Il progetto:

implementare una soluzione software per simulare un risiko non classico.

OGGETTI DEL GIOCO

classe Piece -> la classe piece deve implementare tutte le pedine e ha un membro “color”

classi figlie: Tank, Airplane, Submarine

classe WorldMapArea -> è una classe che ha almeno questi membri: il nome, il boolean CanBeConquered;

una lista ConfiningAreas (ognuna può essere sia di tipo SeaArea sia di tipo TerritoryArea)

SeaArea-> ha una list di Piece, tutti di tipo Submarine; ha gli stessi membri di WorldMapArea, ma CanBeConquered è per forza settato a False

TerritoryArea ->ha due list di Piece, una chiamata StrikerPieces, l’altra DefenderPieces (o OwnerPieces);

CanBeConquered è settato a True; può avere un Piece di tipo Flag, e questo deve essere unico.

Card -> ha un membro “color” e un membro “figure” o “action”

Deck -> sempre una list (?) - questa deve essere una list particolare, a lunghezza fissa, e i cui oggetti vengono rimossi da essa.

OGGETTI DA CREARE

Aree della mappa geografica.

Pedine.

FASI DEL GIOCO

1. Scelta del numero dei giocatori e del territorio da scartare nel caso di 5 giocatori.
2. Scelta dei colori da assegnare a ciascun giocatore (assegnazione random).
3. Estrazione carta dal mazzo e lettura del colore.
4. Scelta TerritoryArea. Durante la scelta territori i controlli devono passare da un giocatore all’altro e deve essere possibile mettere una sola bandierina su ciascun territorio, e una sola bandierina per ogni turno; si fanno tanti giri quanti sono i territori occupabli divisi per il numero di giocatori.
5. Estrazione carta dal mazzo e lettura del colore;
6. Il computer aggiorna i punti produzione di ogni player in ragione del numero di territori in possesso. Punti produzione = numero territori / 3 (approssima per difetto all’intero più grande minore del numero - floor)
7. Giro dei giocatori e richiesta di estrarre Card dal Deck, oppure di consegnare carte al Deck (scartare delle carte)
8. Giro dei giocatori, durante il quale ogni giocatore crea nuove armate e contestualmente al proprio turno, le sistema sui propri territori. Solo i propri territori si possono occupare con le armate. Metti un controllo sullo stack di carte in possesso di ciascun giocatore (per usare le carte speciali). Oppure abilita l’uso delle carte e quando le agganci col mouse dai la possibilità di trascinare solo alcuni tipi di carte. Nello stesso giro, dopo tutti gli spostamenti, viene eseguito un controllo su tutti i territori nemici e viene avviato un combattimento per ogni territorio nemico occupato. Per ogni giocatore, prima tutti gli spostamenti, poi tutti i combattimenti; i combattimenti vengono avviati solo se lo stack dei Piece nemici in alcuni stati nemici non è vuoto.
9. Combattimento. Viene avviato un metodo del singolo combattimento. Il metodo deve ricevere due riferimenti ai player che occupano il territorio. Deve decidere chi lancia i dadi per primo: li lancia per primo l’attaccante se ha almeno un aereo in più sul territorio in questione rispetto al difensore; altrimenti sempre il difensore. In seguito, si ha che il computer decide quanti dadi può usare il lanciatore. E’ il computer a deciderlo (per l’attaccante un dado in più per ogni Submarine sui mari che bagnano il territorio; per il difensore 2 dadi in più per la guarnigione). Il computer attiva un tasto che permette al giocatore di lanciare i dadi, e a fianco del quale (o dentro il quale) appare il numero di dadi che si possono lanciare. Il lanciatore elimina all’avversario tante armate quante le esplosioni; se il lanciatore è l’attaccante, allora la bandierina viene eliminata con una sola esplosione.

CLASSI DA CREARE

GameController -> deve contenere dei metodi riguardanti il flusso del gioco;

ha una lista di player;

ha una lista di territori;

ha il metodo Attack;

ha il metodo DecideTheFirstLauncher;

ha il metodo LaunchNDices;

PlayerController -> il controller ha dei membri che indicano lo stato del giocatore, e dei membri che ne modificano lo stato; rappresenta “model” e “controller” del giocatore insieme;