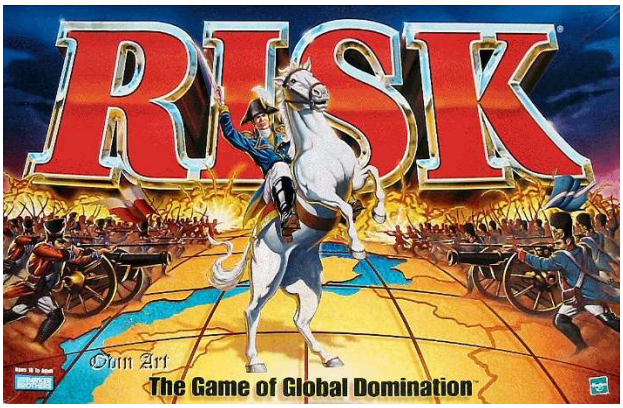
WORLD CONQUEST SPRINT III DESIGN DOCUMET

A Design Document for the Game ‘World Conquest’ for Raffle Games



FEBRUARY 8, 1993

TEAM ONE
University of Sussex

Contents

Sprint III

Design Objectives

For this sprint, our implementation aims are:

User

- [U1] Players must be able to **select a colour** at the beginning of the game.
- [U2] Players must be able to **trade in** matched sets of cards to take in additional armies.
- [U6] Players must be able to **claim territories** at the following phases:
    - Initial phase
    - Attack phase
- [U7] Players must be able to **shuffle** the card pack.
- [U8] Players must be able to **win the game** by capturing all 42 territories.
- [U13] Players must be able to **see all information regarding other players on the board** Including:
    - Player turn
    - Each player's colour

Functional

- [F1] The system/software will require at least one **autonomous computer player agent**.
- [F2] The system/software must have the capability of being **played by one person**
- [F4] The system/software should have a game player **agent who can respectively play the game**.
    - The game player agent should be able to play the game to the same extent as a human player.
- [F5] System/software must have a means of **uploading initial data**
- [F7] The system/software must have a mechanism that allows **players to select a colour**
- [F8] The system/software must have a **dice-rolling simulation** that will be used for:
    - Determine the starting player (whoever gets the highest roll starts the game)
    - Attacking territories
    - 2 red and 3 white die
- [F14] The system/software must have a **software representation of a deck of 56 Risk cards (draw pile).**
    - The placement of the draw pile should not obstruct any other features of the game.
- [F15] The system must simulate the **shuffling of the deck**.
- [F16] The system/software must simulate **trade in simulation**.
    - Trainers should be able to trade in sets of cards for additional armies.
- [F18] The system/software must have a mechanism that simulates **Risk Card rewards**. Example:
    - award a risk card if a territory has been captured, then declare the end of the player's turn.
- [F19] The system/software must have **3 representations of Armies:**
    - Infantry
    - Cavalry
    - Artillery
- [F23] The system/software must have a **chart reference point**.

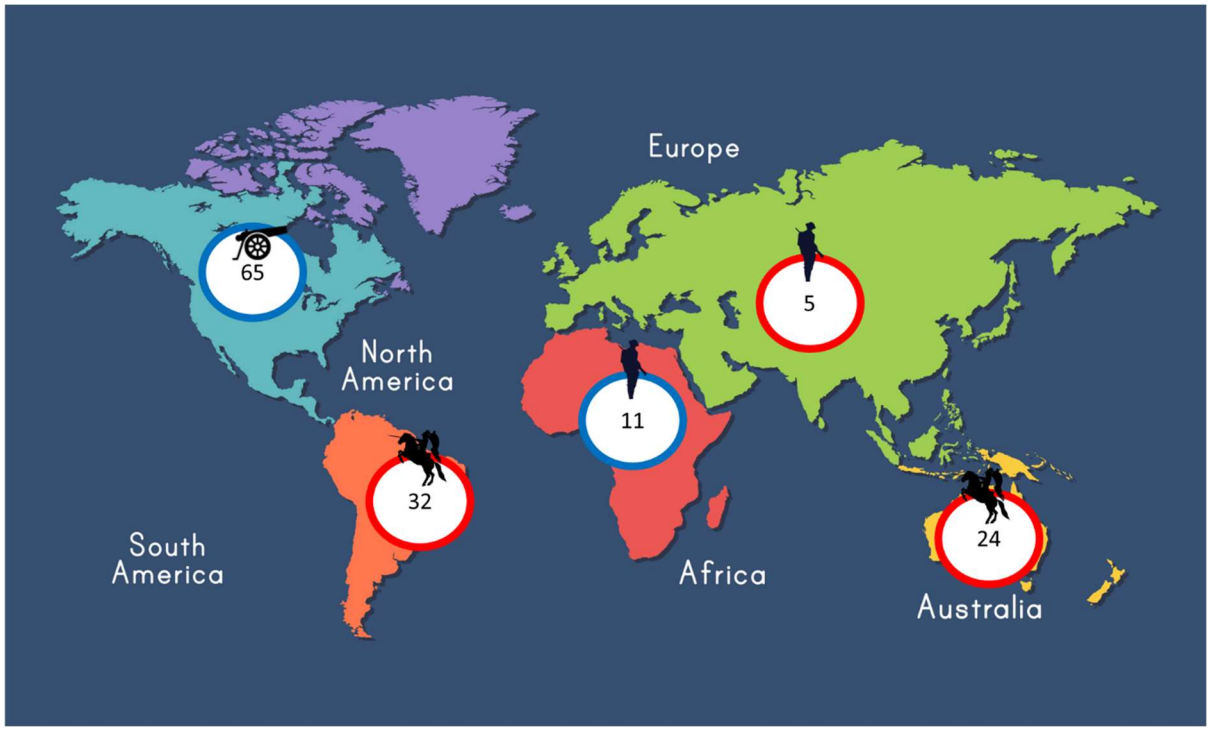
- shows players on the left side of the board game how many armies they will receive for each continent.

Domain

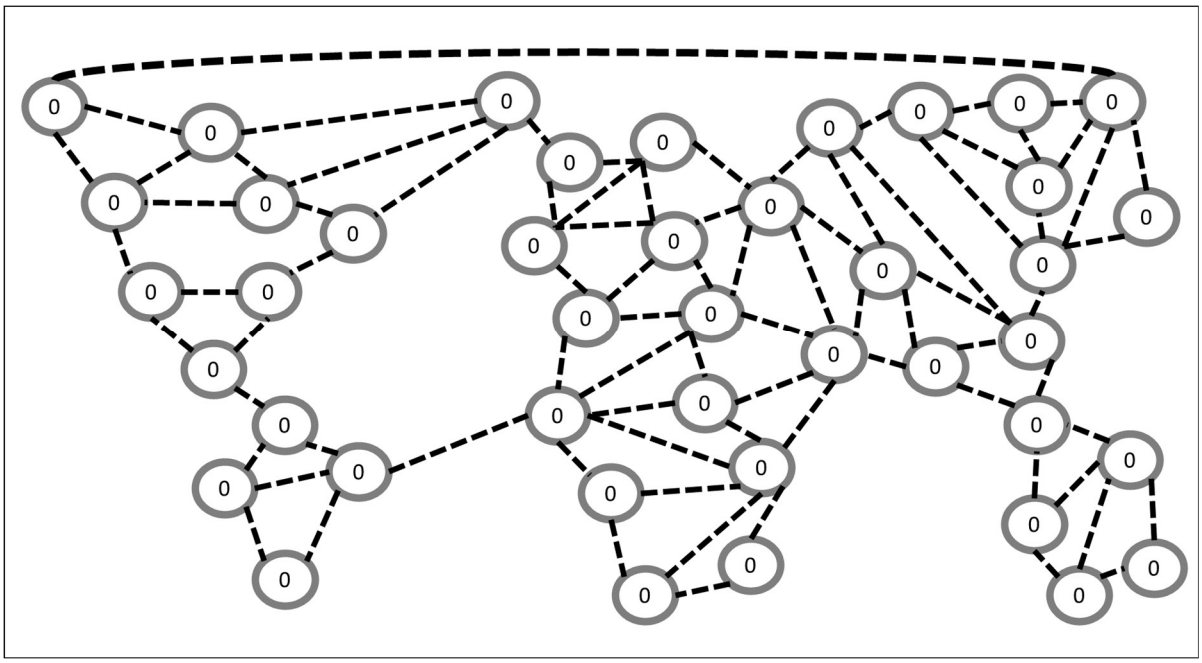
- [D1] The system/software may benefit from a title screen
- [D2] The system/software may need an intro screen

UI Design

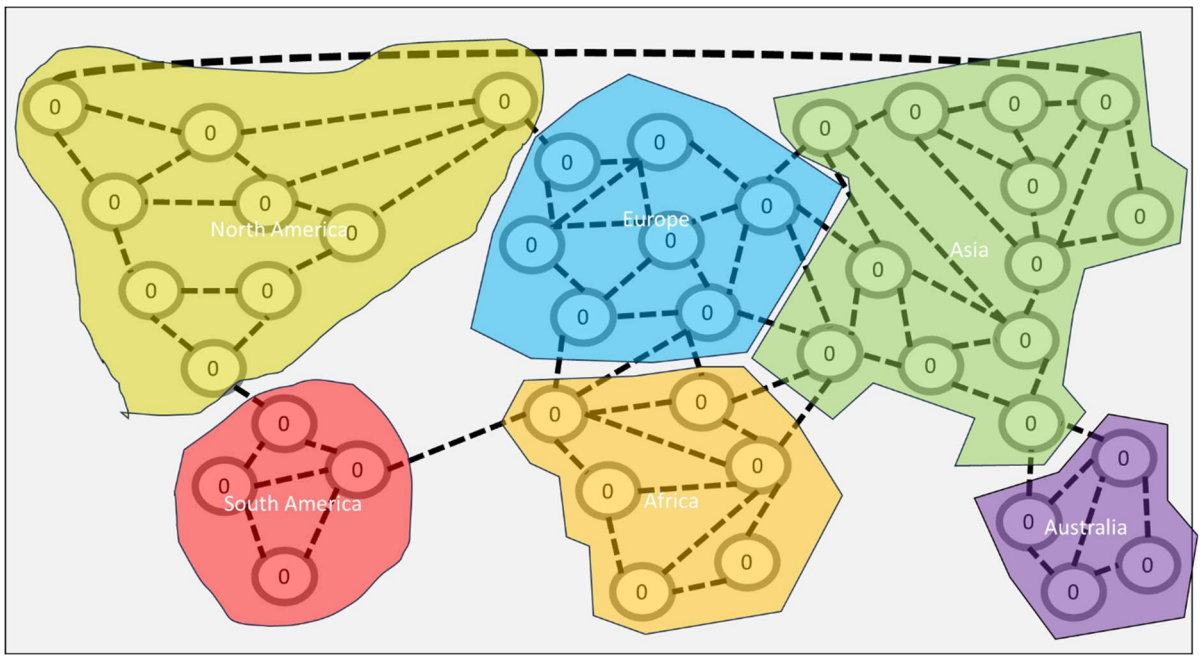
The UI design started with our weekly meetings where we described what we wanted our games map layout to functions. We fell appoint the approach of representing the conquered territories as circles – with a silhouette of the infantryman, cavalry or artillery to be an visual display of home many troops were occupying a territory visually



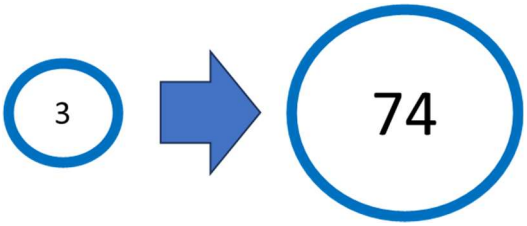
UI concept art – using google images(Artillery: https://commons.m.wikimedia.org/wiki/File:Cannon-symbol.svg, Cavalry: https://www.carstickers.com/products/stickers/medieval-and-mythical-stickers/knight-castle-stickers/details/warrior-on-horse-with-sword-sticker-9627/, infantry: https://clipground.com/ww1-soldier-silhouette-clipart.html, Map: https://static.vecteezy.com/system/resources/previews/000/406/567/original/vector-illustration-of-world-map-isolated.jpg)



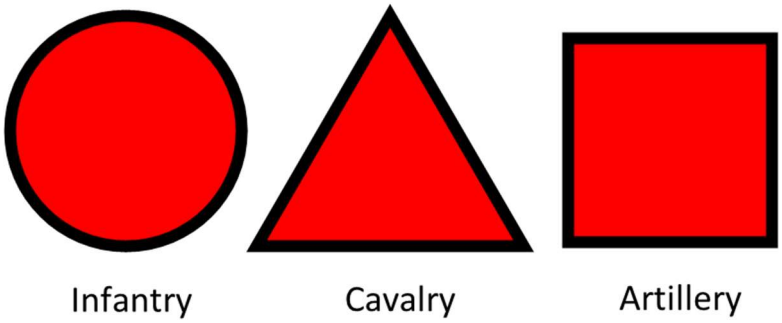
The above illustrates the node structure required for the full 42 territories (the above displays this territory as unassigned by a player, as if at the start of the game). Below is how this random bag of nodes is collected into continents of the board game.



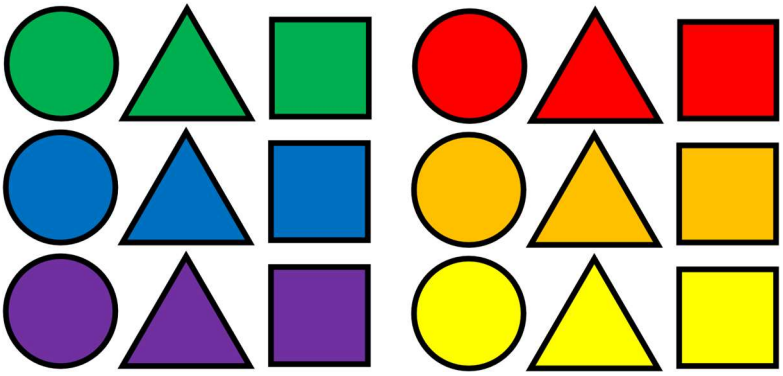
Now moving onto the nodes themselves – we decided another way to visually represent the number of troops in a territory was to have the node increase in size as the number of troops increased, thereby displaying how well a player is doing visually by that player's territory occupying more of the screen (in more than one way).



The icons for the troops I have simplified into basic shapes for now – but in later implementations, these will take the form of silhouettes of what they should represent.



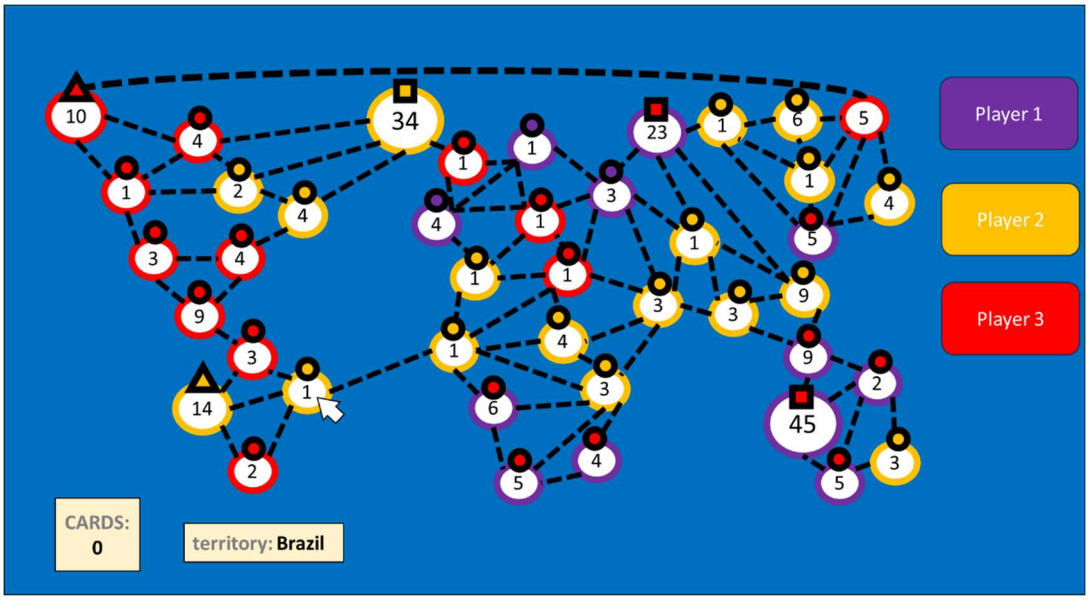
Pieces will be represented by PNG shapes that will be superimposed onto the territory as an extra visual representation of the number of troops on a given territory. So, a way this can be done can be that: from 1 to 9 there is an infantry icon on top of the territory troop number, 10 to 19 a cavalry icon and from 20 onwards the artillery icon; this does not need to be the implementation for this, coders can play around with numbers ranges that give the best visual signifier, as well as user feedback.



An additional aesthetic choice would be to have the troop pieces be the same colour as the player – but this can also be substituted by using a silhouette (such as in the first draft example UI concept art).

A title screen mock-up

This this sprint has in its requirements a title screen [D1] – I have created a very basic title screen that can be further improved at a later stage but is here as a stand in for functional purposes



A rough UI draft of what this sprint UI will look like

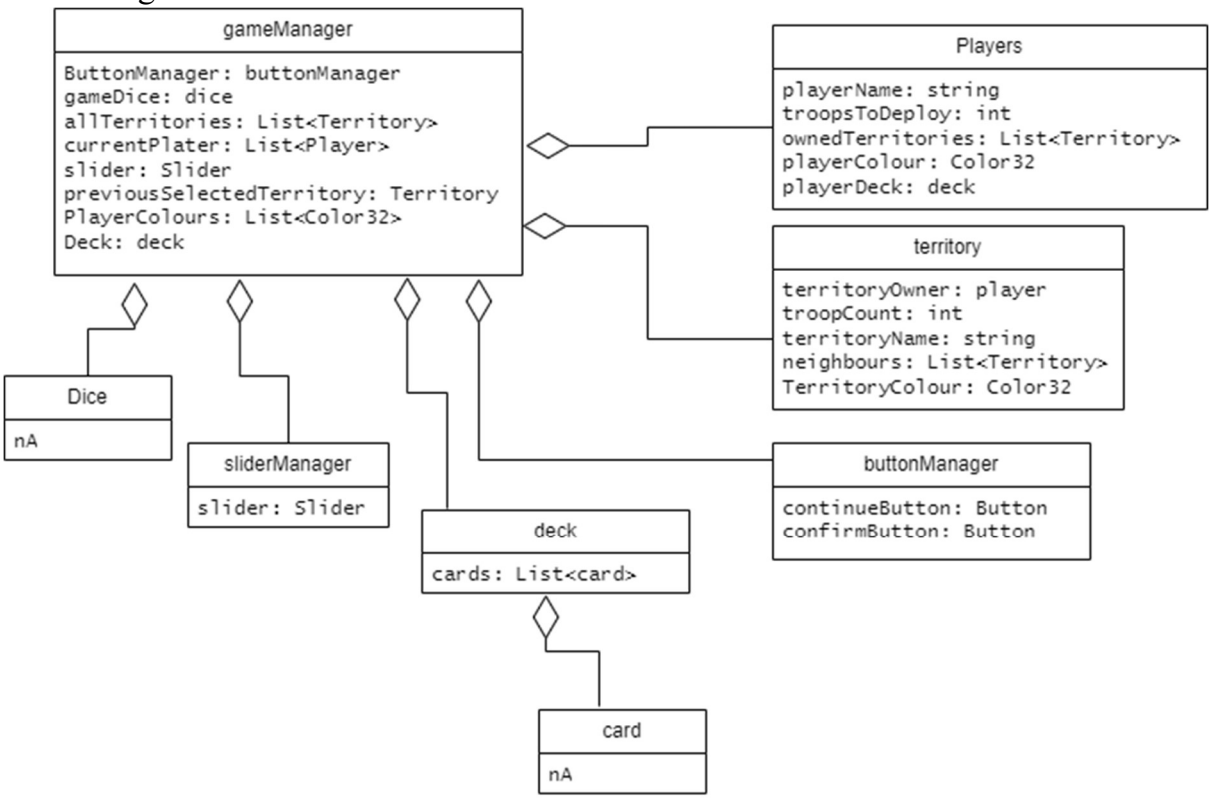
And finally, we have a vary rough final UI that this sprints prototype should look like. There can be changes to the placement to the placement of certain entitles (Such as the player icon) but all the elements illustrated above should be in the final visible.

Class table

Class No.	Class Name	Attributes	Comments
1	player	• `playerName` • `troopsToDeploy` • `ownedTerritories: List<Territory>` • `playerColour: Color32` • `playerDeck: deck`	• The player is now given the capability to hold cards. • It has a deck assigned to each player.

			• Players can select their colour
2	territory	• `territoryOwner` • `troopCount` • `territoryName` • `neighbours: List<Territory>` • `territoryColour: Color32`	• No significant changes
3	ButtonManager	• `continueButton` • `confirmButton: Button`	• No significant changes
4	GameManager	• `buttonManager` • `gameDice` • `allTerritories: List<Territory>` • `currentPlayers: List<Player>` • `slider: Slider` • `previousSelectedTerritory: Territory` • `PlayerColours: List<Color32>` • `Deck: deck`	• There are 42 territories to manage • A title screen/intro will be implemented in the Game Manager at the start of the game.
5	Dice	*`No attributes`*	• No significant changes
6	SliderManager	• `slider: Slider`	• No significant changes
7	card	*`No Attributes`*	• An empty class mean to represent the risk cards. • May have properties and types depending on the implementation
8	deck	• `cards:list<card>`	• ‘deck’ is a list of cards • Is used in the player to store the cards a user earns • Is used in the gameManager to store all possible cards used in the game. (there are a total of 56 cards)

UML Diagrams



Updated Design/Design Changes

Type	Design	Implementations

Conclusion

[Analysis of Design and how it related to Implementation]

References

Parker Brothers, 1993. *RISK - The World Conquest Game,* Beverly: Tonka Corporation.