

Projet LO02

Jest

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

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 (1) le patron de conception *Strategy*, utilisé en autres, pour les joueurs virtuels, et (2) le patron *Visitor*, utilisé pour le comptage des scores. Vous préciserez 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.

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_A19.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, le diagramme de classes final et l'état fonctionnel de l'application..

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.
- 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.
- 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.

3. Consignes générales

3.1. Calendrier

Semaine	Evénement
2 (16/09)	Distribution des projets et constitution des binômes
6, 7 et 9 (14/10, 21/10 et 4/11)	Soutenance Jalon 1
12 et 13 (25/11 et 2/12)	Soutenance Jalon 2
18 (6/01)	Soutenance
19 (vendredi 17/01 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.



Brett J. Gilbert

3 or 4 players • 10 minutes

THE CARDS

The game contains 18 cards:

- **16 suit cards:** Each of the 4 suits contains 4 cards: Ace, 2, 3 & 4. The Ace is a special card and can become a 5 in some circumstances.
- **Joker:** The Joker is here to cause trouble for the players: Watch out for him!
- **Reference card:** Shows the hierarchy of the suits and the ways in which the card combinations score.

SETUP

Place the reference card face-up in the middle of the play area. Then shuffle the 16 suit cards and the Jester together face-down. Deal 2 cards face up beside the reference card: these cards are the **trophies**. Place the remaining cards face down as the draw deck.

The game is probably best with 3 players, but if playing with 4 players, deal only 1 face-up trophy.

AIM OF THE GAME

The player who collects a 'Jest' of cards that is worth the most points at the end of the game is the winner.

HOW TO PLAY

Each game is made up of a sequence of rounds. In each **round** each player receives 2 cards and makes an **offer** of these cards to the other players. In turn, players take 1 card from one of the offers on the table and adds this card to their face-down **Jest**.

At the end of the game, the players compare their Jests. Each of the **trophies** is awarded to the player whose Jest meets the trophy's condition, shown in the orange band. The trophy will change the value of the Jest of the player who receives it — for better or worse!

Each round is played in three phases. 1: Deal cards, 2: Make offers, 3: Take cards.

1. Deal cards

In the first round, deal each player 2 cards face-down from the draw deck.

In later rounds, first collect up the cards not taken from the players' offers in the previous round (each player will have 1 card remaining in front of them) and place these cards in a face-down stack. Then add to the stack a number of cards from the draw deck equal to the number of players. Shuffle this stack and deal 2 cards to each player.

2. Make offers

All players look at the two cards dealt to them. Each player first chooses 1 card to play **face down**. When everyone has done this, all players then reveal their other card **face up** beside the first card. The pair of cards in front of you — one face up, one face down — is your **offer** to the other players.

3. Take cards

Compare the face-up cards in each offer. The player whose face-up card has the highest face value takes the first turn. The Joker has a face value of 0. In this phase of the game, an Ace always has a face value of 1.

Breaking ties: If 2 or more players have equally valuable face-up cards in their offers, break the tie in favour of the player whose face-up card is in the strongest suit. Spades is the strongest suit, then Clubs, Diamonds and Hearts. The reference card illustrates this hierarchy.

On your turn you must take 1 card from **another** player's **complete** offer and add it to your **Jest**. Keep the cards in your Jest face down, separate from any other cards in play.

- When you take a card from an offer, you can't look at the face-down card before choosing.
- You may always check the cards in your own offer and Jest.
- You may not take a card from an incomplete offer. You must choose an offer that still contains 2 cards.

Next player

The player from whose offer you took a card goes next, unless, later in the round, that player has already had a turn and taken a card. In that case, compare the face-up cards in the offers of the remaining players. The player whose card has the highest face value goes next. Break ties as before, if necessary.

If you are the final player in a round and your own offer is the only one that still contains 2 cards, you **must** take a card from your offer. This is the only time that taking from your own offer is allowed!

End of the round

At the end of the round, each player will have added 1 card to their Jest, and 1 card from each player's offer will remain on the table in front of them.

- If there are cards remaining in the draw deck, the game continues. Begin a new round.
- If there are no cards left in the draw deck, the game ends. All players **must** now take the 1 card remaining from their offer in the final round, **add it to their Jest**, and then reveal their cards.

END OF THE GAME

Before the winner is decided, compare everyone's Jests to discover whose Jest meets the condition in the orange band of each trophy. Award both trophies simultaneously. If you win a trophy you must add it to your Jest.

The player who has the highest value Jest — *after* the trophies have been awarded — is the winner!



Highest
Player with highest face value card in this suit.



Lowest
Player with lowest face value card in this suit.



Majority
Player with most cards of this face value.



Joker
Player with the Joker.



Best Jest
Player with highest value Jest.



Best Jest, No Joke
Player with highest value Jest, but without the Joker.

If 2 or more players tie when meeting a 'Majority' condition, award the trophy to the player with the card

of that face value in the strongest suit. If 2 or more players tie when meeting a 'Best Jest' condition, award the trophy to the player whose Jest contains the card with the highest face value (which, if also a tie, should be broken in favour of the card in the strongest suit).

JEST VALUE

The reference card summarizes how the different suits and card combinations affect the value of your Jest.

Suits

- Spades ♠ and Clubs ♣ always *increase* the value of your Jest by their face value.
- Diamonds ♦ always *reduce* the value of your Jest by their face value.
- Hearts ♥ are worth nothing unless you have the Joker.

Joker & Hearts

- If you have the Joker and **no Hearts ♥**, the Joker is worth a bonus 4 points.
- If you have the Joker and **1, 2 or 3 Hearts ♥**, the Joker is worth nothing and every Heart *reduces* the value of your Jest by its face value.
- If you have the Joker and **all 4 Hearts ♥**, the Joker is worth nothing but every Heart *increases* the value of your Jest by its face value. (In a 4-player game, this score is only possible if the trophy is either the Joker or a Heart ♥.)

Aces

If you have an Ace which is the *only* card of that suit in your Jest, the card becomes a 5, with a face value of 5. Otherwise it remains an Ace, with a face value of 1.

Black pairs

If you have a Spade ♠ and a Club ♣ with the *same* face value, the pair is worth a bonus 2 points in addition to the face values of the cards.

*With thanks to the following playtesters:
Andrew Sheerin, John Turner, David Johnston, Matt Dunstan,
Sam Mercer, Adam Harper, Richard Stokes.*



Rules v1.0 • © 2013 Brett J. Gilbert
www.goodlittlegames.co.uk



