database is Triangle. Several attributes are composite: piece_pos and label_pos in Territory, and the ‘datetime’ type attributes. In the relational model and actual implementation, piece_pos is split into piece_x and piece_y and label_pos is split into label_x and label_y. It should be noted that MySQL handles reading of the multiple integers stored in ‘datetime’ types. Relationships are implemented either through a relationship entity, such as LineInTerritory, or by putting foreign keys directly in entities such as ter_id in Triangle. The latter type relationship implementations are indicated by the ‘relationship fulfilled’ column in the table below.

Attributes Table (part 1)

Entity or Relationship	Attribute	Data Type	Primary Key Member	Foreign key from	Relationship fulfilled	Default value	not null
Line	lin_id	int	Y				
	pos_1	int					Y
	pos_2	int					Y
Ter_ln_relation	ter_id	int	Y	Terriotry			Y
	ln_id	int	Y	Line			Y
Triangle	tri_id	int	Y				
	pos_1	int					Y
	pos_2	int					Y
	pos_3	int					Y
	ter_id	int	Y	Territory			Y
Territory	ter_id	int	Y				
	name	varchar				null	Y
	abbrev	varchar				null	
	map_id	int		Map	occupies		Y
	ter_type	enum					Y
	piece_pos	int					Y
	label_pos	int					Y
	supply	tinyint		Country:cty_id			Y
	coastal	tinyint					Y
Adjacent	ter_id	int	Y	Territory			Y
	adj_ter_id	int	Y	Territory:ter_id			Y
Operands	opr_id	int	Y				
	ter_id	int		Territory	participates		Y
	ord_id	int		Order	refers to		Y
Supplier	ter_id	int	Y	Territory	represents		Y
	cty_id	int	Y	Country	belongs		
Country	cty_id	int	Y				
	usr_id	varchar		User	controls	null	
	name	varchar				null	
	color	varchar				#ffffff'	
Order	ord_id	int	Y				
	cty_id	int		Country	gives		Y
	pce_id	int		Piece	takes		Y
	gameyear	year					Y
	season	enum					Y
	order_type	int					Y
	destination	int				null	
	executed	int				0	

Attributes Table (part 2)

Entity or Relationship	Attribute	Data Type	Primary Key Member	Foreign key from	Relationship fulfilled	Default value	not null
User	usr_id	string	Y				y
	name	string					y
	screen_name	string					y
	pass_hash	string					y
	email	string					y
	salt	string					y
	last_login	datetime					y
	creation	datetime					y
	status	string				null	
Piece	pce_id	int	Y				
	cty_id	int		Country	owns		Y
	ter_id	int		Territory	occupies		Y
	pce_type	enum			IS-A	null	
Game_Membership	orders_given	int				0	
	usr_id	string	Y	User			Y
	gam_id	int	Y	Game			Y
Message	msg_id	int	Y				y
	from_usr	string		User			y
	to_usr	string		User			y
	msg	string					y
	time_sent	datetime					y
	have_read	int				0	
	subject	string					y
Game	gam_id	int	Y			null	
	map_id	int		Map	uses	null	
	gameyear	year				1999	
	season	enum				fall	
	turn_start	datetime				null	
	pic	string				null	
	turn_length	time				24:00:00	
	turn_stage	int				0	
	ended	int				0	
Session	session_id	string	Y				y
	msg_sig	string					y
	last_update	datetime					y
	sig_id	string					y
	user_id	string		User	has		y
Map	map_id	int	Y				
	pic	string				null	
	world_name	string					y


pos_1
pos_2
ln_id
Line
tri_id
pos_1
pos_2
Triangle
pos_3
In
In
label_pos
supply
coastal
piece_pos
ter_type
ter_id
Territory
is
to
Adjacent
name
abbrev
Represents
Participates
Occupies
Supplier
opr_id
Operands
Belongs To
Refers To
cty_id
name
color
Country
Gives
ord_id
season
gameyear
Order
order_type
executed
destination
Controls
Owns
Takes
usr_id
status
name
creation
screen_name
User
last_login
pass_hash
email
salt
GamePiece
pce_id
Receives
orders_given
Member Of
Has
msg_id
subject
msg
Message
read
time_sent
IS-A
gam_id
season
pic
gameyear
turn_length
Game
turn_start
turn_stage
ended
msg_sig
last_update
Session
sig_id
session_id
Fleet
Army
map_id
pic
world_name
Uses
Map
Contains

5. Integrity Constraints for the ER Model

Enforcement

Constraints will be enforced in several ways in Diplomacloud. Primary key constraints will be enforced by MySQL by specifying them when the table is created. Referential integrity constraints will be enforced by a combination of methods. All foreign keys will be specified by the table create commands. However, because there are some subtleties about when things can and cannot be deleted in our database, they will all be specified as ON DELETE RESTRICT. To enforce weak entities being deleted we will make use of stored procedures to do the deletion. The application will not be able to directly delete rows from tables, it will only be able to delete through stored procedures. In this way the stored procedures will be written carefully so as to maintain consistency in our database. This approach appeals to us because the application programmer will know exactly what is happening in the database at all times. Instead of rows automatically disappearing due to an on delete cascade, they will disappear as specified by the stored procedure. Each integrity constraint is listed in the create tables section of this document. Additionally, foreign key and primary key constraints are listed in the attributes table.

```
-- Tim Henderson
-- Table Creation for diplomacy

START TRANSACTION;
--DROP DATABASE IF EXISTS diplomacy;
--CREATE DATABASE diplomacy DEFAULT CHARACTER SET ascii COLLATE ascii_general_ci;
USE diplomacy;
---------------------------------------  Schema  ------------------------------------
--  users (usr_id : varchar(64), name : varchar(256), email : varchar(256),
--         screen_name : varchar(128), pass_hash : varchar(64), salt : varchar(64),
--         last_login : datetime, creation : datetime, status : varchar(500))
--
--  session (session_id : varchar(64), sig_id : varchar(64), msg_sig : varchar(64),
--              usr_id : varchar(64), last_update : datetime)
--
--  message (msg_id : int(11), from_usr : varchar(64), to_usr : varchar(64),
--           time_sent : datetime, subject : varchar(256), msg : varchar(10000)
--           read : tinyint(1))
--
--  map (map_id : int(11), world_name : varchar(128), pic : varchar(64), keep : tinyint(1))
--
--  game (gam_id : int(11), map_id : int(11), pic : varchar(64),
--        gam_season : enum('spring', 'fall'),  gam_year : year(4), turn_start : datetime,
--        turn_length : time, turn_stage : int(11),  ended : tinyint(1))
--
--  game_membership (usr_id : varchar(64), gam_id : int(11), orders_given : tinyint(1))
--
--  turn_stages (trs_id : int(11), name : varchar(64), description : varchar(256),
--               fall : tinyint(1))
--
--  country (cty_id : int(11), usr_id : varchar(64), name : varchar(128), color : varchar(7))
--
--  territory (ter_id : int(11), map_id : int(11), name : varchar(128), abbrev : varchar(4),
--             piece_x : int(11), piece_y : int(11), label_x : int(11), label_y : int(11),
--             ter_type : enum('land', 'sea'), supply : tinyint(1), coastal : tinyint(1))
--
--  adjacent (ter_id : int(11), adj_ter_id : int(11))
--
--  supplier (ter_id : int(11), cty_id : int(11))
--
--  triangle (tri_id : int(11), ter_id : int(11), x1 : int(11), y1 : int(11), x2 : int(11),
--            y2 : int(11), x3 : int(11), y3 : int(11))
--
--  line (ln_id : int(11), x1 : int(11), y1 : int(11), x2 : int(11), y2 : int(11))
--
--  ter_ln_relation (ter_id : int(11), ln_id : int(11))
--
--  piece (pce_id : int(11), cty_id : int(11), ter_id : int(11),
--         pce_type : enum('fleet', 'army'))
--
--  order_type (odt_id : int(11), order_text : varchar(128))
--
--  orders (ord_id : int(11), cty_id : int(11), pce_id : int(11),
--          season : enum('spring', 'fall'), year : year(4), order_type : int(11),
--          destination : int(11), executed : tinyint(1))
--
--  operands (opr_id : int(11), ord_id : int(11), ter_id : int(11))
---------------------------------------  Schema  ------------------------------------

DROP TABLE IF EXISTS users;
CREATE TABLE users
(
    usr_id varchar(64) NOT NULL,
    name varchar(256) NOT NULL,
    email varchar(256) NOT NULL,
    screen_name varchar(128) NOT NULL,
    pass_hash varchar(64) NOT NULL,
    salt varchar(64) NOT NULL,
    last_login datetime NOT NULL,
    creation datetime NOT NULL,
    status varchar(500),
```

```
    CONSTRAINT pk_users PRIMARY KEY (usr_id),
    CONSTRAINT uq_email UNIQUE (email),
    CONSTRAINT uq_screen_name UNIQUE (screen_name)
);

DROP TABLE IF EXISTS sessions;
CREATE TABLE sessions
(
    session_id varchar(64) NOT NULL,
    sig_id varchar(64) NOT NULL,
    msg_sig varchar(64) NOT NULL,
    usr_id varchar(64) NOT NULL,
    last_update datetime NOT NULL,
    CONSTRAINT pk_session PRIMARY KEY (session_id),
    CONSTRAINT fk_usr_id FOREIGN KEY (usr_id)
        REFERENCES users(usr_id) ON DELETE RESTRICT
);

DROP TABLE IF EXISTS message;
CREATE TABLE message
(
    msg_id int(11) AUTO_INCREMENT,
    from_usr varchar(64) NOT NULL,
    to_usr varchar(64) NOT NULL,
    time_sent datetime NOT NULL,
    subject varchar(256) NOT NULL,
    msg varchar(10000) NOT NULL,
    have_read tinyint(1) DEFAULT 0,
    CONSTRAINT pk_message PRIMARY KEY (msg_id),
    CONSTRAINT fk_from_usr FOREIGN KEY (from_usr)
        REFERENCES users(usr_id) ON DELETE RESTRICT,
    CONSTRAINT fk_to_usr FOREIGN KEY (to_usr)
        REFERENCES users(usr_id) ON DELETE RESTRICT
);

DROP TABLE IF EXISTS map;
CREATE TABLE map
(
    map_id int(11) AUTO_INCREMENT,
    world_name varchar(128),
    pic varchar(64) NOT NULL,
    CONSTRAINT pk_map PRIMARY KEY (map_id)
);

DROP TABLE IF EXISTS turn_stages;
CREATE TABLE turn_stages
(
    trs_id int(11) AUTO_INCREMENT,
    name varchar(64),
    description varchar(256),
    fall tinyint(1) DEFAULT 0,
    CONSTRAINT pk_turn_stages PRIMARY KEY (trs_id)
);

DROP TABLE IF EXISTS game;
CREATE TABLE game
(
    gam_id int(11) AUTO_INCREMENT,
    map_id int(11),
    pic varchar(64),
    gam_season enum('spring', 'fall') DEFAULT 'fall',
    gam_year year(4) DEFAULT 1999,
    turn_start datetime NULL,
    turn_length time DEFAULT '24:00:00',
    turn_stage int(11) DEFAULT 0,
    ended tinyint(1) DEFAULT 0,
    CONSTRAINT pk_game PRIMARY KEY (gam_id),
    CONSTRAINT fk_map_id FOREIGN KEY (map_id)
        REFERENCES map(map_id) ON DELETE RESTRICT,
    CONSTRAINT fk_turn_stage FOREIGN KEY (turn_stage)
        REFERENCES turn_stages(trs_id) ON DELETE RESTRICT
);
```

```
DROP TABLE IF EXISTS game_membership;
CREATE TABLE game_membership
(
    usr_id varchar(64) NOT NULL,
    gam_id int(11) NOT NULL,
    orders_given tinyint(1) DEFAULT 0,
    CONSTRAINT pk_game_membership PRIMARY KEY (usr_id, gam_id),
    CONSTRAINT fk_usr_id FOREIGN KEY (usr_id)
        REFERENCES users(usr_id) ON DELETE RESTRICT,
    CONSTRAINT fk_gam_id FOREIGN KEY (gam_id)
        REFERENCES game(gam_id) ON DELETE RESTRICT
);

DROP TABLE IF EXISTS country;
CREATE TABLE country
(
    cty_id int(11) AUTO_INCREMENT,
    usr_id varchar(64),
    name varchar(128),
    color varchar(7) DEFAULT '#ffffff',
    CONSTRAINT pk_country PRIMARY KEY (cty_id),
    CONSTRAINT fk_usr_id FOREIGN KEY (usr_id)
        REFERENCES users(usr_id) ON DELETE RESTRICT
);

DROP TABLE IF EXISTS territory;
CREATE TABLE territory
(
    ter_id int(11) AUTO_INCREMENT,
    map_id int(11) NOT NULL,
    name varchar(128),
    abbrev varchar(4),
    piece_x int(11) NOT NULL,
    piece_y int(11) NOT NULL,
    label_x int(11) NOT NULL,
    label_y int(11) NOT NULL,
    ter_type enum('land', 'sea') NOT NULL,
    supply tinyint(1) NOT NULL,
    coastal tinyint(1) NOT NULL,
    CONSTRAINT pk_territory PRIMARY KEY (ter_id),
    CONSTRAINT fk_map_id FOREIGN KEY (map_id)
        REFERENCES map(map_id) ON DELETE RESTRICT
);

DROP TABLE IF EXISTS adjacent;
CREATE TABLE adjacent
(
    ter_id int(11) NOT NULL,
    adj_ter_id int(11) NOT NULL,
    CONSTRAINT pk_adjacent PRIMARY KEY (ter_id, adj_ter_id),
    CONSTRAINT fk_ter_id FOREIGN KEY (ter_id)
        REFERENCES territory(ter_id) ON DELETE RESTRICT,
    CONSTRAINT fk_adj_ter_id FOREIGN KEY (adj_ter_id)
        REFERENCES territory(ter_id) ON DELETE RESTRICT
);

DROP TABLE IF EXISTS supplier;
CREATE TABLE supplier
(
    ter_id int(11) NOT NULL,
    cty_id int(11) NOT NULL,
    CONSTRAINT pk_supplier PRIMARY KEY (ter_id, cty_id),
    CONSTRAINT fk_ter_id FOREIGN KEY (ter_id)
        REFERENCES territory(ter_id) ON DELETE RESTRICT,
    CONSTRAINT fk_cty_id FOREIGN KEY (cty_id)
        REFERENCES country(cty_id) ON DELETE RESTRICT
);

DROP TABLE IF EXISTS triangle;
CREATE TABLE triangle
(
```

```
    tri_id int(11) AUTO_INCREMENT,
    ter_id int(11) NOT NULL,
    x1 int(11) NOT NULL,
    y1 int(11) NOT NULL,
    x2 int(11) NOT NULL,
    y2 int(11) NOT NULL,
    x3 int(11) NOT NULL,
    y3 int(11) NOT NULL,
    CONSTRAINT pk_triangle PRIMARY KEY (tri_id),
    CONSTRAINT fk_ter_id FOREIGN KEY (ter_id)
        REFERENCES territory(ter_id) ON DELETE RESTRICT
);

DROP TABLE IF EXISTS line;
CREATE TABLE line
(
    ln_id int(11) AUTO_INCREMENT,
    x1 int(11) NOT NULL,
    y1 int(11) NOT NULL,
    x2 int(11) NOT NULL,
    y2 int(11) NOT NULL,
    CONSTRAINT pk_line PRIMARY KEY (ln_id)
);

DROP TABLE IF EXISTS ter_ln_relation;
CREATE TABLE ter_ln_relation
(
    ter_id int(11) NOT NULL,
    ln_id int(11) NOT NULL,
    CONSTRAINT pk_ter_ln_relation PRIMARY KEY (ter_id, ln_id),
    CONSTRAINT fk_ter_id FOREIGN KEY (ter_id)
        REFERENCES territory(ter_id) ON DELETE RESTRICT,
    CONSTRAINT fk_ln_id FOREIGN KEY (ln_id)
        REFERENCES line(ln_id) ON DELETE RESTRICT
);

DROP TABLE IF EXISTS piece;
CREATE TABLE piece
(
    pce_id int(11) AUTO_INCREMENT,
    cty_id int(11) NOT NULL,
    ter_id int(11) NOT NULL,
    pce_type enum('fleet', 'army'),
    CONSTRAINT pk_pce_id PRIMARY KEY (pce_id),
    CONSTRAINT fk_ter_id FOREIGN KEY (ter_id)
        REFERENCES territory(ter_id) ON DELETE RESTRICT,
    CONSTRAINT fk_cty_id FOREIGN KEY (cty_id)
        REFERENCES country(cty_id) ON DELETE RESTRICT
);

DROP TABLE IF EXISTS order_type;
CREATE TABLE order_type
(
    odt_id int(11) AUTO_INCREMENT,
    order_text varchar(128) NOT NULL,
    CONSTRAINT pk_order_type PRIMARY KEY (odt_id)
);

DROP TABLE IF EXISTS orders;
CREATE TABLE orders
(
    ord_id int(11) AUTO_INCREMENT,
    cty_id int(11) NOT NULL,
    pce_id int(11) NOT NULL,
    gam_season enum('spring', 'fall') NOT NULL,
    gam_year year(4) NOT NULL,
    order_type int(11) NOT NULL,
    destination int(11),
    executed tinyint(1) DEFAULT 0,
    CONSTRAINT pk_orders PRIMARY KEY (ord_id),
    CONSTRAINT uq_orders UNIQUE (cty_id, pce_id, gam_season, gam_year),
    CONSTRAINT fk_cty_id FOREIGN KEY (cty_id)
```

```
        REFERENCES country(cty_id) ON DELETE RESTRICT,
    CONSTRAINT fk_pce_id FOREIGN KEY (pce_id)
        REFERENCES piece(pce_id) ON DELETE RESTRICT,
    CONSTRAINT fk_order_type FOREIGN KEY (order_type)
        REFERENCES order_type(odt_id) ON DELETE RESTRICT,
    CONSTRAINT fk_destination FOREIGN KEY (destination)
        REFERENCES territory(ter_id) ON DELETE RESTRICT
);

DROP TABLE IF EXISTS operands;
CREATE TABLE operands
(
    opr_id int(11) AUTO_INCREMENT,
    ord_id int(11) NOT NULL,
    ter_id int(11) NOT NULL,
    CONSTRAINT pk_operands PRIMARY KEY (opr_id),
    CONSTRAINT fk_ord_id FOREIGN KEY (ord_id)
        REFERENCES orders(ord_id) ON DELETE RESTRICT,
    CONSTRAINT fk_ter_id FOREIGN KEY (ter_id)
        REFERENCES territory(ter_id) ON DELETE RESTRICT
);

COMMIT;
```

The Rules of Diplomacy®

★★★★ 4TH EDITION 2000

THE GAME OF INTERNATIONAL INTRIGUE

Set In Pre-World War I Europe

- **PLAYERS:** 2 to 7
- **AGES:** 12 and up

COMPLEXITY LEVEL

- Advanced
- **Challenging**
- Moderate

- **CONTENTS**
  - Mapboard
  - Conference maps
  - 70 Army playing pieces
  - 70 Fleet playing pieces
  - Flag marker sheet

GAME DESIGNER: Allan B. Calhamer

At the beginning of the 20th century, Europe was a complicated cauldron of political intrigue. You are about to travel back to those times and change the course of history in your favor.

PLAYERS AND COUNTRIES

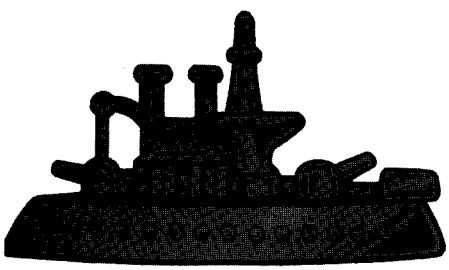
The game of DIPLOMACY is best played by seven players. Rules for fewer players are included in the Alternate Way to Play section of this booklet. Each player represents one of the seven "Great Powers of Europe" in the years prior to World War I. These Great Powers include England, Germany, Russia, Turkey, Italy, France and Austria-Hungary (hereafter referred to as Austria). At the start of the game, the players randomly decide which Great Power each will represent. This is the only element of chance in the game.

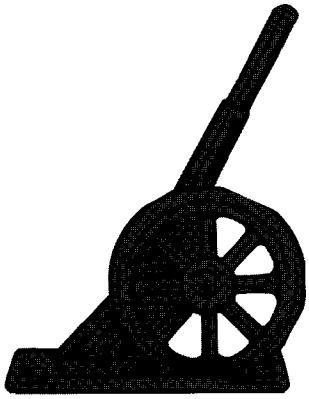
Note: At various places in the rules, the term "country" is used generically to represent "Great Power."

OBJECT OF THE GAME

As soon as one Great Power controls 18 supply centers, it is considered to have gained control of Europe. The player representing that Great Power is the winner.

However, players can end the game by agreement before a winner is determined. In this case, all players who still have pieces on the board share equally in a draw.





MAPBOARD

Boundaries: Boundaries between major countries are marked with heavy white lines. All major powers are also divided into provinces and supply center provinces by thinner black lines. The oceans and waterways are also divided into separate provinces by thin black lines. All countries and provinces (land and water) are identified by name.

Types of Provinces: There are three types of provinces: *Inland, water* and *coastal.* Only Armies move on inland provinces and only Fleets move on water provinces. A coastal province is land that is adjacent to one or more water provinces. For example, Denmark, Brest, and Spain are coastal provinces. *An Army or a Fleet can occupy a coastal province.*

Supply Centers: 34 inland and coastal provinces on the mapboard are designated as supply centers. Each supply center is marked with a star. A Great Power has as many Armies or Fleets as the number of supply centers it controlled at the end of the last Fall turn. Consequently, there will never be more than 34 Armies and Fleets (also referred to as "units") on the mapboard at one time. A country gains or loses units in accordance with the number of supply centers it controls. More on this later.

UNITS (ARMIES AND FLEETS)

Each Army unit is represented by a cannon playing piece. Each Fleet unit is represented by a battleship playing piece. The unit colors of each Great Power are displayed on the edge of the mapboard and indicated in the chart below. If an expanding Great Power runs out of Army or Fleet units, the units of an eliminated country can be used.

All units have the same strength. No one Army is more powerful than another. No single Fleet is stronger than another. During the game, various units will support each other to increase their strength and attack weaker adversaries.**There can only be one unit in a province at a time.** No exceptions.

STARTING POSITIONS

Supply Centers: At the start of the game, each Great Power controls three supply centers, with the exception of Russia, which controls four. Place the appropriate unit on the designated supply center as shown in the table below. **Note that "A" indicates an Army and "F" indicates a Fleet.**

The 12 remaining supply centers are not occupied at the start of the game.

Country	Unit color	Unit city	Unit city	Unit city
AUSTRIA	red	A Vienna	A Budapest	F Trieste
ENGLAND	dark blue	F London	F Edinburgh	A Liverpool
FRANCE	light blue	A Paris	A Marseilles	F Brest
GERMANY	black	A Berlin	A Munich	F Kiel
ITALY	green	A Rome	A Venice	F Naples
RUSSIA	white	A Moscow	F Sevastopol	A Warsaw
		F St. Petersburg (SC)		
TURKEY	yellow	F Ankara	A Constantinople	A Smyrna

Flag Markers: Carefully punch out the various flag markers from the cardboard sheet. There is one set of markers for each Great Power. Players can use these markers to identify which supply centers they control on the mapboard. One side shows the flag of the Great Power and the other side shows the color of the units of that country. Use whichever side is more helpful.

HOW TO PLAY

OVERVIEW

DIPLOMACY is a game of negotiations, alliances, promises kept, and promises broken. In order to survive, a player needs help from others. In order to win the game, a player must eventually stand alone. Knowing whom to trust, when to trust them, what to promise, and when to promise it is the heart of the game. Remember, you are a diplomat first, a commander second.

At the beginning of each turn, players meet together one-to-one or in small groups to discuss their plans and suggest strategies. Alliances between players are openly or secretly made, and orders are (hopefully) coordinated. Immediately following this period of "diplomacy," each player secretly writes an order for each of his/her units on a slip of paper. When all players have written their orders, the orders are simultaneously revealed, and then the orders are all resolved. Some units are moved, some have to retreat, and some are removed. Resolving orders is the most challenging part of the rules, requiring complete knowledge of the rules.

Each turn represents six months of time. The first turn is called a Spring Turn and the next a Fall Turn. After each Fall Turn, each Great Power must reconcile the number of units it controls with the number of supply centers it controls. At this time some units are removed and new ones are built.

Each turn has a series of *phases*. Here are the phases in a complete two-turn year:

Spring four-phase turn

1. Diplomatic Phase
2. Order Writing Phase
3. Order Resolution Phase
4. Retreat and Disbanding Phase

Fall five-phase turn

1. Diplomatic Phase
2. Order Writing Phase
3. Order Resolution Phase
4. Retreat and Disbanding Phase
5. Gaining and Losing Units Phase

After a Fall Turn, if one Great Power controls 18 or more supply centers, the game ends and that player is declared the winner.

1. DIPLOMATIC PHASE

During this phase, players meet to discuss their plans for upcoming turns. Alliances are made and strategies are set. These "diplomatic negotiations" take place before each turn. Negotiations last 30 minutes before the first turn and 15 minutes before each turn thereafter. Negotiations may end sooner if all players agree.

Conversations, deals, schemes, and agreements among players will greatly affect the course of the game. During diplomatic negotiations, players may say anything they wish. Some players usually go to another room or organize private groups of two or three. They may try to keep their conversations secret. They may try to overhear the conversations of others. These conversations usually consist of bargaining or joint military planning, but they may include exchanges of information, denouncements, threats, spreading of rumors, and so on. Public announcements may be made and documents may be written, made public, or kept secret, as the players see fit. These discussions and written agreements, however, do *not* bind a player to anything he/she may say. Deciding whom to trust as situations arise is an important part of the game.

Using the Conference Maps during diplomatic negotiations is an excellent way to keep track of locations, strategies and alliances.

2. ORDER WRITING PHASE

Each player secretly writes "orders" for each of his/her units on a slip of paper. All players then reveal orders at the same time. Each player reads his/her orders while others make sure that what they hear is what is written. A legal order must be followed. An order written by mistake, if legal, must be followed. An "illegal" or ambiguous order or an order that is judged to be unsuccessful is not followed. A unit that is given an illegal order (or given no order) must stand in place. (The unit holds.) A poorly written order that has only one meaning must be followed.

ORDER DATES

Orders alternate between Spring and Fall beginning with the year 1901. For example, the first set of turn orders are considered to be "Spring 1901." The second set are considered to be "Fall 1901." The third set are considered to be "Spring 1902" and so on.

ORDER FORMAT

Players should make a list of their units and the provinces they occupy for easy reference during diplomatic conferences. In each set of orders, the type of unit is written first ("A" or "F") followed by the province that each unit occupies. For example, "A Paris" or "A Par" is short for an Army in Paris. This is followed by the order that the unit is given. For example, "A Par Holds" means that the Army in Paris should hold, or stay in place. The designation of "A" or "F" in orders is to remind players of their pieces. If you leave out the unit designation in an order, the order does not fail since there can be only one possible unit in a province.

ABBREVIATIONS

Players may refer to the abbreviations shown on the back cover of this booklet for countries or provinces when writing their orders. A number of provinces begin with the same three letters – so many of those provinces have special abbreviations. When in doubt, write it out.

Keep in mind that only one unit can be in a province (inland, water or coastal) at the same time – so there should not be any confusion as to which unit is being ordered.

GAMEMASTER

If an additional knowledgeable person is available, that person could serve as the gamemaster. The gamemaster could keep time for the negotiation sessions, collect and read orders, resolve issues, and make rulings when necessary. This role should be strictly neutral.

TYPES OF ORDERS

On each turn, each Great Power can order all, some, or none of its units to do one of the following:

- Hold
- Move
- Support
- Convoy

Note: Only Fleets can be ordered to convoy.

UNDERLINED RESULTS

Examples of orders are listed throughout this booklet. Orders that are not executed (because of interference by other orders) are underlined. This is a Diplomacy standard that has been in effect for years and is used in many Diplomacy strategy guides and other literature. While playing the game, there is no need to underline orders.

HOLD ORDER

You can attempt to keep a unit in place by ordering it to "hold." Not giving a unit an order is interpreted as ordering it to hold. Following is an example of a hold order:

F London Holds (or) F Lon-Holds

Note: In this booklet, examples of hold orders that failed are underlined to show that the unit was not able to hold (or stay) in a province.

Additional information about hold orders is included in the following sections.

MOVE ORDER

Throughout the game, units will be ordered to move to provinces that are occupied. This is referred to as "attacking," and will be discussed in detail later in the rules.

Writing a Move Order

A move order is written with a dash to separate the unit type and location from the order. For example, an order to move from Paris to Burgundy would look like this:

A Paris-Burgundy (or) A Par-Bur

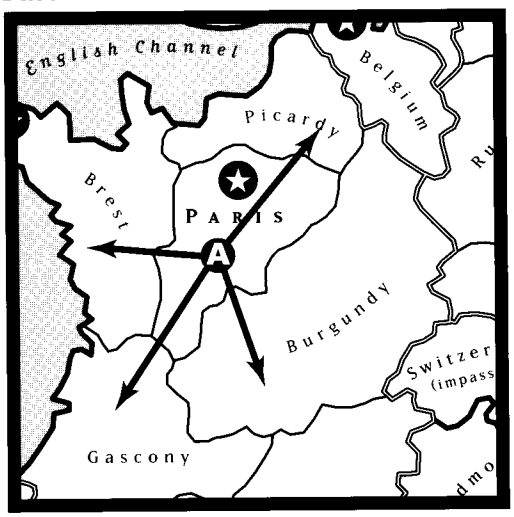
Army Movement

An Army can be ordered to move into an adjacent inland or coastal province. Armies cannot be ordered to move into a water province. Since no two units can occupy the same province at the same time, an Army that is ordered to move to an adjacent province can end up not moving at all (because of the positions or orders of other units). This is explained in the *Conflicts* section later in the rules.

Note: An Army can move across water provinces from one coastal province to another via one or more Fleets. This is called a "convoy" and is explained in the Convoy Order section later in the rules.

Army Movement Example: An Army in Paris could move to Brest, Picardy, Burgundy or Gascony. See Diagram 1.

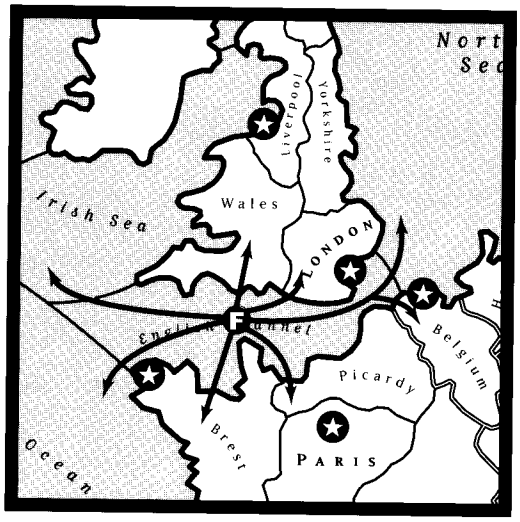
DIAGRAM 1



Fleet Movement

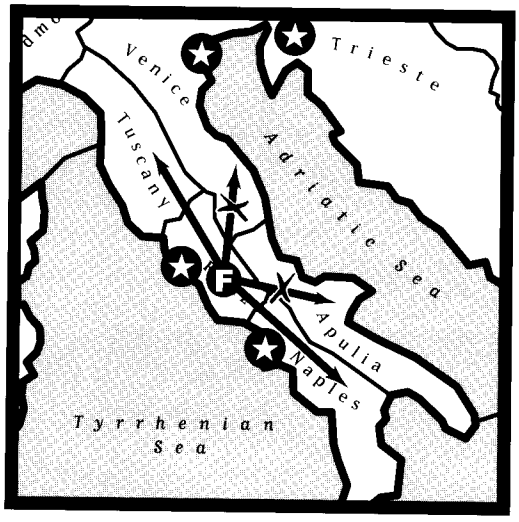
A Fleet can be ordered to move to an adjacent water province or coastal province. Fleets cannot be ordered to move to an inland province. Diagram 2 shows that a Fleet in the English Channel can move to the Irish Sea, Wales, London, Belgium, Picardy, Brest, the North Sea or the Mid-Atlantic.

DIAGRAM 2



When a Fleet is in a coastal province, its warships are considered to be at any point along the coast of that province. A Fleet in a coastal province can be ordered to move to an adjacent coastal province only if it is *adjacent along the coastline* (as if the Fleet was moving down the coast). For example, in Diagram 3 a Fleet in Rome can be ordered to move from Rome to Tuscany or to Naples (or to the Tyrrhenian Sea). But a Fleet in Rome cannot be ordered to move to Venice or Apulia because, although those provinces are adjacent along an inland boundary, they are not adjacent *along the coastline*.

DIAGRAM 3



Restricted Movement

Any location on the mapboard that is not named cannot be occupied. Switzerland is impassable and cannot be occupied. With the exception of England, islands cannot be occupied.

Specific Movement Clarifications

There are a few tricky areas on the map. How to move into and out of them is explained below:

Bulgaria, Spain, and St. Petersburg: These are the only coastal provinces that have two separately *identified coasts*. A Fleet entering one of these provinces enters along one coast and can then move to a province adjacent to that coast only. The Fleet, nevertheless, is considered to be occupying the entire province. Such a Fleet should be placed on the coastline rather than completely inland. For example, a Fleet at Spain's North Coast cannot be ordered to move to the Western Mediterranean or to the Gulf of Lyon or to Marseilles. It is, however, considered to be occupying all of Spain.

If a Fleet is ordered to one of these provinces, and it is possible for the Fleet to move to either coast, the order must specify which coast, or the Fleet does not move. For example, a Fleet in Constantinople can move to Bulgaria's East or South Coast. The order would be written "F Con-Bul EC" or "F Con-Bul SC." Likewise, a Fleet in the Mid-Atlantic Ocean can move to Spain's North or South Coast, but the order must specify which coast.

Kiel and Constantinople: Because of the waterways that run through these two provinces, they are considered as having one coast. Fleets can enter them along one coast and be considered anywhere along the coastline. For example, a Fleet could move from the Black Sea to Constantinople on one turn ("F Bla-Con") and then on a later turn move from Constantinople to the Aegean Sea (or other adjacent provinces). Likewise, a Fleet could move from Holland to Kiel on one turn and then move from Kiel to Berlin on a later turn (through the Kiel Canal) without having to go around or to Denmark. Armies can also pass into and out of these provinces, freely bridging these waterways. This does not mean that units can jump over these provinces.

Sweden and Denmark: An Army or Fleet can move from Sweden to Denmark (or vice versa) in one turn. A Fleet moving from the Baltic Sea cannot move directly to the Skaggerak province (or vice versa), but must first move to Sweden or Denmark. The common border with Denmark does not separate the coast of Sweden into two coastlines. Denmark does not border on Berlin.

Standoffs

The following common situations involve forces of equal strength trying to occupy the same province at the same time. These situations are called standoffs. These rules apply when one or more countries are involved. There are a few exceptions to these rules, which are described later.

- **Units of equal strength trying to occupy the same province cause all those units to remain in their original provinces.** If two or more units are ordered to the same province, none of them can move. (This is also true of equally *supported* units, which will be explained in the next section.) In Diagram 4, if the German Army in Berlin is ordered to Silesia and the Russian Army in Warsaw is ordered to Silesia, neither unit will move and Silesia will remain vacant.

Note: In this booklet, examples of move orders that failed are underlined to show that they were not successful. (The unit did not end up moving to the ordered province.)

DIAGRAM 4

GERMANY:
A Ber–Sil

RUSSIA:
A War–Sil

- **A standoff does not dislodge a unit already in the province where the standoff took place.** If two units (or forces of equal strength) attack the same province, thus standing each other off, a unit already in that province is not dislodged. So, in Diagram 4, if there had been a unit holding in Silesia, the results would be the same and the unit in Silesia would remain.

- **One unit not moving can stop a unit or series of units from moving.** If a unit is ordered to hold, or is prevented from moving, and other units are ordered into its province, those other units cannot move. (It's like a traffic backup!) In Diagram 5, there is a Russian Army in Prussia. The Russian player told Germany that he would move out of Prussia (but he lied and ordered the Army to hold instead). The German player ordered his Army from Berlin to Prussia and his Fleet from Kiel to Berlin. The result is that nothing moves.

DIAGRAM 5

GERMANY:
F Kiel–Ber
A Ber–Pru

RUSSIA:
A Pru–Holds

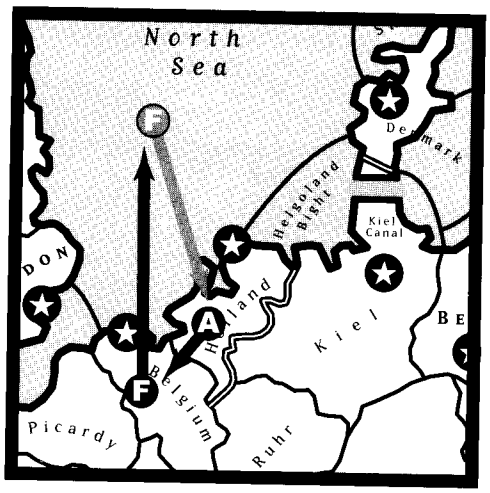
- **Units cannot trade places without the use of a convoy.** If two units are each ordered to the province that the other occupies, neither can move. For example, in Diagram 6, neither unit would move. (There is a way around this through the use of convoys, explained later in the rules.)

DIAGRAM 6

GERMANY:
F Ber–Pru
A Pru–Ber

- **Three or more units can rotate provinces during a turn provided none directly trade places.** For example, in Diagram 7, all orders would succeed as no one unit directly trades places with another.

DIAGRAM 7



ENGLAND:
A Hol–Bel
F Bel–Nth
FRANCE:
F Nth–Hol

SUPPORT ORDER

This is the most critical and complex section of the rules. The "support" and "cutting support" rules must be understood in order to resolve most orders.

Overview

Since all units have equal strength, one unit cannot attack and advance against another without help. That "help" is called support. If an attack is successful, the attacking unit moves into the province to which it was ordered. If the unit that was attacked had no orders of its own to move elsewhere, it is defeated and dislodged from the province. The dislodged unit must retreat or be disbanded. Retreats are explained later.

An Army or Fleet can provide support to another Army or Fleet. Support can be offensive (supporting an attacking move order) or defensive (supporting a hold, support or convoy order). By supporting each other, attacking or defending units gain increased strength. For example, a unit holding with two supports has the strength of three: itself plus two supporters. Support can be provided to a fellow unit or to another Great Power's unit. *Support can be given without consent and cannot be refused!* This causes some wonderfully unexpected moments in the game.

A unit moves with its own strength combined with all of its valid supports. Unless it is opposed by a unit that is equally or better supported, it can complete its move. One unit supporting another provides a combined strength of two and so will defeat an opponent's unit that is unsupported. Likewise, a unit with two supporting units (strength of 3) will defeat an opponent's unit with only one support (strength of 2).

How to Support

A unit gives up the chance to move on a turn in order to support another unit's order. The province to which a unit is providing support must be one to which the supporting unit could have legally moved during that turn. Thus, an Army in Brest cannot support a Fleet in the English Channel because an Army can't move into a water province. Likewise, a Fleet in Rome cannot support a unit's move to Venice because, although adjacent by land, the Fleet cannot move to Venice from Rome.

A Fleet that can move to a province with two separate coasts (a Fleet in the Mid-Atlantic, for example) can support another Army or Fleet into that province (in this case Spain), without regard to separate coastlines.

Writing a Support Order

1. Write your unit type (A or F).
2. Follow this with the province in which your unit is located.
3. Then, write an "S" (for Support).
4. Finally, write the type, current location, and destination of the unit receiving support (if the supported unit is moving).

Example: "A Par S A Mar-Bur" orders an Army in Paris to support an Army in Marseilles moving into Burgundy.

Supporting a Unit

- **A unit not ordered to move can be supported by a support order that only mentions its province.** A unit that is ordered to hold, convoy, support, or not ordered at all can receive support in holding its position. For example, if the order is written "F Den S F Bal," then the Fleet in Denmark will support the Fleet in the Baltic Sea as long as the Fleet in the Baltic is holding, convoying, or supporting. If the Fleet in the Baltic attempts to move, then the support from Denmark is invalid.
- **A unit ordered to move can only be supported by a support order that matches the move the unit is trying to make.** For example, an Army in Bohemia is ordered to support an Army in Munich in its move to Silesia (A Boh S A Mun-Sil). However, the Army in Munich is ordered to move to Tyrolia instead (A Mun-Tyr). The support order fails because the move it is supporting is not the move that was ordered. This support order does not become a support order to hold.

Simple Support

In Diagram 8, the French Army in Gascony supports the Army in Marseilles to Burgundy. The German Army in Burgundy will be dislodged.

DIAGRAM 8

FRANCE:
A Mar–Bur
A Gas S A Mar–Bur

GERMANY:
A Bur–Holds

In Diagram 9, the German Army in Silesia is supported by the Fleet in the Baltic in pushing the Russian Army out of Prussia. Note that the German Army and Fleet are both adjacent to the target province (Prussia) but not to each other. A unit does not have to be adjacent to the unit it is supporting. However, it must be next to the province into which it is giving support and must be able to legally move there itself.

DIAGRAM 9

GERMANY:
A Sil–Pru
F Bal S A Sil-Pru

RUSSIA:
A Pru–Holds

Support in Standoffs

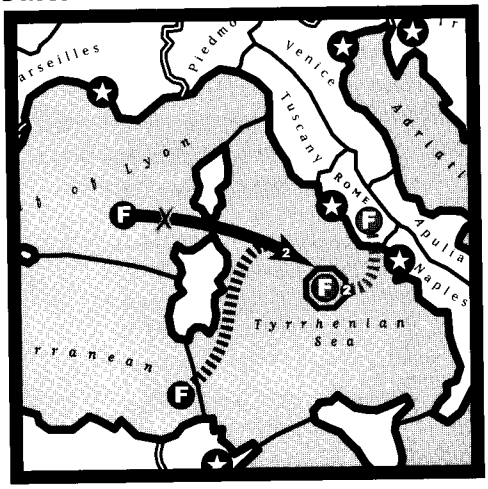
Diagrams 10 and 11 show two common standoff situations. In both cases, a strength of two meets a strength of two and all units stand in place. In Diagram 10, if there had been a Fleet in the Tyrrhenian, it would not be dislodged by the standoff. (A standoff does not dislodge a unit already in the province where the standoff took place.)

DIAGRAM 10

FRANCE:
F Gol–Tyn
F Wes S F Gol–Tyn

ITALY:
F Nap–Tyn
F Rom S F Nap–Tyn

DIAGRAM 11



FRANCE:
F Gol–Tyn
F Wes S F Gol–Tyn

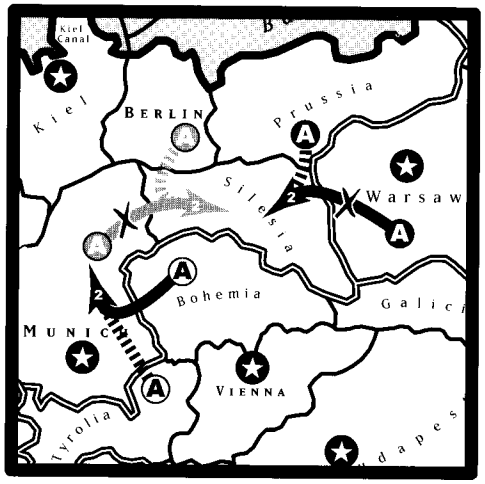
ITALY:
F Tyn–Holds
F Rom S F Tyn–Holds

Dislodgment in Standoffs

- **A dislodged unit can still cause a standoff in a province different from the one that dislodged it.** When two or more equally supported units are ordered to the same province, neither can move – even if one of them is dislodged from a province *other than the one that is the target of the standoff during the same turn.*

In Diagram 12, the Austrian attack from Bohemia successfully dislodges the Germany Army in Munich. However, that Army in Munich still causes a standoff with the Russian Army trying to enter Silesia.

DIAGRAM 12



AUSTRIA:
A Boh–Mun
A Tyr S A Tyr–Mun

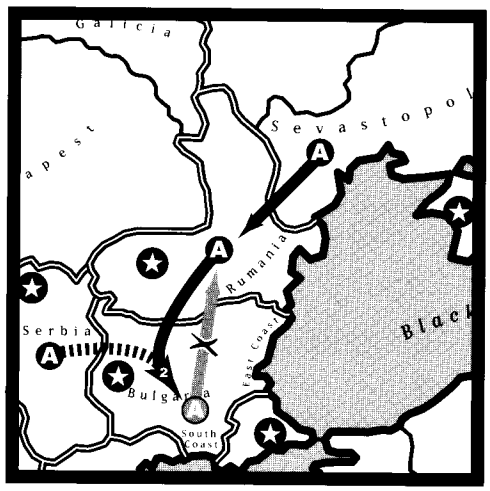
GERMANY:
A Mun–Sil
A Ber S A Mun–Sil

RUSSIA:
A War–Sil
A Pru S A War–Sil

- **A dislodged unit, even with support, has no effect on the province that dislodged it.** If two units are ordered to the same province and one of them is dislodged by a unit coming from that province, the other attacking unit can move. This situation does not result in a standoff since the dislodged unit has no effect on the province that dislodged it.

In Diagram 13, the Russian Army in Rumania dislodges the Turkish Army in Bulgaria. That Turkish Army, and the Russian Army in Sevastopol are both ordered to Rumania, which would normally cause a standoff. However, because Rumania dislodged the Army in Bulgaria, it has no effect on Rumania at all. This allows the Sevastopol Army to enter Rumania. The Army in Bulgaria must retreat.

DIAGRAM 13



TURKEY:
A Bul–Rum

RUSSIA:
A Rum–Bul,
A Ser S A Rum–Bul,
A Sev–Rum

In Diagram 14, even though the Turkish unit has support, it fails to prevent the unsupported Russian move into Rumania because a unit coming from Rumania dislodged the Turkish unit.

DIAGRAM 14

TURKEY:
A Bul–Rum
F Bla S A Bul–Rum

RUSSIA:
A Rum–Bul
A Gre S A Rum–Bul
A Ser S A Rum–Bul
A Sev–Rum

In the previous two examples, if Russia had not ordered "A Sev-Rum," Rumania would have been vacant, but not as the result of a standoff. (There was no standoff.) This is discussed further in the *Retreats* section later in the booklet.

Cutting Support

Support can be cut. This will cause the support order to fail and support will not be given.

Note: In this booklet, examples of support orders that failed are underlined to show that the support was cut, not to show that the supported unit's order failed.

- **Support is cut if the unit giving support is attacked from any province except the one where support is being given.** The support is cut whether this attack on the supporting unit succeeds or not.

In Diagram 15, the support from the Army in Silesia is cut by an attack from Bohemia. Note that it was enough to attack the Army giving support to cut that support. It was not necessary to dislodge the supporting unit to cut that support.

DIAGRAM 15

GERMANY:
A Pru–War
A Sil S A Pru–War

RUSSIA:
A War Holds
A Boh–Sil

- **Support is cut if the unit giving support is dislodged.** If a unit ordered to support another unit is dislodged by an attack from any province (including the province into which it is giving support) then the support is "cut." The unit that was to receive support does not receive it.

In Diagram 16, the German support is not cut by the attack from Warsaw because that is the province into which support is being given. To cut support, the Army in Warsaw would have to *dislodge* the Army in Silesia, not merely attack it.

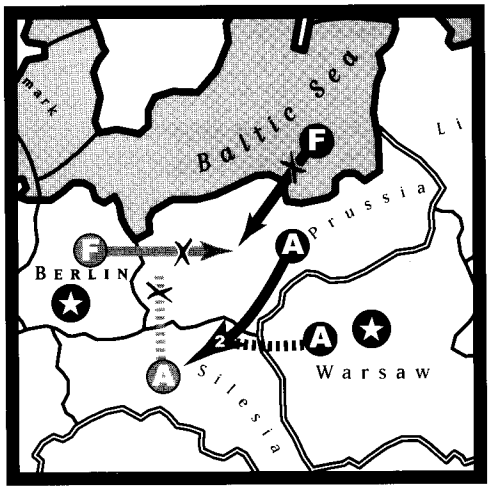
DIAGRAM 16

GERMANY
A Pru–War
A Sil S A Pru–War

RUSSIA:
A War–Sil

In Diagram 17, the Russian Army coming from Prussia dislodges the German Army in Silesia. The support of the Silesian Army is thus cut and the German Army in Berlin stands off the Russian Fleet in the Baltic.

DIAGRAM 17



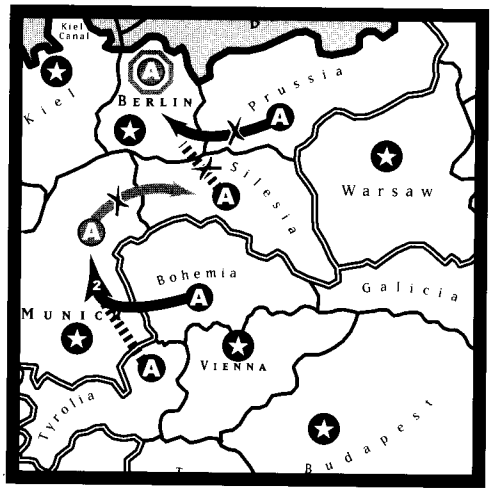
GERMANY:
F Ber–Pru
A Sil S A Ber–Pru

RUSSIA:
A Pru–Sil
A War S A Pru–Sil
F Bal–Pru

- **A unit being dislodged by one province can still cut support in another.** Just as a unit being dislodged by one province can still cause a standoff in another, a unit still manages to cut support even if it is dislodged. Just make sure that the dislodgment is not coming from the province where the unit is giving support. (Remember this rule: A dislodged unit, even with support, has no effect on the province that dislodged it.)

In Diagram 18, even though the German Army in Munich is dislodged by a Russian attack, it is still able to cut the support of the Russian Army in Silesia. This prevents the Russian Army in Prussia from entering Berlin.

DIAGRAM 18



GERMANY:
A Ber Holds
A Mun–Sil

RUSSIA:
A Pru–Ber
A Sil S A Pru–Ber
A Boh–Mun
A Tyr S A Boh–Mun

Note: In complicated situations, it helps to first determine what support, if any, is cut. Once this is determined, it is easier to resolve orders.

CONVOY ORDER

Convoying an Army Across One Water Province

A Fleet in a water province (not a coastal province) can convoy an Army from any coastal province adjacent to that water province to any other coastal province adjacent to that water province. To do this, the Army must be ordered to move to the intended province and the Fleet must be ordered to convoy it.
A Fleet cannot convoy a Fleet.

Writing Convoy Orders

Just as "S" indicates support, the letter "C" is used to indicate convoy. Following is an example of a convoy order:

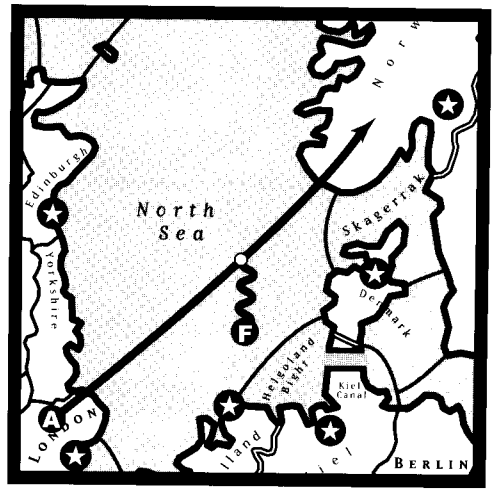
A Ank-Sev. F Bla C A Ank-Sev.

A Fleet cannot convoy more than one Army during the same turn. The order to the Fleet must contain both the location and the destination of the Army being convoyed. Just as with support orders, the convoy order must match the move order given by the Army being convoyed. For example, if the Army in Rumania is ordered to Armenia (A Rum-Arm) and the convoy order is written to take it to Ankara (F Bla C A Rum-Ank), then the convoy would fail and the Army would remain in Rumania.

Note: Fleets in any *coastal* province (including Constantinople, Denmark and Kiel) cannot convoy.

In Diagram 19, the Fleet in the North Sea convoys the Army in London to Norway.

DIAGRAM 19



ENGLAND:
A Lon-Nwy
F Nth C A Lon-Nwy

"Support" Cannot be Convoyed

Only Armies can be convoyed. "Support" cannot be transported from one Army via a convoy to another unit. For example, the orders shown below in bold are illegal and clearly fail.

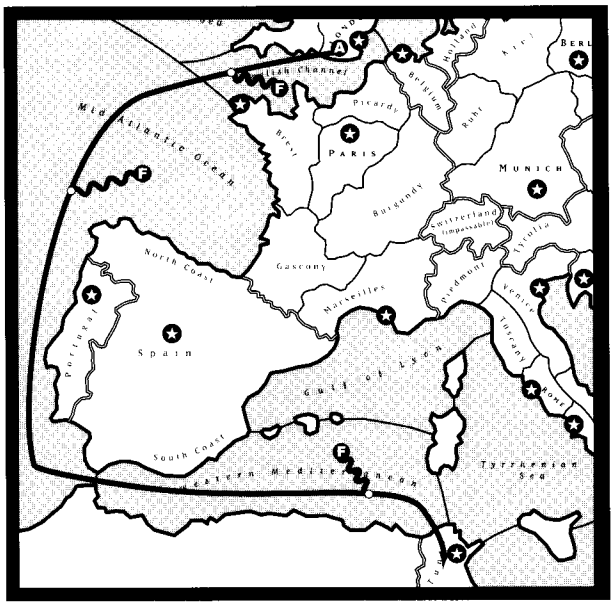
England: A Pic-Bre, **A Lon S A Pic-Bre**
F Eng C A Lon S A Pic-Bre
France: F Bre Holds.

Convoying an Army Across Several Water Provinces

If Fleets occupy adjacent water provinces, an Army can be convoyed through all these water provinces on one turn, landing in a coastal province adjacent to the final Fleet in the chain.

In Diagram 20, the English Army from London goes to Tunis on a single move, with help from the French player.

DIAGRAM 20



ENGLAND:
A Lon–Tun
F Eng C A Lon–Tun
F Mid C A Lon–Tun

FRANCE:
F Wes C English A Lon–Tun

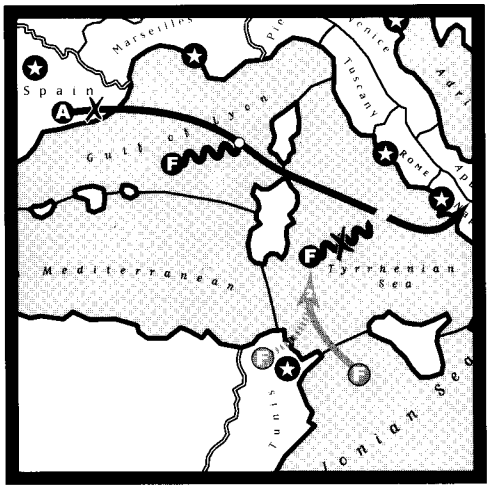
Disrupting a Convoy

- **Dislodgment of a fleet in a convoy causes the convoy to fail.** If a Fleet ordered to convoy is *dislodged* during the turn, the Army to be convoyed remains in its original province. An attack on a convoying Fleet, which does not dislodge it, does not affect the convoy.
- **A convoy that causes the convoyed Army to standoff at its destination results in that army remaining in its original province.** If a convoyed Army arrives at its destination province and is unable to stay there because of a standoff with another unit(s), then that convoyed Army must remain in its original coastal province. (It could still be forced out of its original province by a successful attack there.) An Army can be supported into its destination province to help avoid a standoff.

Note: In this booklet, examples of convoy orders that failed are underlined to show that the underlined Fleet was dislodged. Other Fleets in a convoy chain will not be underlined.

In Diagram 21, the Fleet in the Tyrrhenian is dislodged, so the French Army does not move from Spain to Naples.

DIAGRAM 21



FRANCE:
A Spa–Nap
F GoL C A Spa–Nap
F Tyn C A Spa–Nap

ITALY:
F Ion–Tyn
F Tun S F Ion–Tyn

Rare Cases and Tricky Situations

The above rules should resolve most situations that arise in DIPLOMACY. There are, however, a few exceptions and rare situations that can occur. They are explained below.

Self Dislodgment

A country cannot dislodge or support the dislodgment of one of its own units, even if that dislodgment is unexpected. This is one time when support is refused or negated when it would otherwise be legal. However, such orders can be written for other reasons, such as creating a standoff. Following are some examples to further explain this rule:

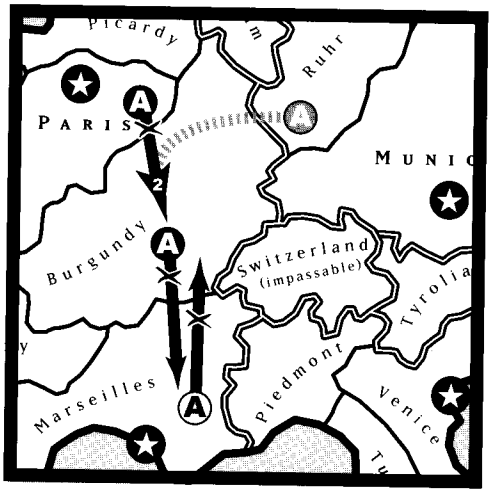
In Diagram 22, the French Army in Paris, supported by its Army in Marseilles cannot dislodge it own Army in Burgundy.

DIAGRAM 22

FRANCE:
A Par–Bur
A Mar S A Par–Bur
A Bur–Holds

In Diagram 23, the French Army in Paris, although supported by the German Army in Ruhr, cannot dislodge its own Army in Burgundy.

DIAGRAM 23



FRANCE:
A Par–Bur
A Bur–Mar

GERMANY:
A Ruh S French A Par–Bur

ITALY:
A Mar–Bur

In Diagram 24, the German Army in Ruhr, supported by the French Army in Paris, cannot dislodge the French Army in Burgundy because France cannot legally support an attack against one of its own units. However, if Germany had supported its own attack (from Munich), then the French Army in Burgundy would be dislodged.

DIAGRAM 24

GERMANY:
A Ruh–Bur
A Mun–Holds

FRANCE:
A Par S German A Ruh–Bur
A Bur–Holds

In Diagram 25, the German Army in Munich is in a standoff with the Austrian Army in Tyrolia, so neither unit moves. German Armies in Ruhr and Silesia tried to create a standoff with each other in Munich. However, the Austrian Army in Bohemia sneakily gave support to the German unit from Silesia into Munich. In most cases, this supported attack from Silesia into Munich would beat the unsupported attack from Ruhr. But since that would result in Germany dislodging one of its own units, the move fails.

DIAGRAM 25

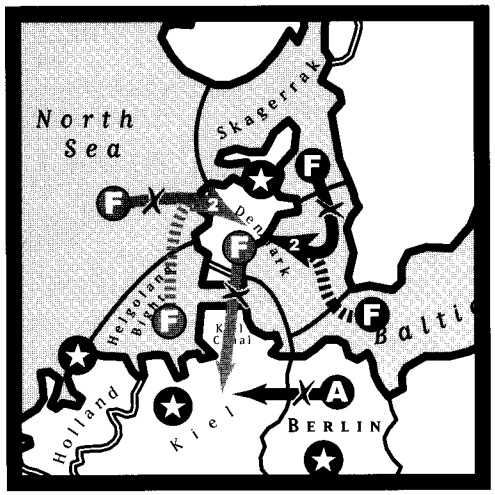
GERMANY:
A Mun–Tyr
A Ruh–Mun
A Sil–Mun

AUSTRIA:
A Tyr–Mun
A Boh S
German A Sil–Mun

The next example demonstrates a situation in which you might write self-dislodgment orders to create a standoff. This is sometimes a good defensive move.

In Diagram 26, England cannot dislodge its own unit, but its supported attack on Denmark is necessary to standoff the supported Russian attack on the same province.

DIAGRAM 26



ENGLAND:
F Den–Kiel
F Nth–Den
F Hel S F Nth–Den

RUSSIA:
A Ber–Kiel
F Skag–Den
F Bal S F Skag–Den

Self Standoff

While a country cannot dislodge its own units, it can create standoffs by ordering two equally-supported attacks on the same province. This is often done to maintain control of three provinces with two units. However, if one of the attacks has more support than the other, it will succeed.

In Diagram 27, the Austrian player is trying to control Serbia, Budapest and Vienna with two units, keeping Budapest vacant. However, the move "A Ser-Bud" succeeds because of unexpected Russian support. It would not succeed if there was an Austrian Army already in Budapest, as it would be dislodging its own unit. The move succeeds whether the support is from a foreign unit (as illustrated) or from a unit of the same country.

DIAGRAM 27

AUSTRIA
A Ser–Bud
A Vie–Bud

RUSSIA:
A Gal S Austrian A Ser–Bud

Cutting Support on Your Own Units

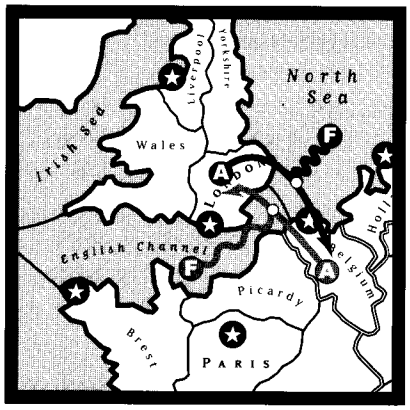
An attack by a country on one of its own units does not cut support. This rule is in the same spirit as the *Self-Dislodgment* rules. A country cannot dislodge one of its own units nor can it cut its own support.

Exchanging Places via a Convoy

Two units can exchange places if either or both are convoyed. This is the exception to the earlier rule that stated, "Units cannot trade places without the use of a convoy."

In Diagram 28, all moves succeed.

DIAGRAM 28



ENGLAND:
A Lon–Bel
F Nth C A Lon–Bel

FRANCE:
A Bel–Lon
F Eng C A Bel–Lon

Land and Convoy Routes

In some rare cases, orders are written so that an Army could arrive at its destination either by land or convoy. When this happens, the following qualifiers apply:

- If at least one of the convoying Fleets belongs to the player who controls the Army, then the convoy is used. The land route is disregarded.
- If none of the convoying Fleets belongs to the player who controls the army, then the land route is used. However, the player controlling the army can use the convoy route if he/she indicated "via convoy" on the Army move order in question.

This prevents foreign powers from kidnapping an Army and convoying it against its will.

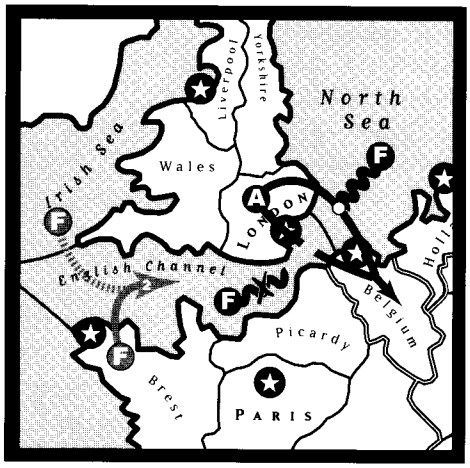
(Note that in the CD-Rom version of DIPLOMACY, it is impossible to specify "via convoy" in an order. In that version, if either the overland route or the convoy route is valid, then the Army will move to its destination. This does allow an Army to be convoyed against it's will.)

More than One Convoy Route

An Army convoyed using alternate convoy orders reaches its destination as long as at least one convoy route remains open. Orders can be written to permit more than one route for convoying an Army from its origin to its destination. The Army is not prevented from moving unless all routes in the order are disrupted.

In Diagram 29, the Army in London has two convoy routes. Since only one was disrupted, the English Army lands in Belgium.

DIAGRAM 29



ENGLAND:
A Lon–Bel
F Eng–C A Lon–Bel
F Nth C A Lon–Bel

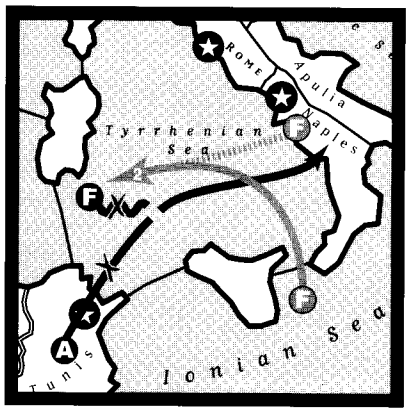
FRANCE:
F Bre–Eng
F Iri S F Bre–Eng

A Convoyed Attack Does Not Cut Certain Supports

A convoyed Army does not cut the support of a unit supporting an attack against one of the Fleets necessary for the Army to convoy. This is a tricky and rare situation, but without this rule (using Diagram 30 as an example), a paradox may occur.

In the orders shown below, France could argue that its Army cut the support of the Fleet in Naples, thus protecting the convoying Fleet from dislodgment. (France could state the rule, "Support is cut if the unit giving support is attacked from any province but the one where support is being given."). Italy could argue that dislodgment of the Fleet disrupted the convoy so that the Army could not arrive in Naples to cut that support. (Italy could state the rule, "Dislodgment of a fleet in a convoy causes the convoy to fail.") Since both rules are contradictory, the above new rule takes precedence. Therefore, the convoy is blocked and support is not cut.

DIAGRAM 30



FRANCE:
A Tun-Nap
F Tyn C A Tun-Nap

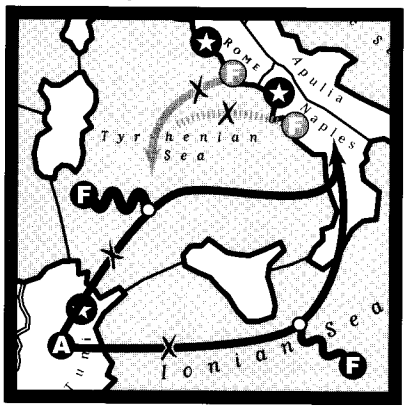
ITALY:
F Ion-Tyn
F Nap S F Ion-Tyn

Two More Tricky Situations

Following are two complicated examples that involve the *Alternate Convoy* rule and the *Convoyed Attack* rule. These situations are rare and do not come up in most games. But, here are the rules in case these issues do arise.

An Army with at least one successful convoy route *will* cut the support given by a unit in the destination province that is trying to support an attack on a Fleet in an alternate route of that convoy. As long as there is one successful convoy route, the landing Army does cut any support given by a unit in the destination province. (Remember the rule: "Support is cut if the unit giving support is attacked from any province but the one where support is being given.") In Diagram 31, France wrote orders that would take its Army to Naples by either of two routes. The move from Tunis fails (because of a standoff with the Fleet in Naples), but it cuts the support of that Fleet because that Fleet is not cutting the successful convoy that came via the Ionian. Therefore, the Fleet in Rome stands off with the Fleet in the Tyrrhenian Sea.

DIAGRAM 31



FRANCE:
A Tun–Nap
F Tyn C A Tun–Nap
F Ion C A Tun–Nap

ITALY:
F Rom–Tyn
F Nap S F Rom–Tyn

In Diagram 32, The Fleet in Naples is dislodged by the combined strength of the Army being convoyed from Tunis and the Army in Apulia. Since the Army coming from Tunis can get to Naples via the Ionian, the Fleet in Naples was not supporting an attack against the Fleet that ultimately convoyed the Army, so its support was cut.

Note: If the Italian orders had been reversed, then the Fleet in the Tyrrhenian Sea would be dislodged and the Fleet in Naples would move to the Tyrrhenian Sea.

DIAGRAM 32

FRANCE:
A Tun–Nap
F Tyn C A Tun–Nap
F Ion C A Tun–Nap
A Apu S A Tun–Nap

ITALY:
F Rom–Tyn
F Nap S F Rom–Tyn

3. THE ORDER RESOLUTION PHASE

After all the orders have been revealed and read, the players (or an assigned gamemaster) must resolve all of the conflicts. Resolution will result in successful moves, failed moves, standoffs, retreats and disbandments. The units on the board are moved and removed as described in the next two phases of play.

4. RETREAT AND DISBANDING PHASE

After all the orders have been revealed and read, the moves made and the conflicts resolved, any dislodged (defeated) units make their retreat. These retreats are written down (just like orders) and are revealed at once. No diplomacy or discussion takes place prior to writing retreat orders – all countries are on their own.

A dislodged unit must retreat to an adjacent province that it could ordinarily move to if unopposed by other units. Sometimes a retreat is made deeper into enemy territory.

A unit cannot retreat to:

- a province which is occupied,
- the province from which the attacker came,
- a province that was left vacant by a standoff during the same turn.

If there is no available province to which to retreat, the dislodged unit is immediately disbanded and removed from the mapboard.

Writing Retreats

If two or more units must retreat, the retreat locations are immediately (and without discussion) written down by the players concerned. The written retreats are then simultaneously revealed. Retreats cannot be convoyed or supported. Each player should write down the location of the dislodged unit and the location to which it is retreating.

Disbandment

If two or more units are ordered to retreat to the same province, they all must be disbanded. If a player fails to order a retreat when necessary, the unit is disbanded. A unit can always voluntarily disband instead of retreating.

5. GAINING AND LOSING UNITS PHASE (AFTER FALL TURN)

Controlling Supply Centers

After each Fall turn, players check to see how many supply centers they control. A country controls a supply center when one of its units occupies that supply center province after a Fall turn has been played and completed.

Once a country gains control of a supply center, it can leave the center vacant and still keep control of it, as long as that center is not occupied by another country at the close of a Fall turn. A unit that moves into a supply center during a Spring turn and moves out of it during the Fall of the same year does not affect the ownership of the supply center. In short, a country retains control of a supply center as long as, at the end of each Fall turn (including retreats), the supply center is either vacant or is occupied by one of its own units.

Adjusting Number of Units

After each Fall turn (including retreats, if any), players adjust their units to match the number of supply centers they control. This may result in some units being disbanded (if the player has lost supply centers that year) or in some units being built (if the player has gained supply centers that year).

As with retreats, gaining and losing units (collectively known as "adjustments") are written and exposed simultaneously without discussion or diplomacy of any kind.

Disbanding

If a country has fewer centers than units, it must disband the excess number of units (owner's choice of which units).

Building

If a country has more supply centers than units, it can place new units in each unoccupied supply center of its home country that it still controls. It cannot build units in supply centers outside its home country.

Example: The French player can build units only in Paris, Brest, and Marseilles throughout the course of the game. However, if Marseilles was under Italy's control and the French player had a unit in Brest, he/she would only be allowed to build in Paris, no matter how many builds France was entitled to on that turn. If the French player vacated Brest and regained control of Marseilles, he/she would be allowed to build there after another Fall turn (provided he/she was still entitled to build on that turn).

Additional Building Rules

- Only an Army unit can be built on an inland province supply center.
- When building a unit on a coastal province supply center, a Fleet or Army must be specified in the written Build order. If Russia builds a Fleet in St. Petersburg, the Russian player must also specify "North Coast" or "South Coast."
- If your country's home supply centers are all occupied by your own (or other players') units, then you cannot build during the current Fall turn. Remember to leave some home supply centers open if you intend to build new units in the Fall.
- If your country has lost all of its home supply centers, you can still fight with the units (supplied by other centers) remaining under your control. In this case, you cannot build new units until you recapture a home supply center and control it at the close of a Fall turn.
- A country can decline to build a unit that it is entitled to for whatever reason (usually a diplomatic one).

Writing Builds and Disbandments

Players write down which units they will disband (if any) and what type of unit will be built in a home supply center (if any). These orders are written without diplomacy or discussion and revealed at the same time. Any vague or invalid orders are ignored.

TIME MANAGEMENT

It is wise to set aside about four hours to play Diplomacy. No more than five minutes should be allowed for writing orders after the diplomatic negotiation period has ended. Diplomacy and other conversation should not be allowed during the writing and reading of orders, between moves and retreats, during and after retreats, or during adjustments.

Newcomers should be given a half-hour (at least) introduction to the game *before* the other players assemble. A few moves should then be played with newcomers so they become familiar with the rules before the game starts.

CIVIL DISORDER

If you leave the game, or otherwise fail to submit orders on a given Spring or Fall turn, it is assumed that your government has collapsed. Your units all hold in position, but do not support each other. If they are dislodged, they are disbanded. No new units are raised for the country.

If a country in civil disorder has to remove units, the units farthest from the country are removed first. If units are equally distant, then remove Fleets before Armies and then in alphabetical order by the provinces in which they are located.

It is probably best, if enough players are present, to allow someone else to replace any player who leaves the game. Players should decide what policies they will follow before starting the game.

ALTERNATE WAY TO PLAY

The following is an alternative way that DIPLOMACY can be played when fewer than seven players are present.

Six Players: Eliminate Italy. Italian units hold in position and defend themselves, but do not support each other. Units belonging to any of the players can support them in their holding position. If Italian units are forced to retreat, they are disbanded.

Five Players: Eliminate Italy and Germany (as described for Italy above).

Four Players: One player plays England, and the other three play the following pairs: Austria/France, Germany/Turkey, Italy/Russia.

Three Players: One player controls England/Germany/Austria; the second, Russia/Italy; and the third, France/Turkey.

Two Players: This version can be played as a World War I simulation. One player controls England/France/Russia while the other plays Austria/Germany/Turkey. Italy is neutral and Italian territory cannot be entered. The game begins in 1914. Before the Fall 1914 adjustments, a coin is flipped. Italy joins the winner of the toss in Spring 1915. The first to control 24 Supply Centers wins. This is also an enjoyable way for two new players to learn the rules.

In games for 2, 3 or 4 players, supply center ownership is computed for each individual country, even though the same person plays more than one country. As with the regular rules, adjustments must be made by each country in accordance with its supply center holdings.

Opening Moves in a Sample Game

This sample game will help demonstrate some of the typical opening moves in a game of Diplomacy. This is intended to be a look at order writing and resolution. No strategies, diplomacy, alliances, or negotiations are discussed here. Besides, it would take too much space to record all of the juicy stuff that goes on!

As you read the orders, you may want set up the mapboard and move each playing piece so that it projects into the province to which it has been ordered. As soon as the final results are clear, the piece should be pushed into its new position or back to its old one.

SPRING 1901

Austria:	A Vie-Tri, A Bud-Gal, F Tri-Alb
England:	A Lvp-Yor, F Lon-Nth, F Edi-Nrg
France:	A Par-Bur, A Mar-Spa, F Bre-Pic
Germany:	A Ber-Kiel, A Mun-Ruhr, F Kiel-Den
Italy:	A Ven-Pie, A Rom-Ven, F Nap-Ion
Russia:	A Mos-Ukr, A War-Gal, F StP-Bot, F Sev-Bla
Turkey:	A Con-Bul, A Smy-Con, F Ank-Bla

Commentary: All orders succeed except for the two units ordered to the Black Sea and the two ordered to Galicia.

Key Rule: Units of equal strength trying to occupy the same province cause all those units to remain in their original provinces.

Retreats: None.

FALL 1901

Austria: A Tri Hold, A Bud-Ser, F Alb-Gre

England: A Yor-Nwy, F Nth C A Yor-Nwy, F Nrg-Bar

France: A Bur-Mar, A Spa-Port, F Pic-Bel

Germany: A Kiel-Hol, A Ruhr-Bel, F Den Holds

Italy: A Ven Holds, A Pie-Mar, F Ion-Tun

Russia: A Ukr S F Sev-Rum, A War-Gal, F Bot-Swe, F Sev-Rum

Turkey: A Bul-Ser, A Con-Bul, F Ank-Bla

Commentary: The units ordered to Belgium, Marseilles and Serbia do not move.
Key Rule: Units of equal strength trying to occupy the same province cause all those units to remain in their original provinces.

Commentary: The order "Con-Bul" also does not succeed.
Key Rule: One unit not moving can stop a unit or series of other units from moving.

Retreats: None.

Builds: Looking at the supply centers, England, Turkey, Austria, Italy, and France are each entitled to one build, and Russia and Germany are entitled to two. All players write down their builds and reveal their orders simultaneously. England builds a new Fleet in Edinburgh "F Edi." Germany builds "F Kiel" and "A Mun." Russia builds "A StP" and "A Sev." Turkey builds "A Smy." Austria builds "A Vie." Italy builds "F Nap." France builds "F Mar." France builds one unit for Portugal "A Por," but none for Spain, which its Army passed through during the Spring turn.

SPRING 1902

Austria: A Tri- Bud, A Vie-bud, A Bud-Ser, F Gre Holds

England: A Nwy-StP, F Nth-Nwy, F Bar S A Nwy-Stp, F Edi-Nth

France: A Bur S F Pic-Bel, A Port-Spa, F Pic-Bel, F Mar Holds

Germany: A Hol-Bel, A Ruh S A Hol-Bel, A Mun-Bur, F Den Holds, F Kiel-Hol

Italy: A Ven Holds, A Pie-Mar, F Tun-West, F Nap-Tyn

Russia: A Ukr S F Rum, A Gal-Bud, A StP-Nwy, A Sev S F Rum, F Swe S Stp-Nwy, F Rum Holds

Turkey: A Bul-Rum, A Con-Bul, A Smy-Arm, F Bla S Bul-Rum

Commentary: The Russian and English units on the Norway/St. Petersburg border fail to move.
Key Rules: (1) Units of equal strength trying to occupy the same province cause all those units to remain in their original provinces. (2) Units cannot trade places without the use of a convoy.

Commentary: This causes the English Fleets in the North Sea and Edinburgh to stay in place.
Key Rule: One unit not moving can stop a unit or series of units from moving.

Note that Sweden and Norway are adjacent along a coastline at the south, so the Fleet in Sweden can support an attack on Norway.

Commentary: The units ordered to Budapest also fail.
Key Rule: Units of equal strength trying to occupy the same province cause all those units to remain in their original provinces.

The Austrian Fleet in Greece could not support the move to Serbia. Because a Fleet cannot move to an inland province, it cannot provide support there.

The Turkish attack on Rumania, although supported, fails because Russia had more units supporting the Rumanian hold order (two-unit attack vs. a three-unit hold).

Commentary: Many other orders also fail, including the move by Turkey to Bulgaria, the Italian move into Marseilles, and the German move into Burgundy.
Key Rule: One unit not moving can stop a unit or series of units from moving.

Commentary: The German attack on Burgundy from Munich cuts the support there. This allows the supported German move into Belgium to succeed (two-unit attack vs. one-unit hold).
Key Rule: Support is cut if the unit giving support is attacked from any province except the one where support is being given.

Retreats: None.

FALL 1902

Austria:	A Vie-Gal, A Tri-Bud, A Ser S Turkish A Bul-Rum, F Gre Holds
England:	A Nwy-StP, F Bar S Nwy-StP, F Nth-Nwy, F Edi-Nth
France:	A Bur-Bel, F Pic S A Bur-Bel, A Spa S F Mar, F Mar S A Spa
Germany:	A Ruh-Bur, A Mun S Ruh-Bur, A Bel S Ruh-Bur, F Den-Swe, F Hol S A Bel
Italy:	A Ven-Pie, A Pie-Mar, F West-MAt, F Tyn-GoL
Russia:	A StP-Nwy, F Swe S StP-Nwy, F Rum S A Sev, A Sev S F Rum, A Gal S F Rum, A Ukr S A Sev
Turkey:	A Bul-Rum, A Con-Bul, A Arm-Sev, F Bla S A Bul-Rum

Commentary: First, look for support that has been cut. Many support orders written on this turn are cut because of the following rule.
Key Rule: Support is cut if the unit giving support is attacked from any province but the one where support is being given.

The supports that are cut include: the Russian Fleet in Sweden (the attack from Denmark), the French Fleet in Marseilles (the attack from Piedmont), the Russian Army in Sevastapol (the attack from Armenia), the Russian Army in Galacia (the attack from Vienna), and the Russian Fleet in Rumania (the attack from Bulgaria). The German Army in Belgium, supporting a move from Ruhr to Burgundy, does not have its support cut since the attack comes from Burgundy, the province where the support is being given.

Commentary: Next, look for standoffs. The Fleet in Marseilles and the Army in Sevastapol successfully standoff their attackers.
Key Rule: Units of equal strength trying to occupy the same province cause all those units to remain in their original provinces.

In fact, the support from Spain and Ukraine is unnecessary here as the individual units would have been enough to hold off the attack.

Commentary: The Army in Vienna can't get into occupied Galacia and the Army in Venice can't get into occupied Piedmont. They remain in place.
Key Rule: One unit not moving can stop a unit or series of units from moving.

Commentary: The French Army trying to get from Burgundy to Belgium fails because the support from Holland makes the forces equal.
Key Rule: Equal strength units trying to occupy the same province cause all those units to remain in their original provinces.

Commentary: If you look at the German Army coming from Ruhr, the support coming from Munich gives a strength of two to the French Army's strength of one. The Army in Ruhr moves into Burgundy and the French Army will have to retreat during the Retreat phase.

The Russian Fleet in Rumania was originally well supported enough to hold off the Turkish attack from Bulgaria. However, both its supports were cut and it now stands alone. This is not enough to hold off the attack, since Turkey is supporting the Bulgarian Army with the Fleet in the Black Sea. The Bulgarian Army moves into Rumania and the Russian Army there will have to retreat during the Retreat phase. The vacating of Bulgaria also allows the Army in Constantinople to enter Bulgaria.

Since the Russian support in Sweden was cut, the English attack from Norway into St. Petersburg succeeds. The Russian Army in St. Petersburg will have to retreat during the Retreat phase. Since the Army in Norway entered St. Petersburg, the other British Fleets can complete their moves.

Retreats: There are three units on the board that must retreat during the Retreat phase (one French, two Russian). The Russian unit in Rumania has no place to retreat (all adjacent territories are occupied) and is immediately disbanded and removed from the board. The Russian and French players then write down the retreat for their one unit. Russia: "A StP-Mos." France: "A Bur-Gas." The units are moved to Moscow and Gascony.

Builds and Disbandments: Russia controls four supply centers but has five units. It must disband one. All other players but Italy get a build. All players write down their builds and disbandments and reveal them simultaneously. Germany builds "F Kiel." Russia removes "A Gal." Turkey builds "F Smy." Austria builds "A Tri." France builds "A Par." England builds "F Lon." Italy does not change.

Conclusion: At this point, with all the neutral supply centers owned by one of seven countries, and some fledgling alliances and conflicts between the players, we will end our sample game. No effort has been made here to analyze the strategy or tactics of these fictitious players. A detailed look at the complexities, strategies and tactical moves of *DIPLOMACY* can be found on our web site: www.avalonhill.com

22 Rules to Help You Resolve Orders

Following is a handy list of rules needed to resolve orders and game play issues. If you are unable to resolve an issue using this listing, refer to the instructions and examples within this booklet for more detailed explanations.

1. All units have the same strength.

2. There can only be one unit in a province at a time.

3. Equal strength units trying to occupy the same province cause all those units to remain in their original provinces.

4. A standoff does not dislodge a unit already in the province where the standoff took place.

5. One unit not moving can stop a series of other units from moving.

6. Units cannot trade places without the use of a convoy.

7. Three or more units can rotate provinces during a turn provided none directly trade places.

8. A unit not ordered to move can be supported by a support order that only mentions its province.

9. A unit ordered to move can only be supported by a support order that matches the move the unit is trying to make.

10. A dislodged unit can still cause a standoff in a province different from the one that dislodged it.

11. A dislodged unit, even with support, has no effect on the province that dislodged it.

12. A country cannot dislodge or support the dislodgment of one of its own units, even if that dislodgment is unexpected.

13. Support is cut if the unit giving support is attacked from any province except the one where support is being given.

14. Support is cut if the supporting unit is dislodged.

15. A unit being dislodged by one province can still cut support in another.

16. An attack by a country on one of its own units does not cut support.

17. A dislodgment of a Fleet necessary to a convoy causes that convoy to fail.

18. A convoy that causes the convoyed Army to standoff at its destination results in that Army remaining in its original province.

19. Two units can exchange places if either or both are convoyed. (This is the exception to Rule 6.)

20. An Army convoyed using alternate convoy orders reaches its destination as long as at least one convoy route remains open.

21. A convoyed Army does not cut the support of a unit supporting an attack against one of the Fleets necessary for the Army to convoy. (This supersedes Rule 13.)

22. An Army with at least one successful convoy route will cut the support given by a unit in the destination province that is supporting an attack on a Fleet in an alternate route in that convoy. (This supersedes Rule 21.)

Abbreviations

Below is a list of commonly accepted abbreviations for the provinces on the DIPLOMACY map. You may devise your own abbreviations, but remember that abbreviations subject to different interpretations may result in the failure of an order.

AUSTRIA:	
Bohemia	Boh
Budapest	Bud
Galicia	Gal
Trieste	Tri
Tyrolia	Tyr
Vienna	Vie

ENGLAND:	
Clyde	Cly
Edinburgh	Edi
Liverpool	Lvp
London	Lon
Wales	Wal
Yorkshire	Yor

FRANCE:	
Brest	Bre
Burgundy	Bur
Gascony	Gas
Marseilles	Mar
Paris	Par
Picardy	Pic

GERMANY:	
Berlin	Ber
Kiel	Kie
Munich	Mun
Prussia	Pru
Ruhr	Ruh
Silesia	Sil

ITALY:	
Apulia	Apu
Naples	Nap
Piedmont	Pie
Rome	Rom
Tuscany	Tus
Venice	Ven

RUSSIA	
Livonia	Lvn
Moscow	Mos
Sevastopol	Sev
St. Petersburg	StP
Ukraine	Ukr
Warsaw	War

TURKEY:	
Ankara	Ank
Armenia	Arm
Constantinople	Con
Smyrna	Smy
Syria	Syr

NEUTRALS:	
Albania	Alb
Belgium	Bel
Bulgaria	Bul
Finland	Fin
Greece	Gre
Holland	Hol
Norway	Nwy
North Africa	NAf
Portugal	Por
Rumania	Rum
Serbia	Ser
Spain	Spa
Sweden	Swe
Tunis	Tun

BODIES OF WATER:	
Adriatic Sea	Adr
Aegean Sea	Aeg
Baltic Sea	Bal
Barents Sea	Bar
Black Sea	Bla
Eastern Mediterranean	Eas
English Channel	Eng
Gulf of Bothnia	Bot
Gulf of Lyon	GoL
Helgoland Bight	Hel
Ionian Sea	Ion
Irish Sea	Iri
Mid-Atlantic Ocean	Mid
North Atlantic Ocean	NAt
North Sea	Nth
Norwegian Sea	Nrg
Skagerrak	Ska
Tyrrhenian Sea	Tyn
Western Mediterranean	Wes

Questions

We will be happy to hear your questions or comments about this game.
Write to:

Avalon Hill c/o Hasbro Games,
Consumer Affairs Dept.,
P.O. Box 200, Pawtucket, RI. 02862.
Tel: 888-836-7025 (toll-free).
Customers in Canada may call 450-670-9820.

Check us out on the Web:
www.avalonhill.com
www.hasbro.com

the end