

**Gruppe 9**

Anders Hvidgaard Poder, 19951439

Elund Christensen, 20074530

Kewin Peltonen, 20054669

Programming Project

AlphaTargui

1.0 D

07-04-2008



1 (8)

**Programming Project**

AlphaTargui

## Gruppe 9

Anders Hvidgaard Poder, 19951439

Elund Christensen, 20074530

Kewin Peltonen, 20054669

Programming Project

AlphaTargui

1.0 D

07-04-2008



2 (8)

## Contents

<b>1</b>	<b>Introduction.....</b>	<b>3</b>
<b>2</b>	<b>CRC-cards .....</b>	<b>4</b>
<b>3</b>	<b>Test-list.....</b>	<b>7</b>
3.1	Section .....	8
3.2	Section .....	8
<b>4</b>	<b>TODO .....</b>	<b>8</b>

## Gruppe 9

Anders Hvidgaard Poder, 19951439

Elund Christensen, 20074530

Kewin Peltonen, 20054669

Programming Project

AlphaTargui

1.0 D

07-04-2008



3 (8)

1

## Introduction

Text

## 2 CRC-cards

- 1) Make AlphaGame CRC – determine responsibilities.

AlphaGame (1)	Collaborators
Responsibilities: <ul style="list-style-type: none"> <li>• Maintain state of 4 players</li> <li>• Who is current player?</li> <li>• Board state (maintain)</li> <li>• Turn state</li> <li>• Round state</li> <li>• Determine winner</li> <li>• Handle attack</li> <li>• Validate move</li> <li>• Calculate revenue</li> <li>• Distribute revenue</li> <li>• Create board state</li> </ul>	<ul style="list-style-type: none"> <li>• Unit Move Tool</li> <li>• Targui Drawing</li> </ul>

- 2) Too much responsibility in the AlphaGame. "Board" and "Player" separated out from AlphaGame because of complexity. Used OO-centered thinking – "The physical model is a simulation of the real world".

Player	Collaborators
Responsibilities: <ul style="list-style-type: none"> <li>- know color (own)</li> <li>- know number of coins in treasury</li> </ul>	(player color)

Board	Collaborators
Responsibilities : <ul style="list-style-type: none"> <li>- Know tiles on board</li> <li>- know players (4)</li> <li>- <del>know current player</del></li> <li>- validate move, determine if attack move</li> </ul>	Tile Player

3) Distribute AlphaGame responsibilities onto the new collaborators

- State of the board → Delegated to "Board".
- Validate move → Delegated to "Board" (has knowledge of tiles and players, can determine if move is valid).
- Who is current player → "Board" know players.

4) Introduce "Tile".

Tile	Collaborators
Responsibilities: <ul style="list-style-type: none"> <li>- Know type</li> <li>- know position on board</li> <li>- know owner (if any)</li> <li>- know number of units on tile (if any)</li> </ul>	Tile type (Position) (Player color)

5) Move "currentPlayer()" back to "AlphaGame" from "Board".

6) AlphaGame CRC update.

AlphaGame (2)	Collaborators
Respons: <ul style="list-style-type: none"><li>- who is current player</li><li>- turn state</li><li>- round state</li><li>- determine winner</li><li>- handle attack</li><li>- calculate revenue</li><li>- distribute revenue</li><li>- create board state</li><li>- who is next player (new)</li></ul>	<ul style="list-style-type: none"><li>• Unit Move Tool</li><li>• Targui Drawing</li><li>• Board</li><li>• Player</li><li>• Tile</li></ul>

### 3 Test-list

The following section contains a test-list, who describes the tests we have identified:

- Player count (obvious - hardcoded).
- Tile positions matches specification (review manually).
- Round (red first, green second, blue third, yellow fourth, red first).
- Turn (expect move first, expect buy second).
- Move
  - Valid: Move to un-occupied tile.
  - Valid: Move to occupied tile where there is no camels.
  - Valid: Move to each of the tile types with no camels.
  - Invalid: Move with other player camels.
  - Invalid: Move without camels.
  - Invalid: Move to the "Salt Lake".
  - Invalid: Move outside the board (cannot be tested as the GUI is unable to make this move. Preconditions should be noted, so this test can be avoided).
  - Invalid: Move more than one tile in a turn.
  - Attack: Attacker has less camels than defender (attacker is defeated).
  - Attack: Attacker and defender has equal numbers of camels (status quo).
  - Attack: Attacker has more camels than defender (defender is defeated).
- Revenues
  - Calculate revenue for all players.
  - Calculate revenue, where one or more players have died.
  - Calculate revenue, where one or more players have lost own settlement.
  - Sums revenue correctly.

**Assumptions:** We assume that having a settlement is enough to get revenue compared to having own settlement.

- Determine winner: Play 25 rounds, one must own “Salt Mine”, determine winner.

### 3.1 Section

Text

### 3.2 Section

Text

## 4 TODO

This section contains our TODO's:

- Make CRC-cards to place responsibility.
- Test-list
- Template.
- Class diagram for AlphaTargui.
- Sequence diagram for “Moves”.
- Software Quality Attributes for the implementation.
- Start test-driven development.