

## Projet LO02

# *The Other Hat Trick*

Il vous est proposé dans ce projet de concevoir avec UML et développer en Java une version électronique du jeu de cartes « The Other Hat Trick ».

## 1. Règles du jeu

Les règles du jeu vous sont données en annexe. Elles servent de document de référence pour l'expression des besoins du logiciel développé.

Au-delà, on considère aussi les besoins suivants :

- L'application devra permettre de jouer à plusieurs joueurs qui seront des joueurs physiques auxquels s'ajoutent des joueurs virtuels. Dans le cas d'un joueur virtuel, le jeu devra lui attribuer une stratégie de jeu simple mais cohérente qui lui permettra de déterminer action mener jouer à chaque instant.
- L'ensemble de l'application sera intégré dans une interface graphique. On pourra utiliser des images telles que celles données en annexe.
- Avant la conception et le développement de l'interface graphique, une interface rudimentaire en ligne de commande permettra de tester le moteur du jeu. Cette interface devra être conservée lors de l'évaluation fonctionnelle et la remise des fichiers.
- On intégrera une notion d'extension qui correspondent à des nouvelles cartes, qu'il vous faudra concevoir et implémenter. En début de partie, il sera demandé au joueur s'il souhaite intégrer ces cartes d'extension ou pas à la partie qui va débiter.
- On intégrera enfin une notion de variantes dans les règles du jeu. En début de partie, le choix d'une variante sera proposé au joueur et c'est la variante choisie qui imposera les règles d'une partie. Il est demandé de concevoir et implémenter deux variantes en plus des règles de base.
- L'architecture retenue devra veiller au respect des règles de la conception orientée objet. L'architecture devra ainsi être (1) modulaire en identifiant des composants indépendants liés entre eux par des relations (2) extensible en permettant le changement de règles exposé ci-avant ou l'ajout de cartes avec de nouvelles fonctions.

## 2. Phases et jalons

Le projet sera découpé en trois phases qui conduiront à des soutenances effectuées durant une séance de TP dédiée. Les trois phases du projet sont :

1. la modélisation UML initiale ;
2. le développement du moteur du jeu, utilisable en lignes de commandes ;
3. le développement de l'interface graphique et la remise du code source documenté.

Pour chaque phase et jalon associé, les consignes à suivre vous sont données ci-après.

### 2.1. Phase 1 : modélisation UML

La modélisation UML proposée sera exposée sous la forme d'une présentation qui devra suivre la structure suivante :

1. Introduction : présentation du projet tel que vous l'avez compris (pas de copier/coller de l'énoncé) et annonce du plan du document
2. Diagramme de cas d'utilisation : identifier l'ensemble des cas d'utilisation du système, leurs relations et les expliquer.

3. Diagramme de classes : décrire et expliquer chaque élément de conception du diagramme. Par ailleurs, vous indiquerez l'étude que vous aurez faite sur le patron de conception *Strategy*, utilisé en autres, pour les joueurs virtuels, et la manière dont vous l'intégrez dans votre modélisation. Vous noterez qu'il est inutile de modéliser les interfaces de commande du jeu (en lignes de commandes ou graphique) ; on ne représentera que le cœur de l'application.
4. Diagramme de séquence : Proposer un diagramme de séquence pour le déroulement d'un tour de jeu.
5. Conclusion : Identifier les aspects sûrs de la modélisation et ceux dont vous pensez que le développement pourrait induire une modification.

## **2.2. Phase 2 : cœur de l'application et interface en lignes de commandes**

Cette phase, validée par une seconde soutenance orale, consistera à faire une première démonstration du développement. A cette étape, le moteur du jeu devra être fonctionnel et utilisable en lignes de commandes. Mais aucune mise en forme particulière n'est demandée (mise en packages, documentation, mise au propre du code, ...). De même, aucun support de présentation ne devra être préparé pour cette seconde soutenance.

## **2.3. Phase 3 : projet complet et documentation**

Cette dernière phase consistera à mettre en œuvre la version finale du projet ainsi que toute la documentation qui l'accompagne. D'un point de vue du travail de développement, il consistera à intégrer une interface graphique au cœur de l'application développée précédemment. Attention, le jeu devra rester jouable à la fois en ligne de commandes et par le biais de l'interface graphique, au sein d'une même partie. Pour ce faire, vous devrez mettre en œuvre le patron de conception MVC, vu en cours en considérant que l'application possède deux vues concurrentes.

Une soutenance orale permettra de valider les aspects fonctionnels de l'application et une remise des fichiers du projet permettra d'évaluer le code, son organisation, sa structure et sa documentation.

## **Evaluation fonctionnelle**

Elle sera effectuée par le biais d'une soutenance qui aura lieu durant votre dernière séance de TP. Elle s'effectue par binôme et dure 15 minutes durant lesquelles vous présenterez :

- L'état fonctionnel de votre application par le biais d'une démonstration. Une fois la démonstration passée, tout développement supplémentaire (correction de bugs, ajout de fonctionnalités, ...) est inutile car l'état fonctionnel de votre projet sera évalué à ce moment. Suite à cela, quelques questions relatives au code que vous avez produit vous seront posées.
- Une présentation qui indiquera :
  1. Le diagramme de classes final. On détaillera ici et justifiera les changements entre la version initiale du diagramme de classes et la version finale, effectivement implémentée dans le code. Pour ce faire, on présentera les deux diagrammes.
  2. L'état actuel de l'application : Cette partie donnera de manière précise l'état de l'application en regard du présent cahier des charges. On y indiquera clairement ce qui a été implanté et ce qu'il ne l'est pas, ce qui fonctionne et ce qui reste buggé.

Remarque : Pour cette évaluation fonctionnelle, il ne vous est pas demandé de mettre au propre le code présenté (respect des conventions d'écriture, commentaire Javadoc, suppression des blocs de codes inutiles mis en commentaires, ...). Par contre, cela devra être fait pour la remise des fichiers.

## Remise des fichiers du projet

Les fichiers du projet à remettre seront déposés sur Moodle, dans un espace dédié à cet effet, et placés dans une archive ZIP dont le nom du fichier doit suivre la convention de nommage suivante :

<Nom1>\_<Nom2>\_ProjetLO02\_A18.ZIP, où <Nom1> et <Nom2> sont données par ordre alphabétique. Par exemple : Doyen\_Langeron\_ProjetLO02\_A18.PDF

L'archive devra comporter les dossiers suivants :

- **src** : contiendra l'ensemble des sources du projet ;
- **dist** : contiendra un fichier nommé projet.jar qui contiendra une archive JAR exécutable du projet. C'est ce fichier qui sera exécuté pour tester si besoin votre projet après la soutenance ;
- **classes** : contiendra l'ensemble des classes compilées du projet ;
- **doc** : contiendra la javadoc du projet.

Enfin, les consignes à observer pour la mise en forme du code développé dans le cadre du projet sont les suivantes :

- L'ensemble du code doit être documenté par le biais de commentaires javadoc. Il ne faudra pas se contenter des balises standard mais décrire précisément la fonction de chaque élément de code (classe, méthode, ...) en respectant les conventions de documentation pour la production logicielle telles que vues durant le semestre.
- Le code devra être propre et suivre les conventions d'écriture spécifiées par Oracle (nommage, indentation, blocs, casse, ...). Aucun code obsolète, placé dans un commentaire, ne devra figurer dans les fichiers sources.

## 3. Consignes générales

### 3.1. Calendrier

Semaine	Événement
2 (17/09)	Distribution des projets et constitution des binômes
6, 7 et 9 (15/10, 22/10 et 5/11)	Soutenance Jalon 1
12 et 13 (26/11 et 3/12)	Soutenance Jalon 2
18 (7/01)	Soutenance
19 (vendredi 18/01 à 23:55)	Remise des fichiers

### 3.2. Respect des droits d'auteurs

Le présent projet consiste en la modélisation et le développement logiciel d'un jeu dont les conditions d'utilisation sont spécifiées ici : <http://www.goodlittlegames.co.uk/about-faq.html>. L'Université de Technologie de Troyes n'utilise cette œuvre que dans seul un cadre d'enseignement et à des fins exclusivement pédagogiques. En effectuant ce travail au sein d'une unité d'enseignement, l'étudiant s'engage à respecter les droits d'auteurs et d'exploitation liés à tout support de stockage et de diffusion numérique du code produit dans le cadre de son projet et à respecter le cadre exclusivement pédagogique associé à cette production. Toute autre forme d'exploitation et de diffusion par un étudiant relève de sa seule responsabilité individuelle.

### 3.3. Autres consignes

- Pour l'ensemble des livrables du projet, aucun retard ne sera accepté et aucune raison ne pourra le justifier.
- Vos intervenants de TP sont vos interlocuteurs exclusifs pour toute question relative au projet.

# The Other

## HAT TRICK

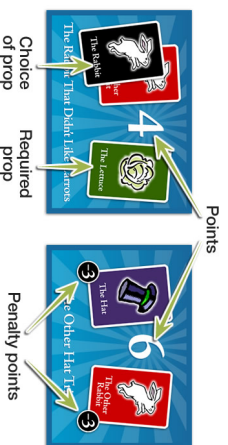
**Brett J. Gilbert**  
3 players • 10 minutes

*Three rival magicians take to the stage in a battle of prestidigitation to produce just the right props at just the right time. Only the magician who performs the most valuable tricks will win!*

### THE CARDS

The game contains 17 cards:

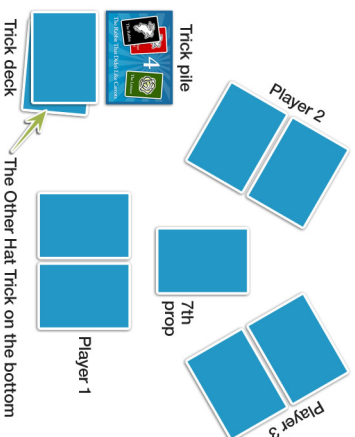
- **7 Props:** The Rabbit, The Other Rabbit, The Hat, The Lettuce and three identical Carrots.
- **10 Tricks:** Each trick card has a name, a points value, and shows the combination of props that are required by a player to successfully perform the trick. The trick called *The Other Hat Trick* also shows penalty points that the players holding the Hat or The Other Rabbit will receive at the end of the game if no-one successfully performs this trick.



### SETUP

Place the **trick** called *The Other Hat Trick* face-down by the side of the play area, then shuffle the other nine trick cards and place these face-down on top, to form a **trick deck** of all ten trick cards. Flip the top trick card face-up to start the **trick pile** beside the deck.

Shuffle the seven **props** and deal two cards face-down in front of each player. Place the final prop card face-down in the middle of the play area without looking at it: this is the '7th prop'.



### AIM OF THE GAME

The player who scores the most number of points by the end of the game is the winner.

### HOW TO PLAY

The youngest player always goes first. Play then continues clockwise. On your turn do the following things in order:

1. **Choose trick:** You must first announce whether you will try to perform the trick face-up on the top of the trick pile, or flip over the next trick card from the deck and try to perform that instead.
  2. **Prepare props:** You must swap one of your props with a prop in front of another player.
  3. **Perform trick:** You may now either successfully perform the trick or announce you have failed.
- During the game, remember two important things:
- You may always look at the two prop cards in front of you, but **never** at the other player's face-down props, or the 7th prop in the middle of the play area, unless the rules allow it.
  - Never pick up both of your props at the same time, or swap them round. When you look at your face-down props, always be careful to replace them in the same position.

### 1. Choose trick

Look at the trick face-up on top of the trick pile. You **must** announce that you will either try to perform this trick, or you will flip the next trick face-up on top of it. If you do this, you must then try to perform the flipped trick instead. You can't change your mind!

### 2. Prepare props

Before you try to perform your trick, you **must** prepare your props by swapping one prop in front of you with a prop in front of another player. You must do this, even if you would prefer not to!

Choose one of the two props in front of you and swap it with one of the props in front of another player. You **cannot** swap your prop with the 7th prop in the middle of the play area.

You cannot look at a face-down prop in front of another player before making the swap. Make sure that each card is placed into the position of the card with which it was swapped.

If one or both of the swapped cards are already face-up (this can happen later in the game), they remain face-up during the swap.

### 3. Perform trick

Once you have prepared your props, you have the opportunity to perform your trick.

If your two props create a combination that matches the face-up trick, you may reveal them both and announce you have successfully performed the trick. If you cannot do this, or chose not to, you have failed to perform the trick and must forfeit.

After you perform or forfeit the trick, your turn ends and play continues clockwise.

### Success

If you reveal your prop cards to show a combination that matches the face-up trick you have successfully performed it! You may, if you wish, reveal them with a flourish and say "*Ta-Dah!*", the other players may then greet this with a small round of applause.

Take the trick card and place it face-up in front of yourself. You have now scored the number of points shown on the card.

After a successful performance you **must**

immediately use 'sleight of hand' to rearrange the props and confuse the other players. Pick both of your revealed props **plus** the face-down 7th prop. Look at all three of these props and secretly choose two to keep **face-down** in front of yourself. Place the remaining prop face-down in the middle of the play area as the

new 7th prop. Remember: The 7th prop is out of play until the next time a player successfully performs a trick.

**Note:** If the trick pile is **empty** at the end of your turn, flip the next trick card from the deck face-up, ready for the next player's turn. Do this after you have performed your sleight of hand.

### Forfeit

If you do not reveal two props in front of you that match the face-up trick, either because you cannot or because you choose not to, you must announce that you have forfeit the trick.

Your penalty is to flip one of your face-down props, revealing it to the other players, and leave it face-up. If one of your props is already face-up, you must flip the other prop. If both props are face-up, there is no additional penalty.

### END OF THE GAME

The game ends when either one player successfully performs *The Other Hat Trick*, or all three players fail to perform this final trick in turn.

Once the game ends, reveal any face-down props, and count up the points on your successfully performed trick cards. If all players failed to perform *The Other Hat Trick*, the props The Hat and The Other Rabbit each score a penalty of -3 points to the player or players who have them at the end of the game. If a player successfully performed *The Other Hat Trick*, then there are no penalties.

The player who scored the most points wins the game.

**good little games**

*With thanks to the following playtesters:  
James & Maureen Gilbert, Alan Paul, Richard Breese,  
John Yount, Rob Harris, Sebastian Bleasdale, Ian Vincent,  
John Turner, Stephen Lanele.*

Rules v1.0 • © 2013 Brett J. Gilbert  
[www.goodlittlegames.co.uk](http://www.goodlittlegames.co.uk)

The Hungry Rabbit

1

The Bunch of Carrots

2

The Vegetable Patch

3

The Rabbit That Didn't Like Carrots

4

The Pair of Rabbits

5

The Vegetable Hat Trick

2

The Carrot Hat Trick

3

The Slightly Easier Hat Trick

4

The Hat Trick

5





The Lettuce



Carrots



The Other Hat Trick

6



The Hat



Carrots



The Rabbit



Carrots



The Other Rabbit