UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

PROGETTO DI "BASI DI DATI E SISTEMI INFORMATIVI"

DOCENTI:

Adriano Peron , Alessandro de Luca

STUDENTI:

Cuomo Daniele N86 1346 , Iervolino Riccardo N86 1608

PROGETTO:

Progetto n°2, prima parte: Archivio Risiko!

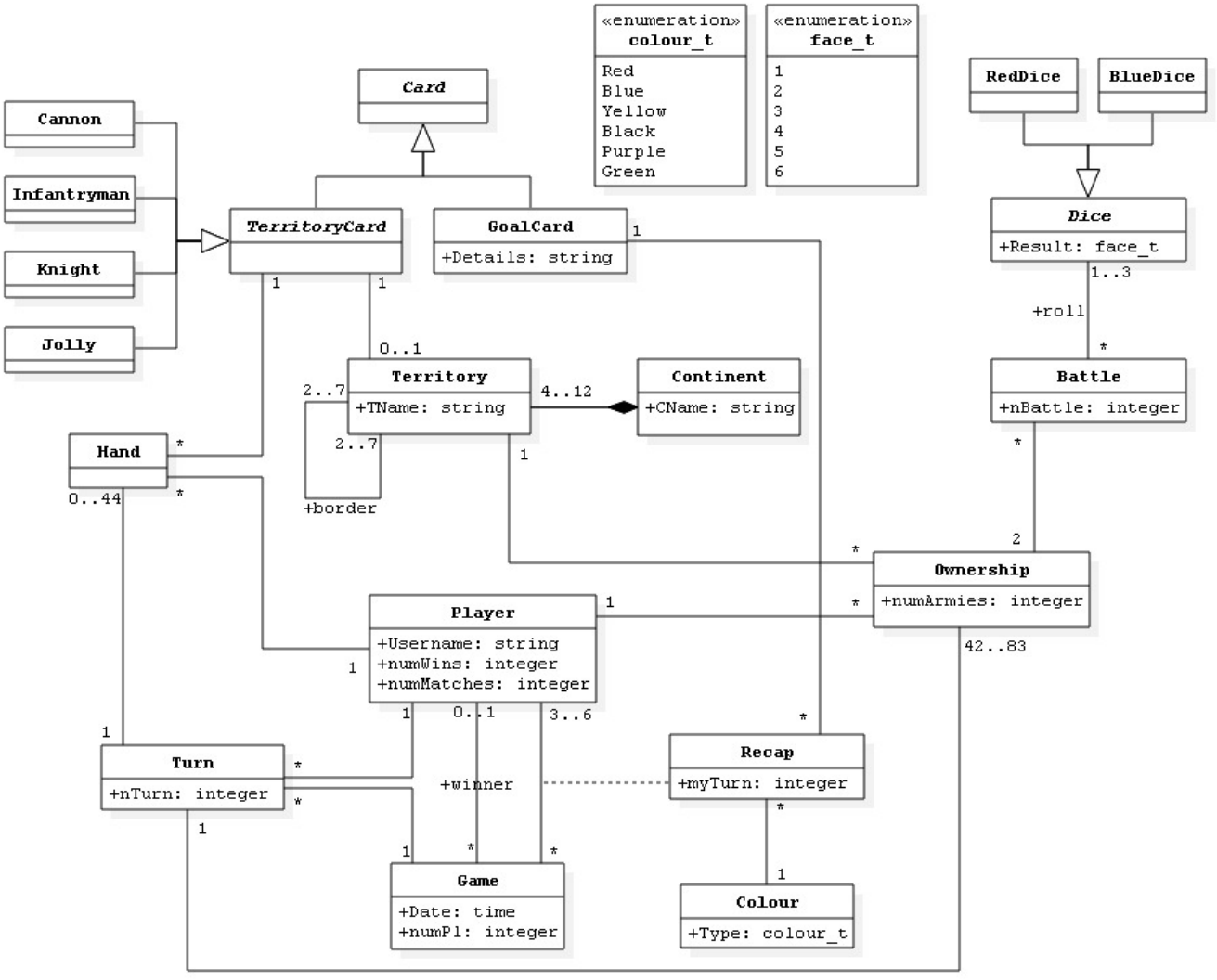
Class Diagram, CD ristrutturato, Dizionario classi e associazioni, Schema relazionale, Definizione tabelle

Descrizione progetto

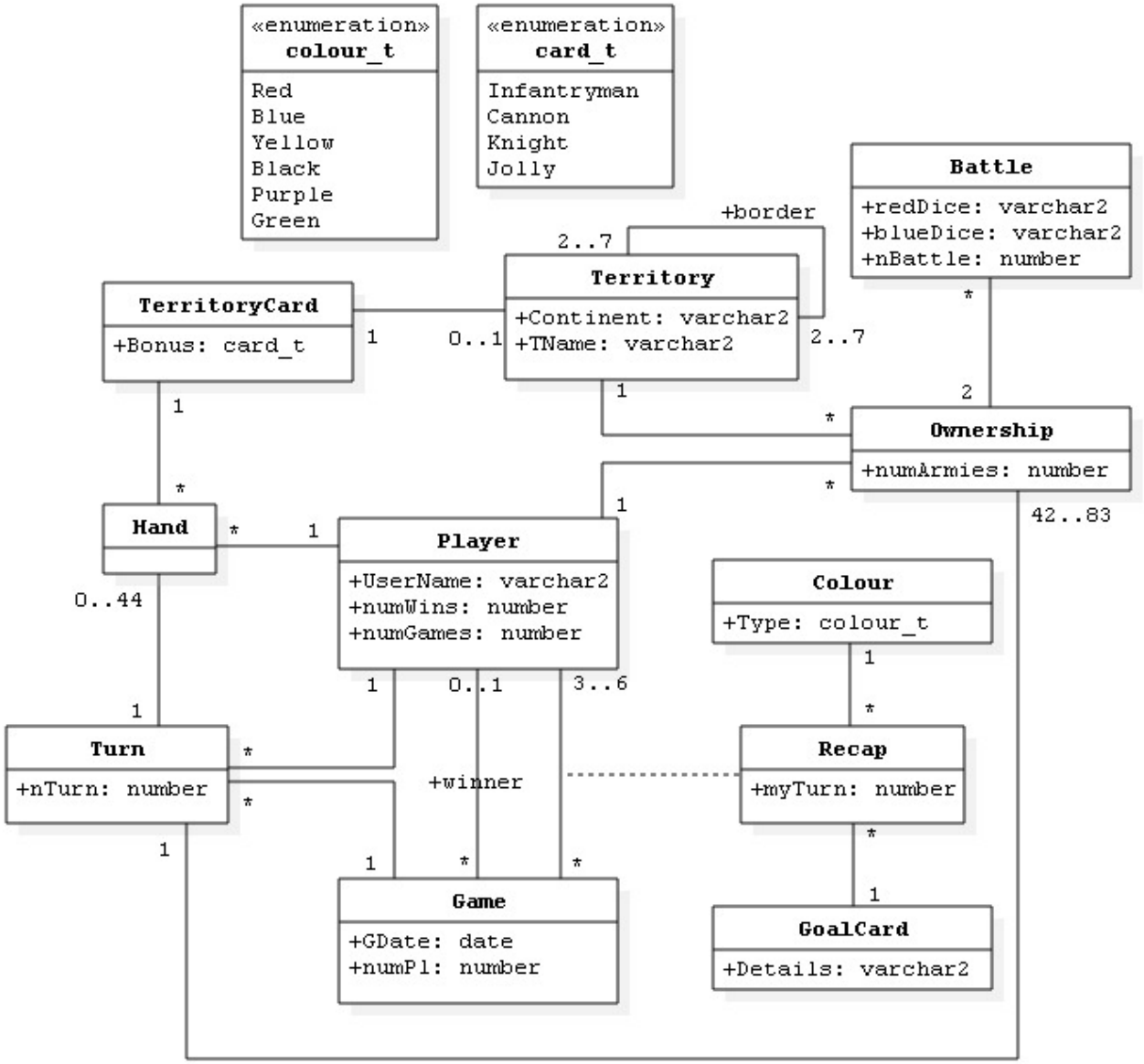
L'applicazione ArchivioRisiko! è finalizzata alla memorizzazione di partite del celebre gioco da tavolo attraverso l'inserimento in sequenza dei passaggi atomici di ogni fase, automatizzati laddove possibile (ad esempio per la rimozione delle armate da un territorio sconfitto). È offerta all'utente la possibilità di simulare la casualità dello stato iniziale di ogni partita, delle carte pescate (territorio e obiettivo) e del lancio dei dadi. Sono presenti nell'applicazione fin dal primo avvio le informazioni sulla plancia di gioco e sui mazzi di carte. Abbiamo scelto il turno come principale perno tra le relazioni, allo scopo di sfruttare la sua intrinseca proprietà di ordine temporale, grazie alla quale il sistema può facilmente individuare in modo univoco informazioni legate allo stato (quali ad esempio i possedimenti o le carte utili alle combinazioni).

I controlli delle modifiche avvengono attraverso trigger PL-SQL, mentre la gran parte degli inserimenti illegali sono impediti dalle procedure di inserimento dell'applicazione a livello Java. Essa è capace di gestire tutte le azioni legali del Risiko classico, servendosi di liste per la selezione rapida tra le sole scelte (sicuramente legali) a disposizione dell'utente.

Class Diagram



Class Diagram ristrutturato



Dizionario

- ***Classi***

- Player: Rappresenta i potenziali giocatori

Attributo	***Tipo***	***Descrizione***
Username	string	Nome col quale si identifica l'utente
numWins	integer	Totale vittorie del giocatore
numGames	integer	Totale partite giocate

- Game: Rappresenta le partite giocate, o in corso

Attributo	***Tipo***	***Descrizione***
GDate	date	Data inizio partita
numPl	integer	Numero giocatori

- Turn: Serve a conservare ogni signolo turno delle partite

Attributo	***Tipo***	***Descrizione***
nTurn	integer	n-esimo turno della partita

- Recap: Conserva le informazioni statiche di un giocatore in partita

Attributo	***Tipo***	***Descrizione***
numPl	integer	Numero giocatori

- Territory: Rappresenta i territori presenti sulla plancia di gioco

Attributo	***Tipo***	***Descrizione***
Continent	string	Continente di cui fa parte il territorio
TName	string	Nome nonché identificatore del territorio

-Battle: Descrive gli esiti dei singoli lanci ai dadi

Attributo	***Tipo***	***Descrizione***
blueDice	string	Lancio dei dadi blu nel formato "x,y,z"
redDice	string	Lancio dei dadi rossi nel formato "x,y,z"
nBattle	integer	Ordine delle battaglie tra due possedimenti

- Ownership: Serve a memorizzare chi possiede un territorio

Attributo	***Tipo***	***Descrizione***
numArmies	ineger	Armate presenti in quel possedimento

- Hand: Serve a memorizzare chi possiede una carta

Attributo	***Tipo***	***Descrizione***

- Territorycard: Rappresenta le 44 carte del gioco

Attributo	***Tipo***	***Descrizione***
Bonus	card_t	Identifica il tipo di bonus per i tris

- Colour: Rappresenta i 6 colori disponibili nel gioco

Attributo	***Tipo***	***Descrizione***
Type	colour_t	Identifica il colore, tra i 6 disponibili

- Goalcard Rappresenta i 18 obiettivi del gioco

Attributo	***Tipo***	***Descrizione***
Details	string	Fornisce per esteso l'obiettivo della carta

- ***Associazioni***

Player-Turn

Molteplicità	***Descrizione***
1 a *	Turno giocato da un giocatore

Winner

Molteplicità	***Descrizione***
0..1 a *	Vincitore della partita

Recap

Molteplicità	***Descrizione***
3..6 a *	Classe di associazione che conserva le informazioni statiche di una partita

Game-Turn

Molteplicità	***Descrizione***
1 a *	Descrive tutti i turni di una partita

Hand-Turn

Molteplicità	***Descrizione***
0..44 a 1	Tutte le carte in gioco in quel turno

Ownership-Turn

Molteplicità	***Descrizione***
42..83 a 1	Tutti i possedimenti di quel turno

Battle-Ownership

Molteplicità	***Descrizione***
* a 2	Una battaglia avviene tra due possedimenti e rappresenta un lancio di dadi

Hand-Player

Molteplicità	***Descrizione***
* a 1	Tutte le carte di un giocatore

OwnerShip-Player

Molteplicità	***Descrizione***
* a 1	Tutti i possedimenti di un giocatore

Territory-Ownership

Molteplicità	***Descrizione***
1 a *	Un singolo territorio appartiene a molti giocatori nell'intera base di dati

Border

Molteplicità	***Descrizione***
2..7 a 2..7	Rappresenta i territori confinanti con un singolo territorio

TerritoryCard-Territory

Molteplicità	***Descrizione***
0..1 a 1	Una carta rappresenta un territorio, eccetto i due jolly

TerritoryCard-Hand

Molteplicità	***Descrizione***
1 A *	Una singola carta è in mano a molti giocatori nell'intera base di dati

GoalCard-Recap

Molteplicità	***Descrizione***
1 a *	Un recap descrive l'obiettivo di un giocatore in una partita

Colour-Recap

Molteplicità	***Descrizione***
1 a *	Un recap descrive il colore di un giocatore in una partita

- ***Enumerazioni***

colour_t

Descrizione
Fornisce tutti e soli i nomi dei colori disponibili nel Risiko classico

card_t

Descrizione
Permette di identificare il tipo di bonus di una carta semplicemente osservandone l'attributo

Schema Relazionale

PLAYER(codPlayer , Username , numWins , numGames **)**

TURN(codTurn , nTurn , Player , Game **)**

GAME(codGame , GDate , numPl , Winner **)**

COLOUR(codClr , Type **)**

RECAP(Colour , myTurn , Goal , Player , Game **)**

GOALCARD(codGoal , Details **)**

TERRITORYCARD(codCard , Bonus , TName **)**

TERRITORY(TName , Continent **)**

BORDER(TName1 , TName2 **)**

BATTLE(Striker , redDice , Defender , blueDice , nBattle **)**

OWNERSHIP(codOwn , numArmies , TName , Owner , Turn **)**

HAND(Card , Player , Turn **)**

Definizione Tabelle

```
CREATE TABLE PLAYER(
  codPl          NUMBER          PRIMARY KEY,
  Username    VARCHAR2(50) UNIQUE,
  numWins      NUMBER,
  numGames   NUMBER );


CREATE TABLE GAME(
  codGm  NUMBER   PRIMARY KEY,
  GDate   DATE,
  numPl   NUMBER   CHECK(numPl > 1 AND numPl < 7),
  Winner  NUMBER
  );


CREATE TABLE RECAP(
  Game     NUMBER      NOT NULL,
  Player   NUMBER      NOT NULL,
  Colour   NUMBER      NOT NULL,
  Goal      NUMBER      NOT NULL,
  myTurn  NUMBER      NOT NULL
  );


CREATE TABLE GOALCARD(
  codGoal NUMBER           PRIMARY KEY,
  Goal       VARCHAR2(500) NOT NULL
  );


CREATE TABLE TURN(
  codTr   NUMBER  PRIMARY KEY,
  Round  NUMBER  NOT NULL,
  currPl  NUMBER  NOT NULL,
  Game   NUMBER  NOT NULL
  );


CREATE TABLE TERRITORYCARD(
  codCard NUMBER          PRIMARY KEY,
  Bonus_t  VARCHAR2(10)  NOT NULL,
  TName    VARCHAR2(20)
  );
```

```
CREATE TABLE TERRITORY(
 TName      VARCHAR2(20) PRIMARY KEY,
 Continent  VARCHAR2(10)  NOT NULL
 );


CREATE TABLE BORDER(
 TName1 VARCHAR2(20) NOT NULL,
 TName2 VARCHAR2(20) NOT NULL
 );


CREATE TABLE HAND(
 Turn      NUMBER  NOT NULL,
 Card      NUMBER  NOT NULL,
 Player    NUMBER  NOT NULL
 );


CREATE TABLE BATTLE(
 redDice    VARCHAR2(7)  NOT NULL,
 blueDice   VARCHAR2(7) NOT NULL,
 Striker     NUMBER         NOT NULL,
 Defender NUMBER          NOT NULL,
 nBattle     NUMBER         NOT NULL
 );


CREATE TABLE OWNERSHIP(
 codOwn      NUMBER           PRIMARY KEY,
 TName       VARCHAR2 (20) NOT NULL,
 Owner       NUMBER           NOT NULL,
 numArmed  NUMBER           NOT NULL,
 Turn          NUMBER           NOT NULL
 );

CREATE TABLE COLOUR(
 CODCL       NUMBER         PRIMARY KEY,
 COLOUR_T VARCHAR2(6)  NOT NULL
 );
```