Effect

Cost

MoveEffect

InfluenceEffect

BlockEffect

AttackEffect

New MoveEffect(4)

moveEffect.execute();

attack and block effect requires a target. So…

Player first selects target from drop down in bottom – RANGED PHASE EnemyTarget

GameContext has EnemyTarget reference.

MultiEffect which has a move effect and terrain reduction effect.

BLLOD RITUAL is tricky

Effect is gain token…but you have a choice…

Effect has

Requirement[]

isFulfillable

Text

Label Basic or Stronger…

Card

2 effects

Name

That’s it

Effect has

Text

IsFulfillable

dropDowntext = “Basic”, “Stronger”

Card

2 effects

Several requirements

initBasicEffect()}{

return new SimpleEffect(getBasicText(),ActionType.Blah);

}

Boolean isBasicFulfillable()

Return Fulfillable.isfulfillable(ActionType type);

Or FulFillable.isFulfillable(ActionTypeChoice Requirement ts);

But they are the same

Requirement…

CardOption

Card Options are dependent on other options, mainly the effect selected. So maybe it should be its own category.

Should the requirement or the Card be responsible for attributing blocks and statuses?

Not sure if I will use table or list. Problem with list is that you can’t do two columns which is what I need for label

Cardview

Classes

Hand contains Cards

Card

Card factory – populates cards??

CardProvider.getCards.

Card.selectEffect();

Card.selectMana();

Card.selectOtherRequirement();

Card.getEffects();

Game.getAvailableMana(‘string color);

Game

Properties

Source

Source.getAvailableMana();

Source.getAvailableMana(stiring color);

Player

Player.useSourceMana(string color);

MapView

Bottom left – directional arrows for movement during Perform movement phase.

Bottom right – Points generated

Middle. Next phase button.

Cards: individual classes?

SimpleCards can be generated by factory. Then we have some complicated ones that involve rerolling source. So they effect game state.

Has a call back method for game type.

registerListener;

if card has listener call card listener.

For example sourceController gets cardFactory then asks to register itself for any new X cards created.

Card.generateEffect calls all its listeners then generates th effect.

Card c = new Card();

c.setBasicText();

c.setAdvancedText();

c.setBasicGeneration();

c.setAdvancedGeneration();

c.setImage();

We use this technique to copydatabase:

<http://www.reigndesign.com/blog/using-your-own-sqlite-database-in-android-applications/>

So to do at home is to create the SQLite database.

ICardListener

.isInterestedIncard(string cardId);

Or Card

Card has CardType; CardNumber;

CardType.Spell

CardType.Artifact

Card is split into Wound and Non-Wound;

NonWoundCard

Database would have

Card Table

Type; Spell artefact etc…

Basic Effect, Powered Effect.

Basic Mana, Powered Mana??

Name

MoveGenCon ICardListener;

AttackGenController

Listens to all types of attack cards played by user;

Keeps track of attack points generated by user;

BlockGenController

In Combat view, jquery controls whether or not the generation is enough to kill enemy. If yes, then HandController should be notified which cards are now in ‘Spent’ state. Can’t do this because a card could be half spent. Just notify the controllers what has been spent.

ICardController has onCardUsed() event. Listeners remove them generation when this event called.

In movement gen phase we have card view;map view/enemy view. Enemy view shows drop down and token of enemies on the selected tile. In map view the sidebar should show enemy pictures.

Need to populate some card objects. Either put shit all in code or in java.

Action:  
Basic effect text

Powered effect

Mana required:

Image

Use the jquery load function to replace stuff.

Some also have USeWiths e.g. discards.

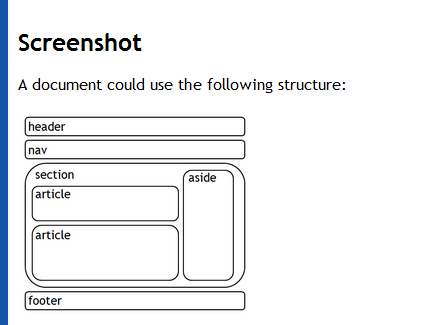
Then basic effect generation.

Powered effect generation.

Card page:  
This is going to be hard. So far can display images on the right.

Movement page:  
Play cards in hand

Map page:  
Simple has map and buttons to control movement.



Variant Summary:  
Remove hexagons till starting hand is 6.

Make a pool of skill cards = [players + 1. Players choose 1 skill in reverse initiative order. Place on top of deck after players draw cards for hand i.e. it should be the first card drawn on the second turn.

Core tiles can only be placed adjacent to two tiles.

Influence can be used to buy magic and advanced actions at any time at any location. However, if you use influence you may not attack.

Ignore night and no unit rules in underground locations.

Hidden scoring variant:  
Setup:  
Remove hexagons until players have starting hand of 6.

Draw advanced action cards = no. players + 1.

From last to first turn, choose adv action cards. Each player draws 2 skills and picks 1 skill.

On defeating enemy, keep the token. Fame is counted at the **end of the game**. Use marker to increase fame in special conditions like fame on cards.

End of each round, Draw advanced actions cards as at beginning. Last player chooses first. Each players draws 2 skills and picks 1. Remove another hexagon.

Faster variant:  
Play 3 rounds. Magic and advanced actions can be bought anywhere. Remove advanced action whose only special ability is to buy AA from anywhere.

Review:  
Really great variant. Close scoring from first and last place. Some thoughts on other variations:

When switching to new day/night time board, all dice should be consumable.

Day time start: Re reroll all black dice

Player option to end day time when there is only one basic colour dice.

Night time start: Reroll all gold dice. Majority of dice should be non-black.

Player option to end night time when there is only one basic colour dice.

On round end, the board is not flipped.

Review:  
Awesome.

Some more configs: Core tiles – must be placed adjacent to two tiles.

Other variants:  
Player hand limit reduced for higher initiative. Good variant.

Crap variant: Decks are recycled so need to think carefully about how to spend cards: Mkakes the game boring and too easy. Much faster though.