

**Faculty of Informatics and Computer Science**

Information System/Computer Networks/Software Engineering/ Computer Science

**Estimation Card Game Artificial Intelligence**

By: Ahmed Hatem Tageldin

Supervised By

Dr. Ayman Ghoneim

Abstract

Estimation is a popular trick-taking card game. There are several applications on the market. However, as most of them focus mainly on the multiplayer component of the game, the Artificial Intelligence component is sometimes neglected. The objective of this project is to create an Estimation Artificial Intelligence system that is capable of simulating the expertise and unpredictability of a real human player. There are other related works similar to Estimation, such as Whist. Whist is an English trick-taking card game, however it is internationally known. Therefore, resources are more readily available for it. Instead of starting from scratch when building the Artificial Intelligence system, similar existing card games can be analysed and their Artificial Intelligence may be extracted. It can be used as a guide or even a template for further refinements.

Attestation &Turnitin Report

I understand the nature of plagiarism, and I am aware of the University’s policy on this.

I certify that this report reports original work by me during my University project except for the following (*adjust according to the circumstances*):

* The technology review in Section 2.5 was largely taken from [17].

**Signature** *Ahmad Hatem* **Date** 15/12/2016

Contents

[Abstract ii](#_Toc484030627)

[Attestation &Turnitin Report iii](#_Toc484030628)

[1 Introduction 1](#_Toc484030629)

[1.1 Overview 1](#_Toc484030630)

[1.2 Problem Statement 1](#_Toc484030631)

[1.3 Scope and Objectives 1](#_Toc484030632)

[1.4 Report Organization (Structure) 1](#_Toc484030633)

[1.5 Work Methodology 1](#_Toc484030634)

[1.6 Work Plan (Gantt Chart) 2](#_Toc484030635)

[2 Estimation Game Logistics 5](#_Toc484030636)

[2.1 Rules 5](#_Toc484030637)

[2.2 Card Suits 5](#_Toc484030638)

[2.3 Game Structure 5](#_Toc484030639)

[2.4 Round Structure 5](#_Toc484030640)

[2.4.1 Bidding 5](#_Toc484030641)

[2.4.2 Gameplay Flow 6](#_Toc484030642)

[2.4.3 Exceptions 6](#_Toc484030643)

[2.5 Scoring 6](#_Toc484030644)

[3 Related Work (State-of-The-Art) 7](#_Toc484030645)

[3.1 Background 7](#_Toc484030646)

[3.2 Literature Survey & Analysis 7](#_Toc484030647)

[4 Proposed solution 9](#_Toc484030648)

[4.1 Solution Methodology 9](#_Toc484030649)

[5 Implementation 10](#_Toc484030650)

[6 Testing and Evaluation 11](#_Toc484030651)

[7 Conclusions and Changes 12](#_Toc484030652)

[7.1 Summary 12](#_Toc484030653)

[References 13](#_Toc484030654)

# Introduction

## Overview

Estimation is a four-player trick-taking card game. It is an Arab variation of the classic English card game Whist. The objective of the game is to estimate the amount of tricks, given a hand of thirteen cards. Each player plays one card in each hand. Whoever wins the hand collects a trick. There are thirteen tricks in each round. The person with the highest score at the end of eighteen rounds is the winner. The aim of the project is to create a playable Estimation game and simulate an artificially intelligent single player computer bot.

## Problem Statement

There are several Estimation card game applications on the market. Most applications focus their efforts on the online multiplayer side of the game, thus neglecting the single player side. However, most of them do not contain an Artificial Intelligence component that an experienced player would find sufficiently challenging.

## [Scope](http://www.cs.stir.ac.uk/~kjt/research/conformed.html) and Objectives

The objective of the project is to create an Artificial Intelligence component for the card game Estimation. By combining expertise in playing the game and passion in Java programming, the expected outcome is an Artificial Intelligence component that can simulate the thought process and the unpredictability of an experienced human player. The main expected deliverable is a Java application. However, seeing as Android development is based on Java, an Android application may be delivered as an extension deliverable if the project can be sufficiently completed in due time.

## Report Organization (Structure)

Section 2 covers the related works, their analysis and their potential usefulness to the project. Section 3 covers the proposed solution of the project, as to how the problem will be approached. Section 4 concludes the report with a summary of all the previous sections.

## Work Methodology

Plan A is to find projects fitting the same criteria and learn from their structure, their source codes and concepts. They can be used as guides. If such projects can be found, it will eliminate the need to start from scratch. For instance, other Estimation applications.

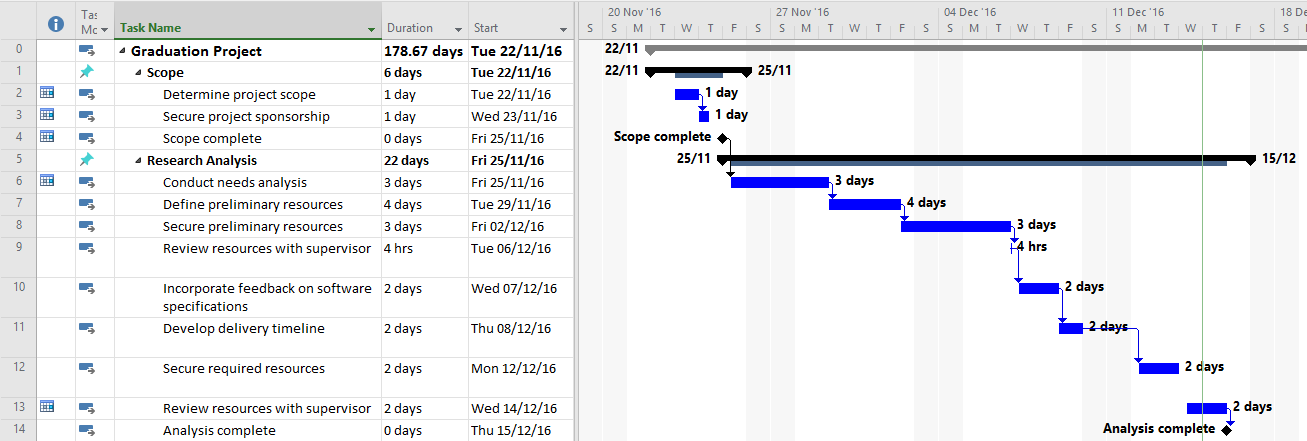
Plan B is to find projects fitting similar criteria and learn from their structure, their source codes and concepts. Although not ideal, it will provide a foundation on which to work from. For instance, variations of trick-taking games such as Whist.

Plan C is to find projects that share the same common basic attributes as Estimation. For instance, any card game application such as Poker, Hearts, Spades, etc.

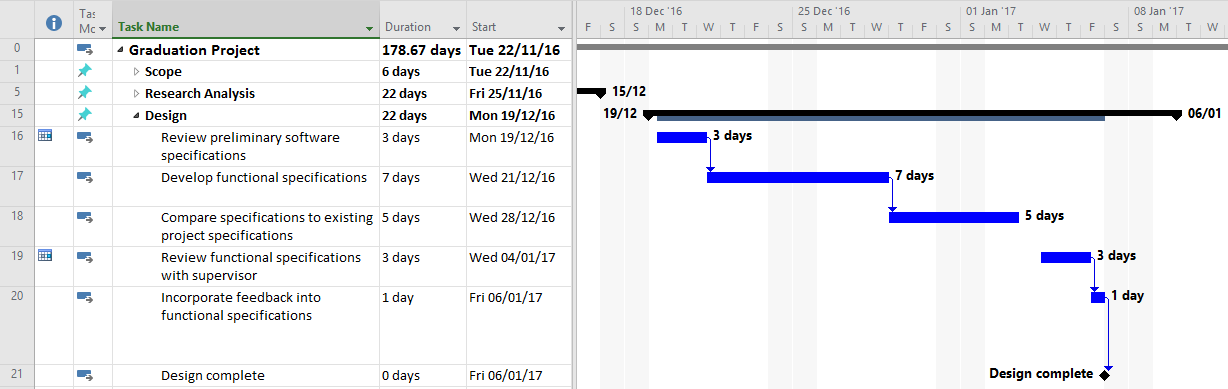
Plan D is to start from scratch, which is the least ideal scenario as it involves much trial and error.

## Work Plan (Gantt Chart)

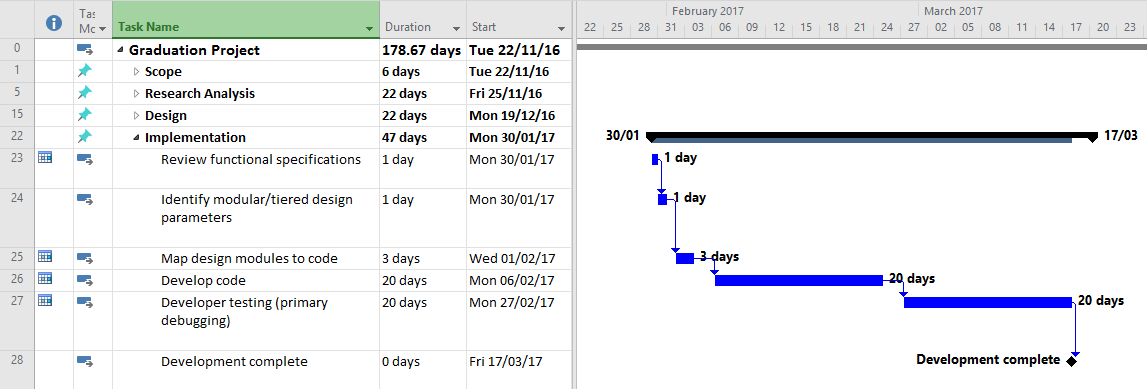
Phase 1



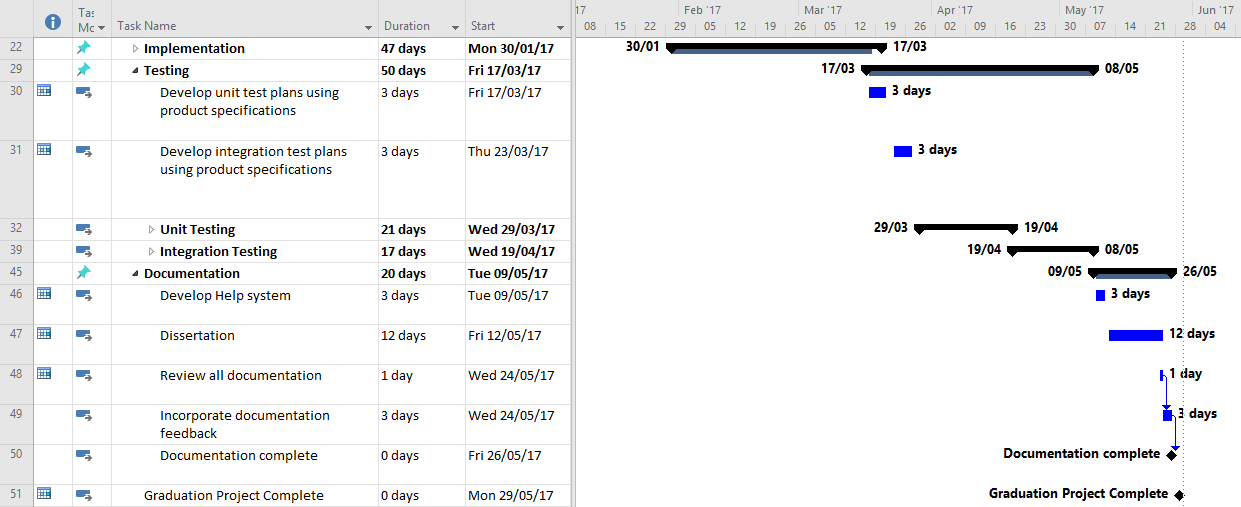
Phase 2



Phase 3



Phase 4



# Estimation Game Logistics

## Rules

The objective of the game is to estimate the amount of tricks, given a hand of thirteen cards. Each player plays one card in each hand. Whoever wins the hand collects a trick. There are thirteen tricks in each round. The card rankings are as follows, from highest to lowest: Ace, King, Queen, Jack, 10, 9, 8, 7, 6, 5, 4, 3 and 2.

## Card Suits

There are four suits in a deck of cards. In Estimation, they are given a certain weight ranking. The ranking of the suits is as follows:

1. Spades
2. Hearts
3. Diamonds
4. Clubs

## Game Structure

A game, or **Session**, of Estimation consists of eighteen rounds. The first thirteen rounds are played normally, with normal bidding rules. The final five rounds of the game are played in a fast bidding mode. The trump suit of each round is predetermined. The order of the trump suits in each of the final five rounds is as follows: Suns, Spades, Hearts, Diamonds and Clubs. If at any time a round is restarted, the score of the following round is incremented by two.

## Round Structure

Each player receives a hand of thirteen cards. The objective of the game is to correctly estimate the amount of tricks that can be taken with the given hand. A round consists of thirteen hands; four cards for each hand, one from each player, totalling 52 cards. The trump suit is the suit that is elevated above its normal rank, which is chosen at the beginning of each round. At the beginning of each round, players take turns bidding for the trump suit.

### Bidding

At the beginning of each round, players will be given a chance to declare a dash call. A dash call means that the player must not collect a single trick for the entire round. A dash call may only be declared before the trump suit bidding begins. The player on the right of the dealer begins bidding for the trump suit. A player may choose to make a bid or pass to the next player. The player with the highest bid may choose the trump suit. The bidder may also choose the trump suit as Suns, which means that there is no trump suit; the highest card of the played suit always wins. The trump suit ranking is as follows: Suns, Spades, Hearts, Diamonds and Clubs. For instance, a call of 5 Suns is greater than a call of 5 Spades, however a call of 6 Spades is greater than a call of 5 Suns. After the trump suit is chosen, the rest of the players declare their estimate bid, starting from the right of the trump suit bidder. The total sum of the players’ bids must not be equal to thirteen. The last player to make a bid estimate, or the player to the left of the trump suit bidder, has a bidding limit imposed on them. The round must either be under thirteen or over thirteen. If the last bidder makes a bid that is under or over the limit by at least two, they are flagged as a risk. If the round is under thirteen, that means that at least one player will take an extra trick. If the round is over thirteen, that means that at least one player won’t collect all of their estimated tricks. This system guarantees that at least one player will lose in each round.

### Gameplay Flow

Once everyone has finished bidding, the trump suit bidder opens play. If the trump suit is Suns, the highest card of the played suit wins the trick. Otherwise, the highest card of the played suit wins the trick, unless someone interjects with a trump suit card. The card thrown by the player must be of the same suit. If a player does not have any cards of that suit, any card may be thrown, including trump suit cards. If more than one trump suit card is played, the highest trump suit card wins. In the fast bidding rounds, the player to the right of the dealer starts bidding in the pre-determined trump suit. The order of the trump suits in each of the final five rounds is as follows: Suns, Spades, Hearts, Diamonds and Clubs. A player wins the round if they collect the exact amount of tricks they bid at the beginning of the round.

### Exceptions

A round may be restarted in two cases; the first of which is during call bidding, if every player chooses to pass instead of bidding. The second case is when every player loses the round. In other words, if no player collects the exact number of tricks they bid at the beginning of the round, the round is restarted. Whenever a round is restarted, the score multiplier for the following round is incremented by two.

## Scoring

The scoring model was taken from an Estimation Calculator application. [1]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case** | | | | **Scoring Equation** |
| Win | Caller / With | Call > 7 | Only winner |  |
| Not only winner |  |
| Call <= 7 | Only winner |  |
| Not only winner |  |
| Not Caller | Dash Call | Only winner |  |
| Not only winner |  |
| Normal Bid | Only winner |  |
| Not only winner |  |
| Lose | Caller / With | Call > 7 | Only loser |  |
| Not only loser |  |
| Call <= 7 | Only loser |  |
| Not only loser |  |
| Not Caller | Dash Call | Only loser |  |
| Not only loser |  |
| Normal Bid | Only loser |  |
| Not only loser |  |

* **With** refers to the case when a player bids a number of tricks equal to the caller. For instance, if the trump suit caller bids 5 Suns and a player bids 5 tricks, they are considered to be **With.** The scoring calculation for the caller and the **With** players is the same.
* The *call* variable refers to the amount of tricks the player bid.
* The *risk* variable refers to the amount of tricks the last bidder called. If a player bids 2 tricks under or over the limit, they are flagged as a risk. If a player bids 4 tricks under or over the limit, they are flagged as a double risk. If a player bids 6 tricks under or over the limit, they are flagged as a triple risk. Each risk is multiplied by 10 and added to the score. For instance, if a player bids 3 tricks under the limit, they are flagged as a double risk; when the score is calculated, 20 points are added to the score, before the multiplier.
* The *difference* variable refers to the absolute value of the predicted number of tricks subtracted from the actual number of tricks collected.
* If only one player wins, 10 points are added to the score calculation, before the multiplier. The same calculation applies if only one player loses.
* The *Multiplier* variable may be affected by several factors. If any of the following phenomena takes place, the*Multiplier* variable is incremented by two:

1. The previous round had no winners
2. The previous round had every player pass in the bidding phase
3. Three players called **With** the trump suit caller
4. Two players declare Dash Call

# Related Work (State-of-The-Art)

## Background

As per plan A, mentioned in section 1.5, the most ideal scenario is to find projects fitting the same criteria. Specifically, other Estimation projects. The main source of such related works is GitHub, which is a web-based repository service where developers can upload their projects and source codes online. The biggest problem with finding resources specifically for Estimation is that it is a region-specific card game, specifically in the Middle East. Therefore, resources are scarce, however, not non-existent.

The alternative to plan A, namely plan B, is to find projects fitting similar criteria. This may include other trick-taking games, such as Whist, Spades, Contract Bridge, Oh Hell, etc. The advantage to plan B is that these card games are international, therefore resources are much more likely to be readily available. The Artificial Intelligence component is likely to be more advanced as the game has a wider market, therefore more competition. The most useful part of finding resources is understanding various concepts of Artificial Intelligence programming, especially relating to card game science. Probabilities and statistics are likely to play an important role in the formulation of the Artificial Intelligence component.

## Literature Survey & Analysis

Existing Estimation projects are scarce, ones with open-source code even more so. There exist two reliable source codes for Estimation. However, they are written in languages other than Java, Python [1] and Ruby. [2] They would require extra effort and time to learn and understand. Seeing as they are Estimation games, the Artificial Intelligence component can simply be extracted and used for concepts.

Projects similar to Estimation are much more likely to yield useful results. One such project is Whist. Whist is a classic English trick-taking card game originating from the 18th century. There exist several Android applications for the game Whist. The most popular of those applications is aptly named “Whist”. The creator of this game has uploaded the source code and the documentation of the game to GitHub. The original source code language is Java. All things considered, this is the main source that will be used for the program structure and architecture. The creator of this project is Nikos Tsaousis. [3] Within the project uploaded to GitHub, he has included the code architecture (Model View Presenter), the class diagram, the techniques used in the creation of the Artificial Intelligence computer opponents and the software design patterns. In comparison to the other Estimation projects, Nikos’s application is much more helpful as it is documented and explained. The only downside of this project is that it is very complex. Therefore, it will take several days of analysis to fully understand. However, once it is understood, it will be a very useful resource with which to compare and learn from.

# Proposed solution

## Solution Methodology

As an expert Estimation player, the experienced gained over the years will be very useful in creating the Artificial Intelligence component. Naturally, as the program is developed, there will be several versions of the Artificial Intelligence with increasing complexity and difficulty as more time is spent on the code. Each version may be used as a benchmark to test against the final product. The game may have several difficulties, with the final version being the hardest; each version modelling a slightly increased difficulty with different tactics, strategies and approaches. The final version should be able to simulate the difficulty and unpredictability of playing against a real player.

Another proposed solution is one that actively learns from the player, gauging their level of expertise, evolving and adjusting its difficulty accordingly. However, this type of system is likely to be more complex because of the learning agent. It may use the advantage of having three computer opponents versus one human opponent. Thus, allowing it a much broader perspective of the game instead of the perspective of one opponent.

# Implementation

# Testing and Evaluation

# Conclusions and Changes

## Summary

Estimation is a popular Arabic card game. However, it being region-specific poses a problem in finding resources for it. An alternative solution to that problem is to find similar games, such as Whist, and using it as a guide to help create an Artificial Intelligence component.

References

1. Mustafa Haddara, *Estimation, cmdline implementation of a card game* <https://github.com/MustafaHaddara/estimation>, September 2016.
2. Bramdemoor, *Estimations card game in ruby*<https://github.com/bramdemoor/Estimations>, December 2014
3. Nikos Tsaousis, Whist is a trick-taking, real-time multiplayer card game  
   <https://github.com/tsanikgr/whist>, December 2015