Kings and Things  
COMP 3004 Winter 2014  
Team 17  
Tyler Babin  
Ryan Bottema  
Brandon Schurman

Table of Contents

Section 1 – Game rules………………………………….1

Section 2 – Requirements……………………………18

Section 3 – Use Cases………………………………....22

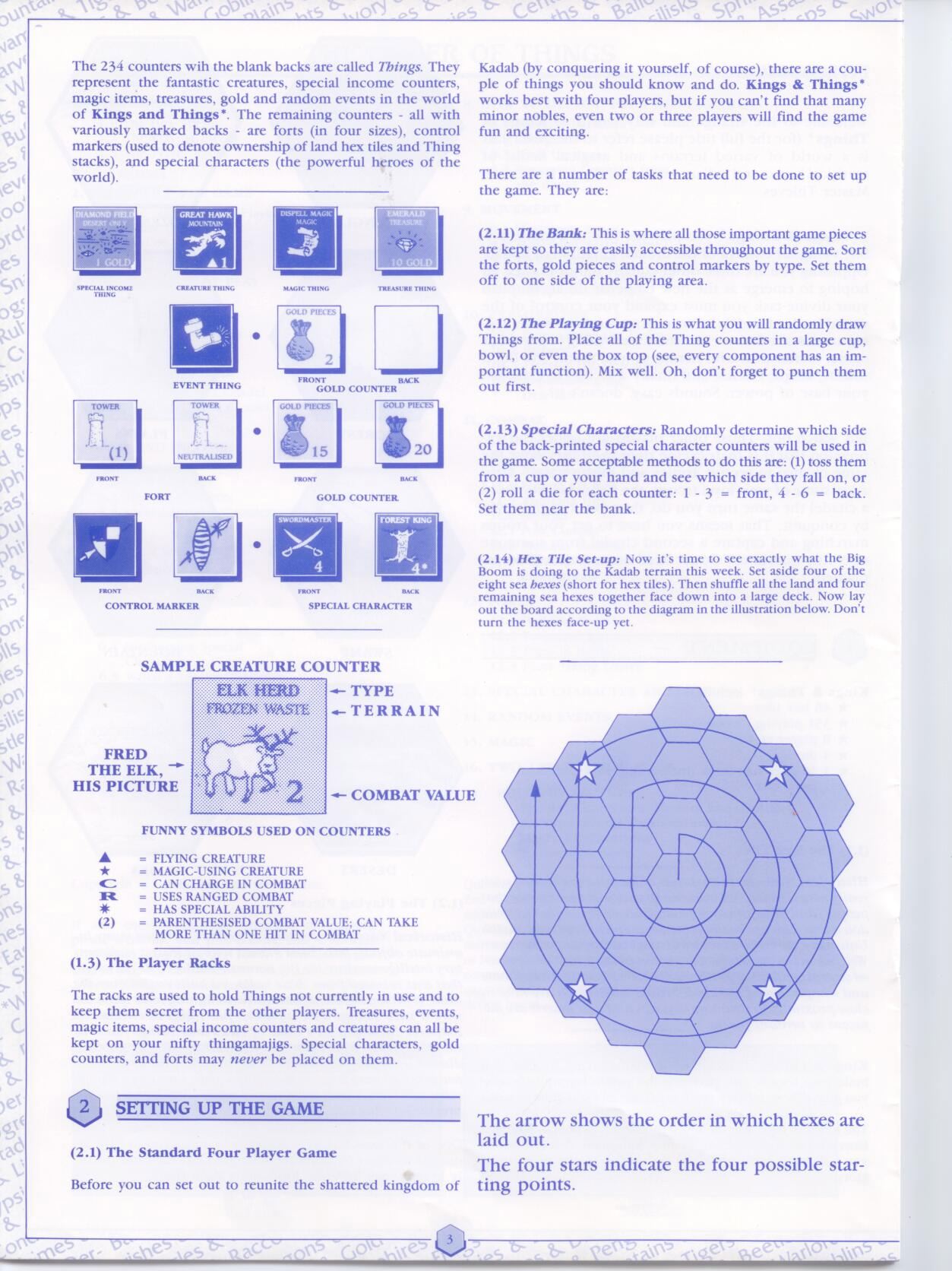
Section 4 – Design Decisions...........................39

Section 5 – Class Diagrams

Section 6 – Sequence Diagrams

Section 1 - Game Rules

Below are the official rules for Kings and Things. Unique Identifiers have been given to the different sections to allow for requirements traceability. These identifiers will cosist of 3 sections: GR-XX-0X, with the first section, GR, standing for Game Rules. The second section, XX, will consist of the current phase of the rules, for example SG for setup game, GC for gold collection, etc. The last section, 0X, will stand for the subsection of the rule. Example: GR-GC-01 would refer to the first section of the Gold Collection Phase.

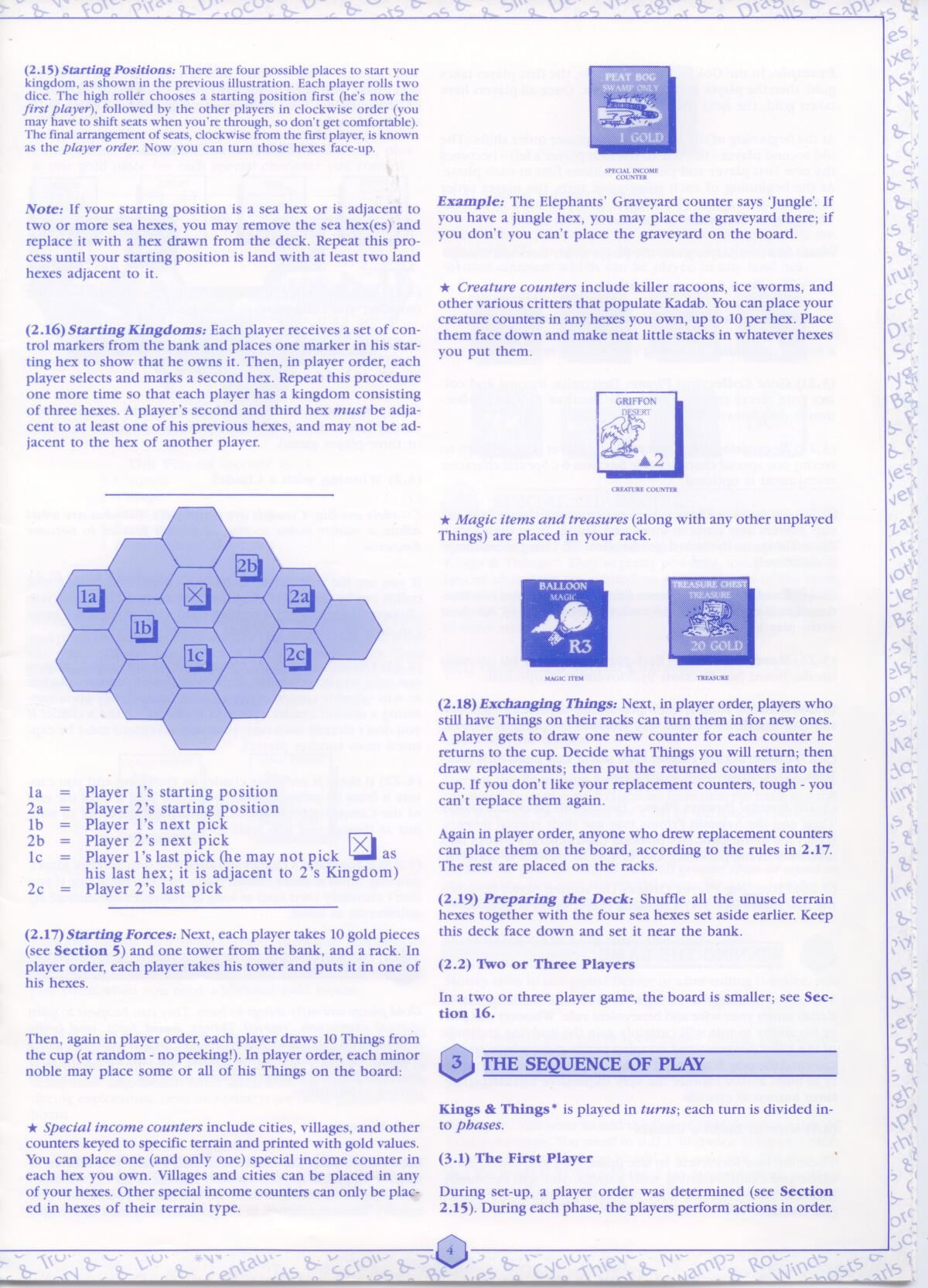


GR-SG-04

GR-SG-03

GR-SG-02

GR-SG-01

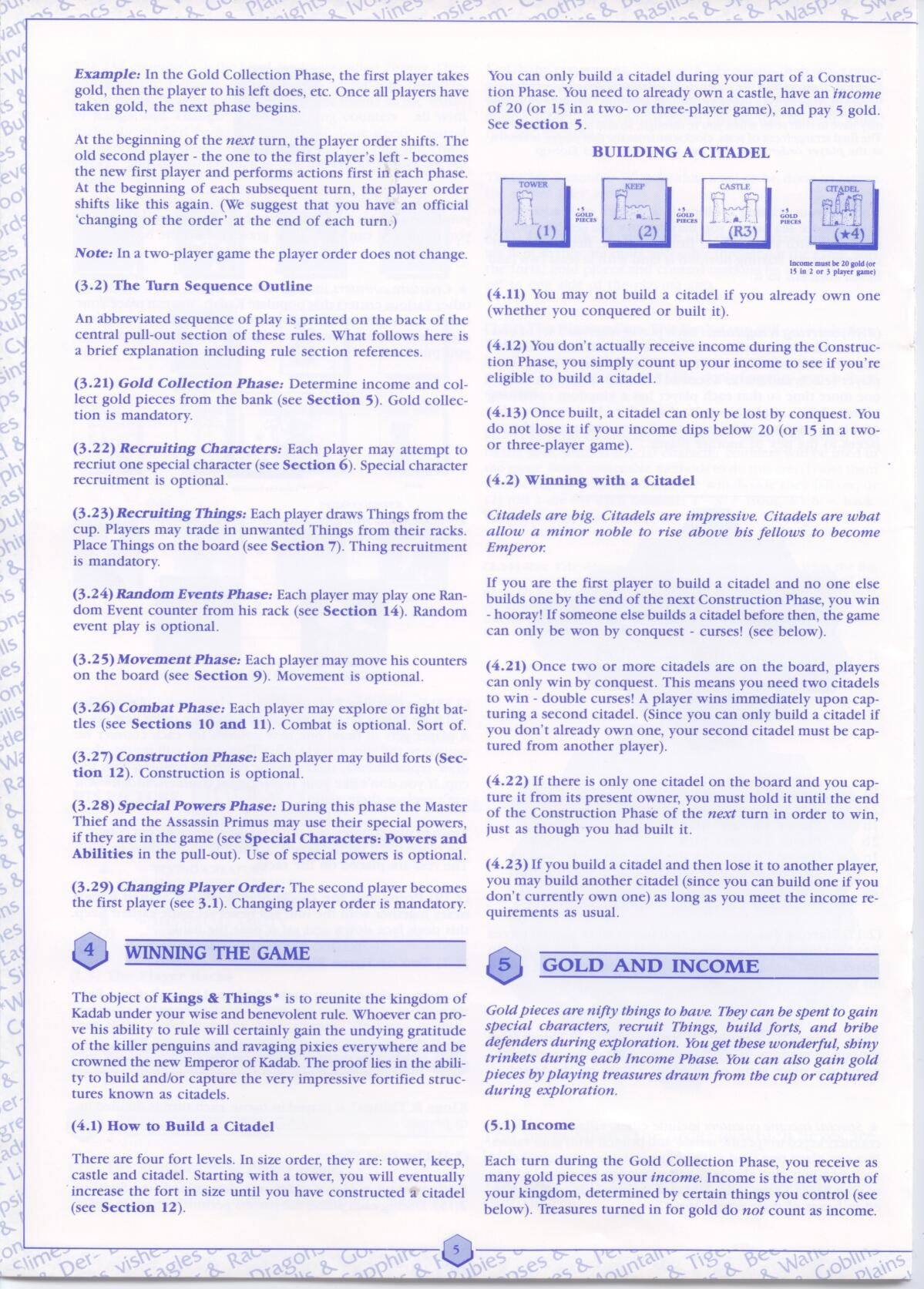


GR-SG-08

GR-SG-07

GR-SG-06

GR-SG-05



GR-TS-09

GR-TS-08

GR-TS-07

GR-TS-06

GR-TS-05

GR-TS-04

GR-TS-03

GR-TS-02

GR-TS-01

GR-GC-01

GR- GV-08

GR- GV-07

GR- GV-06

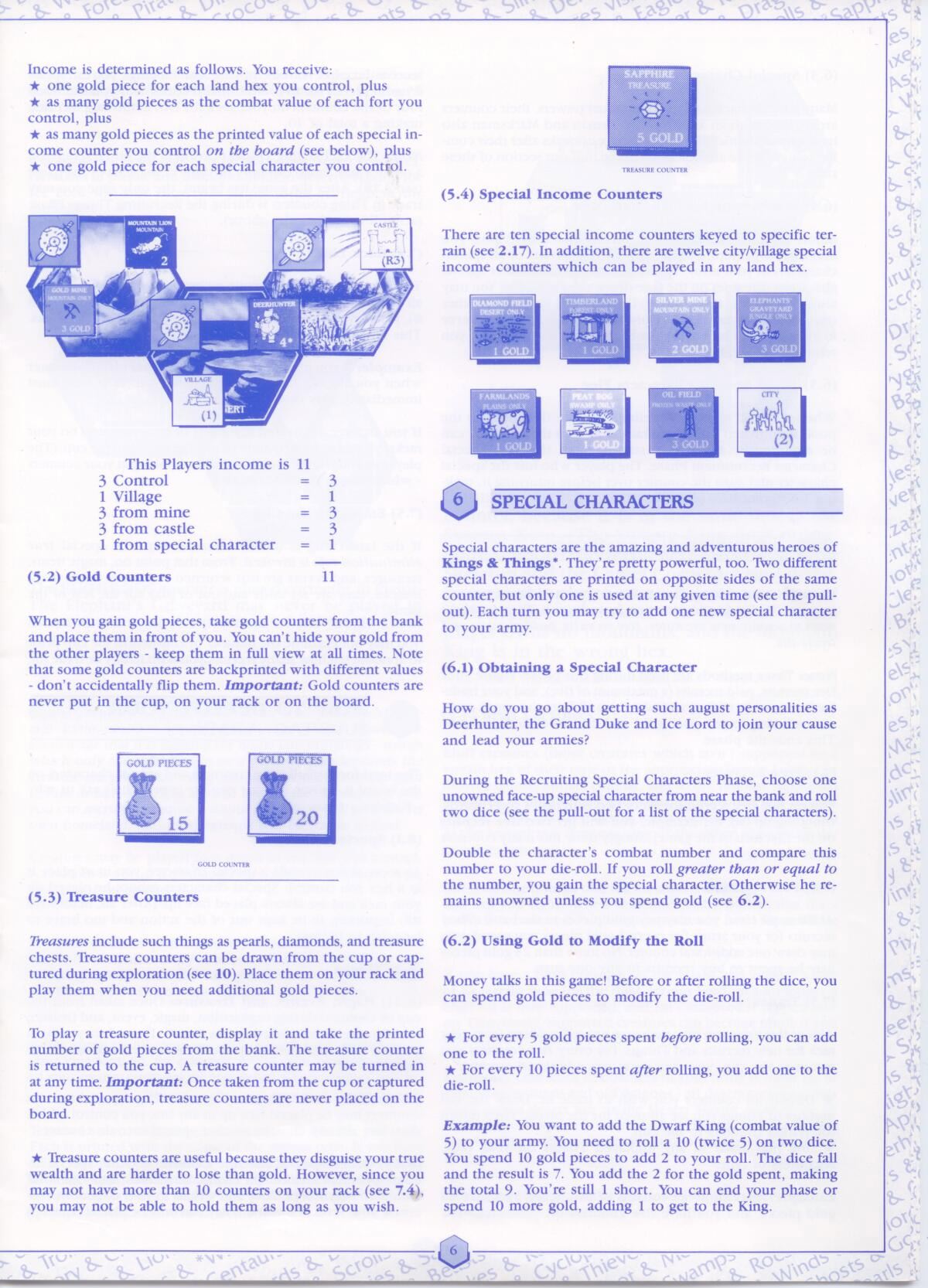
GR- GV-05

GR- GV-04

GR- GV-03

GR- GV-02

GR-GV-01

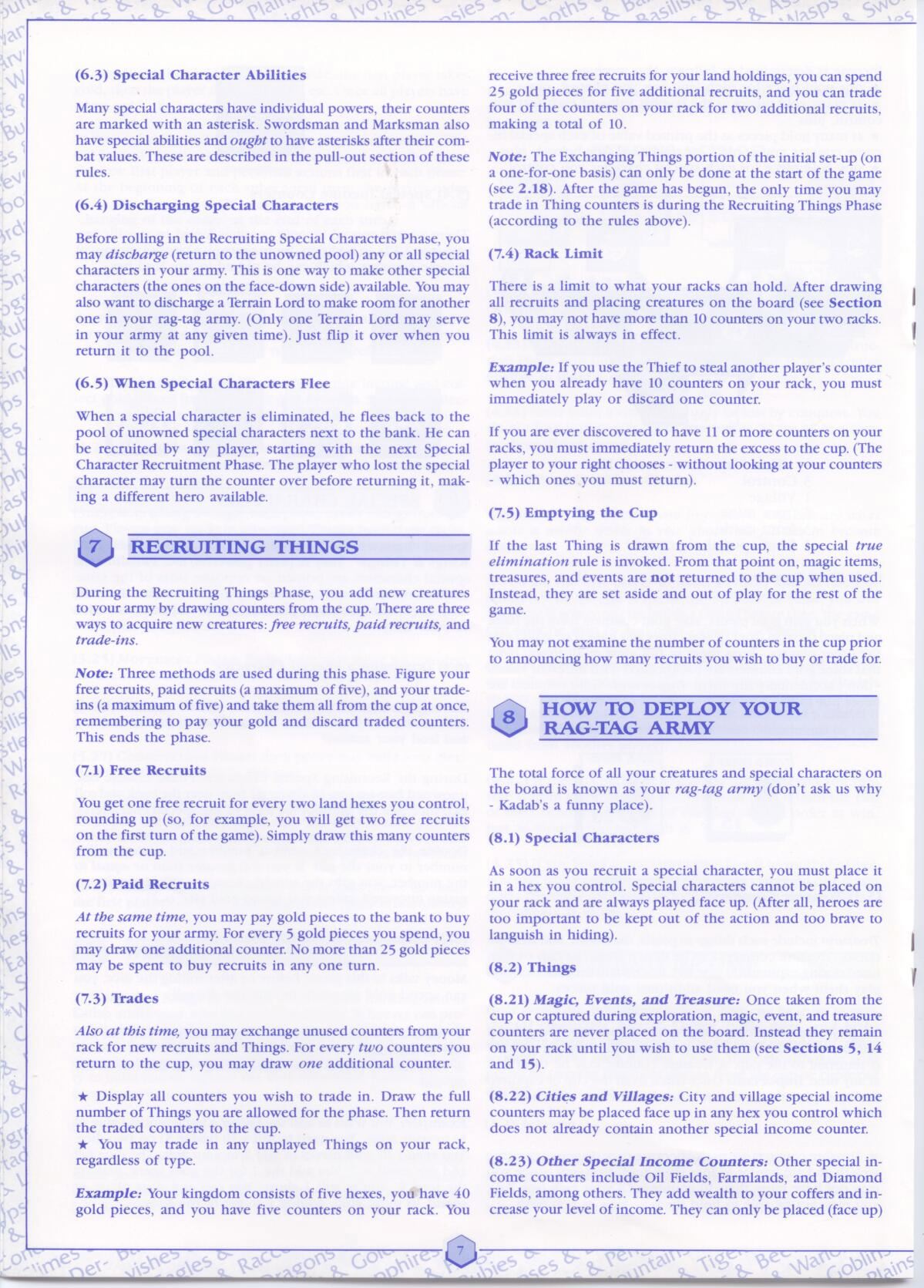


GR-SC-01

GR-GC-02

GR-GC-04

GR-GC-03



GR-DT-02

GR-DT-03

GR-DT-01

GR-RT-04

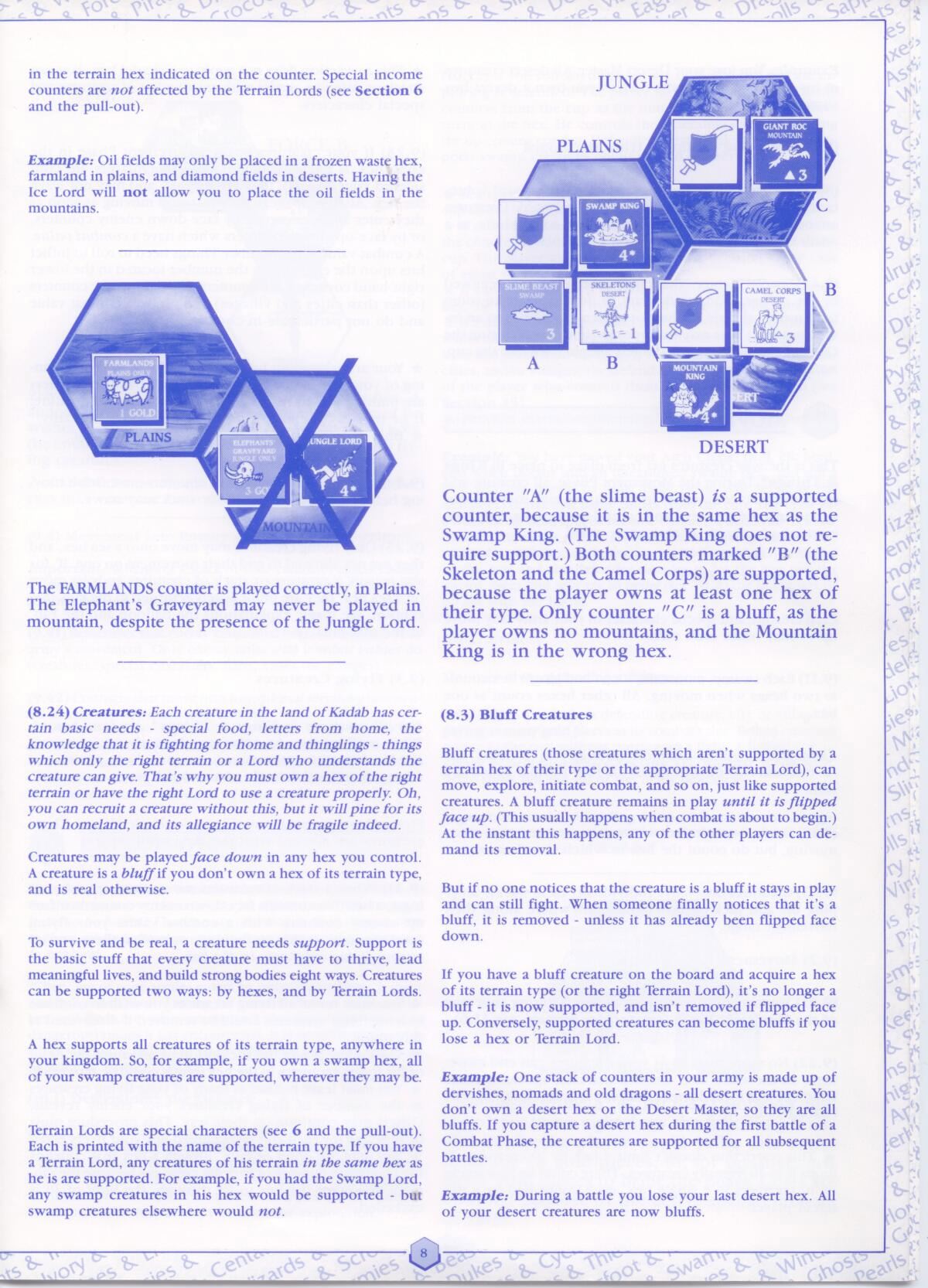
GR-RT-03

GR-RT-02

GR-RT-01

GR-SC-03

GR-SC-02

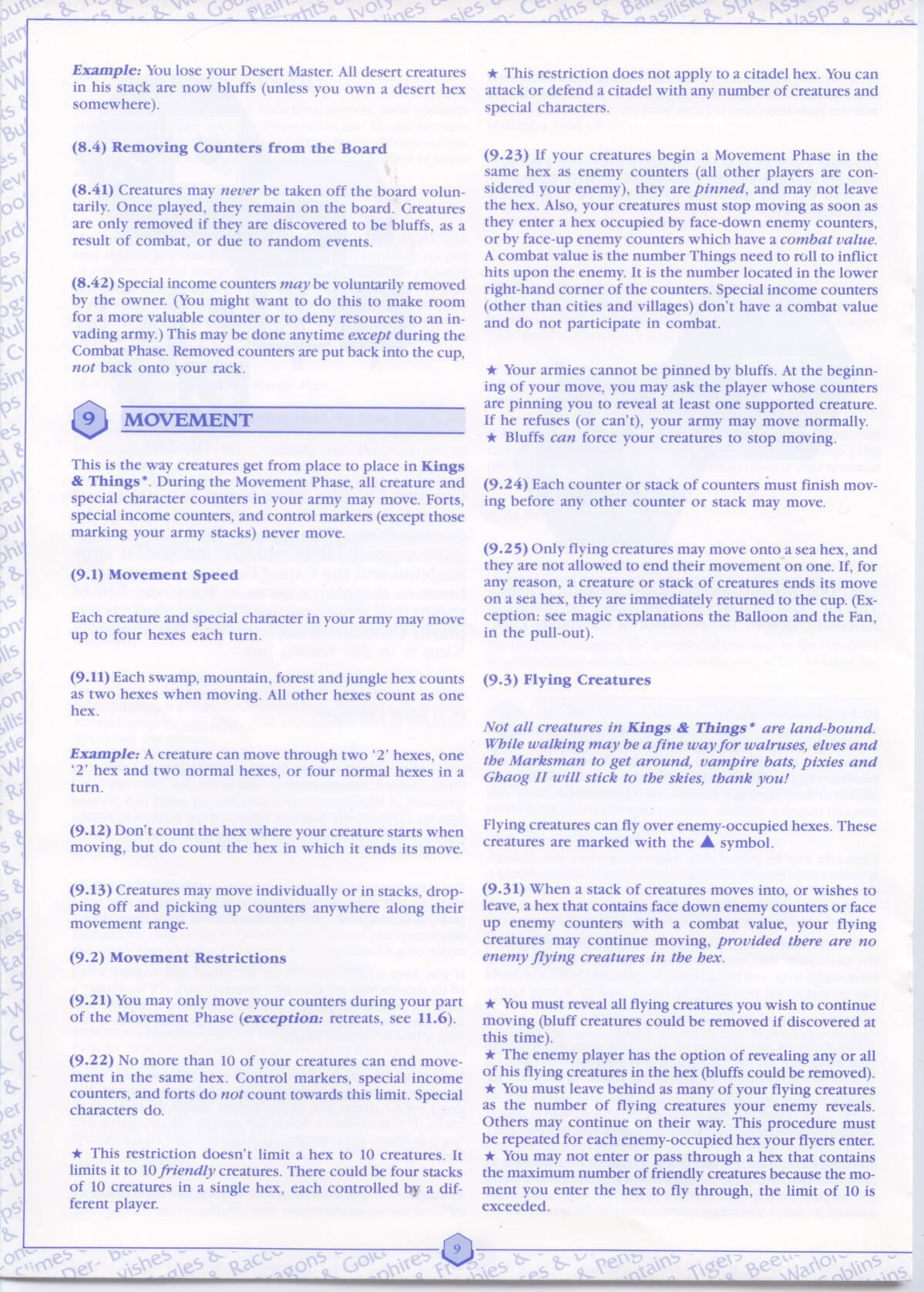


GR-DT-06

GR-DT-05

GR-DT-07

GR-DT-04



GR-MP-06

GR-MP-04

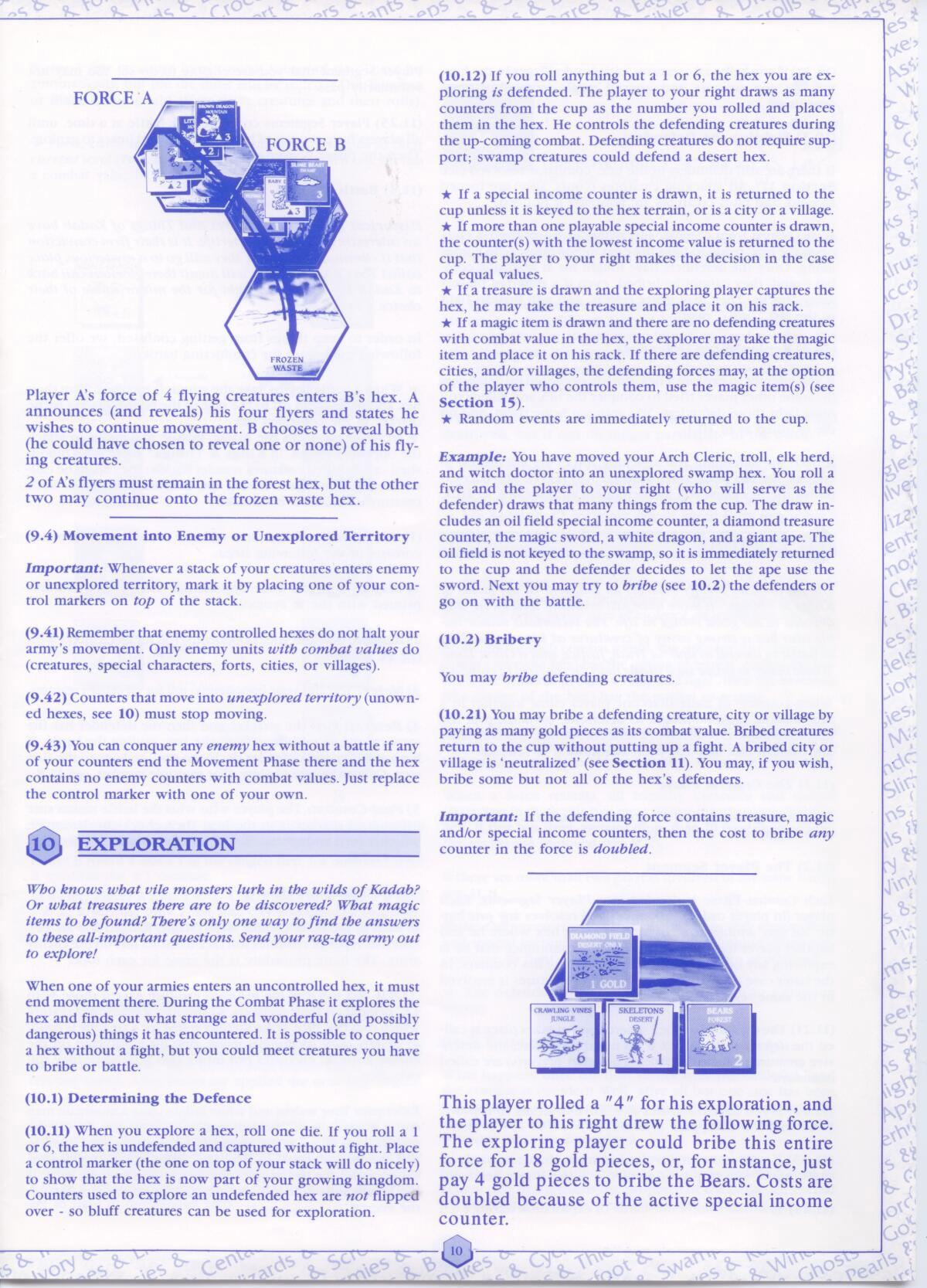
GR-MP-05

GR-MP-03

GR-MP-02

GR-MP-01

GR-DT-08

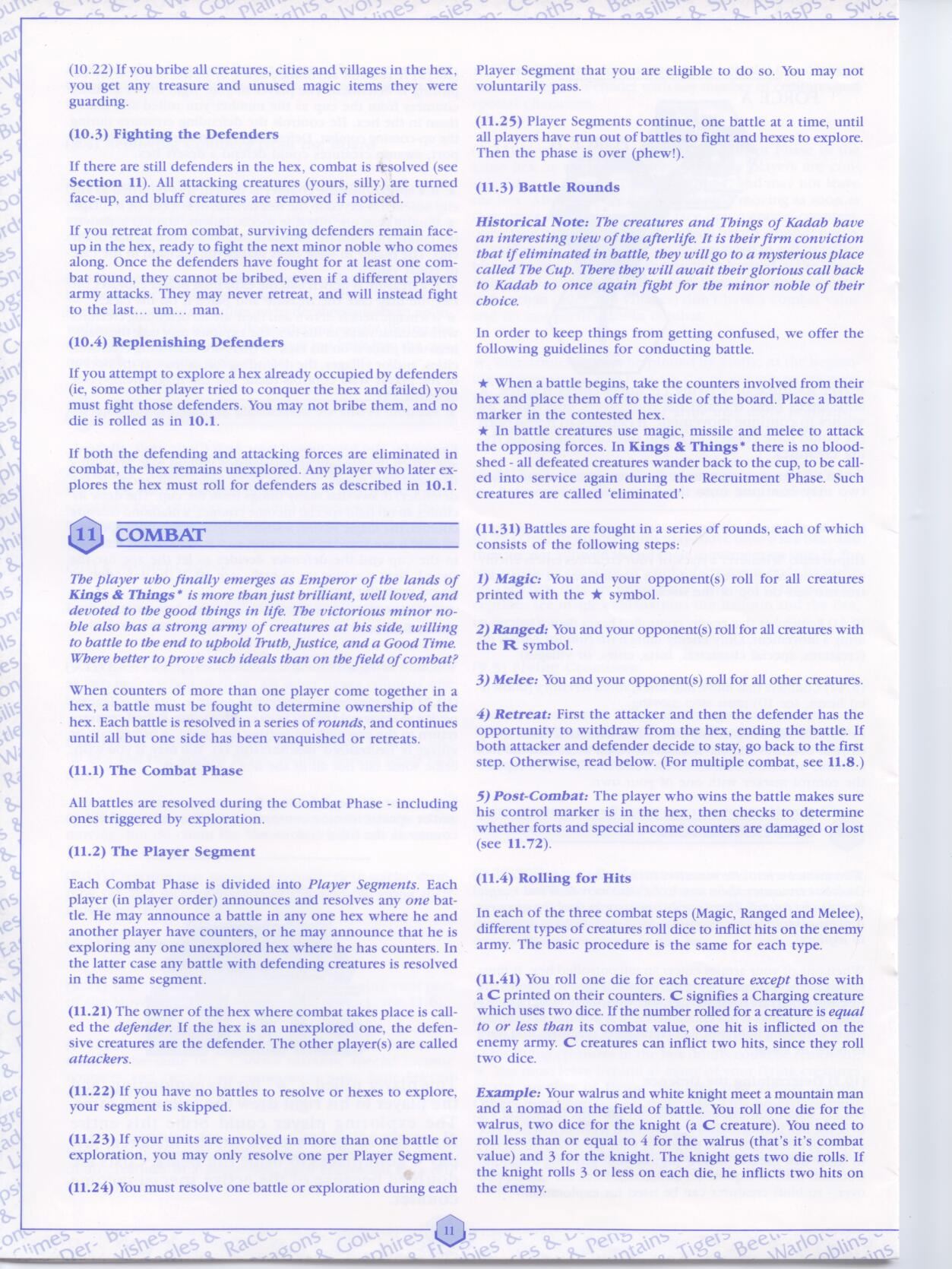


GR-EP-03

GR-EP-02

GR-EP-01

GR-MP-07



GR-CP-04

GR-CP-10

GR-CP-09

GR-CP-08

GR-CP-07

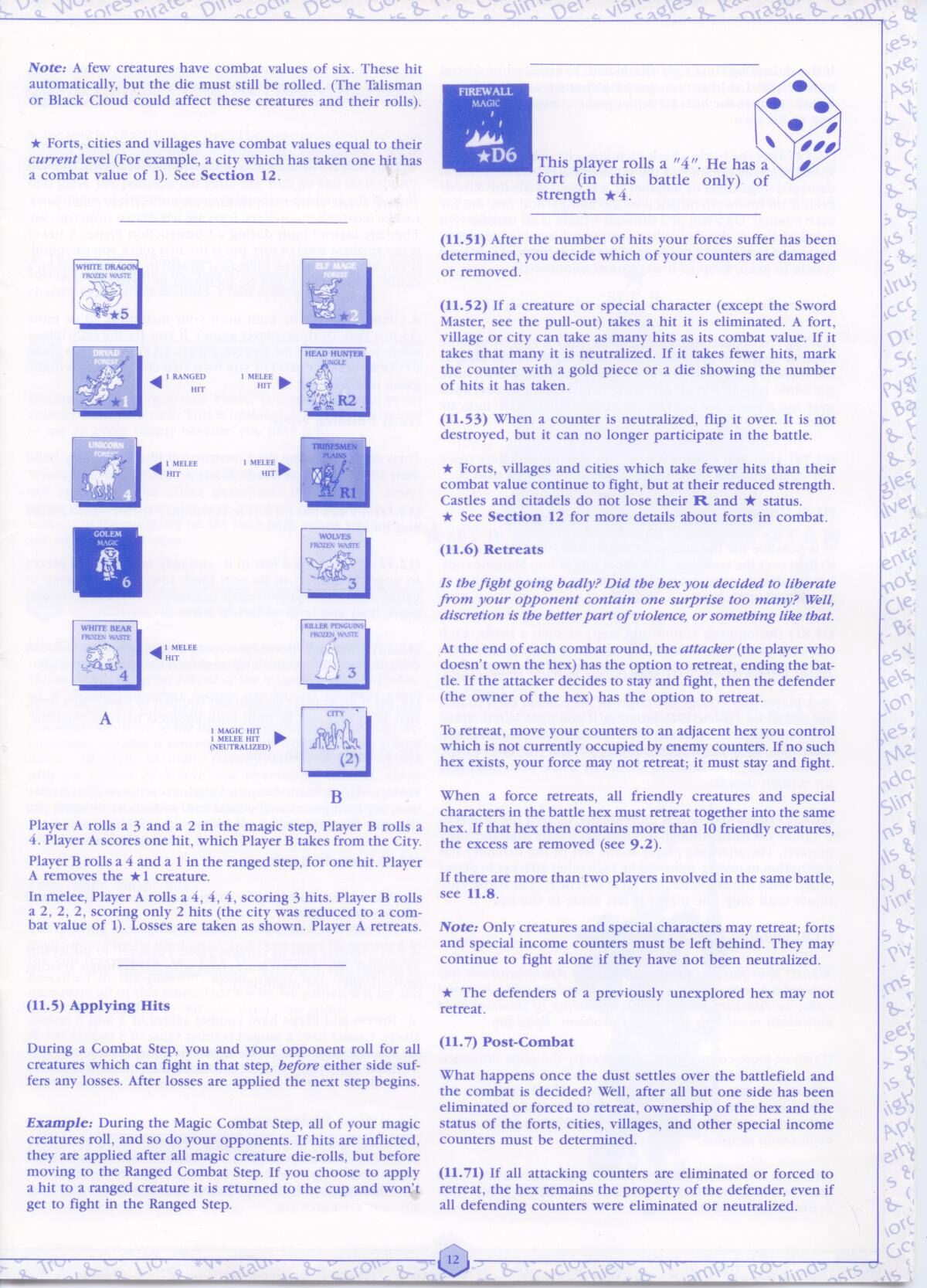
GR-CP-06

GR-CP-05

GR-CP-03

GR-CP-02

GR-CP-01

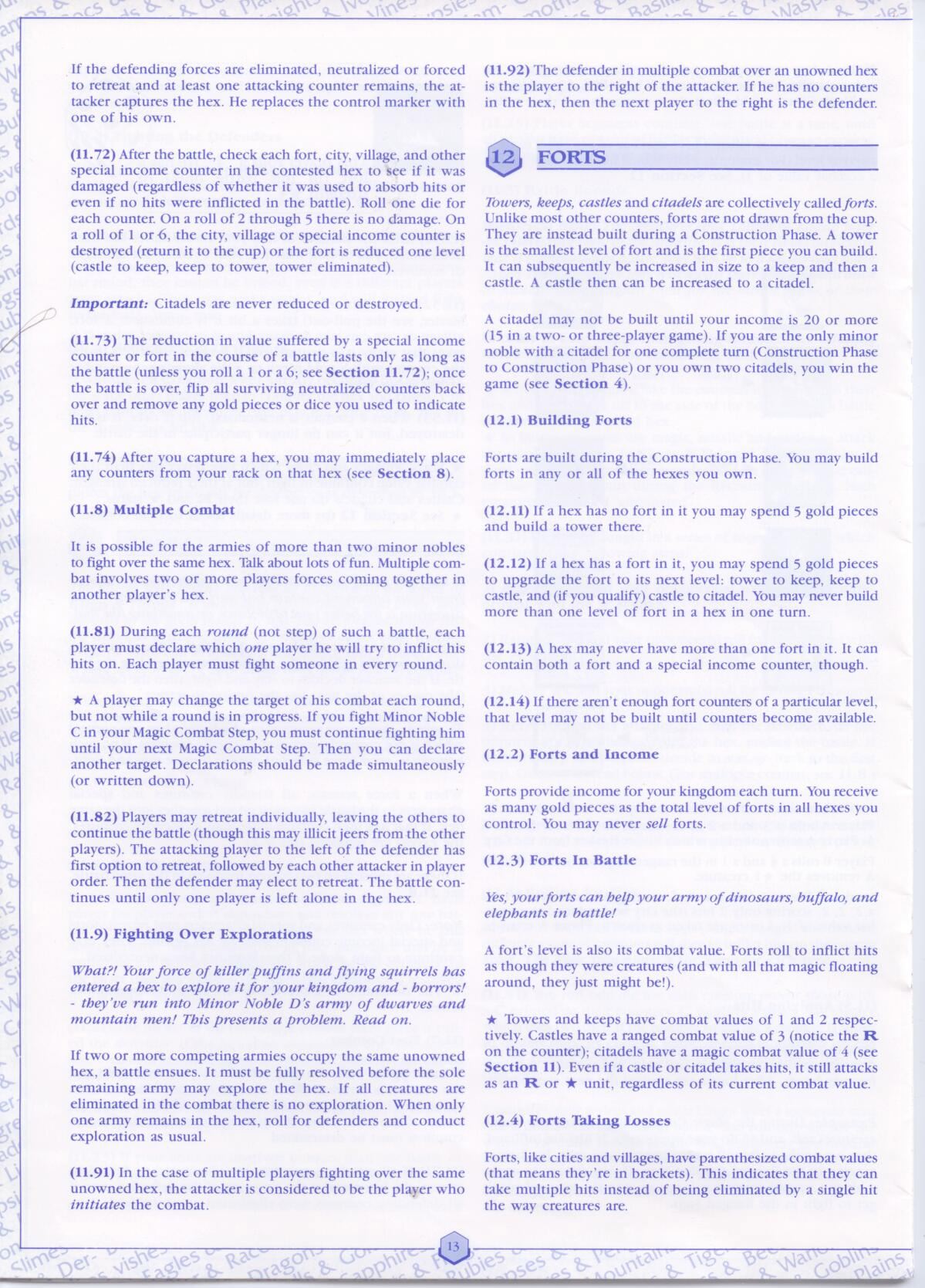


GR-CP-14

GR-CP-13

GR-CP-12

GR-CP-11



GR-FP-06

GR-FP-05

GR-FP-04

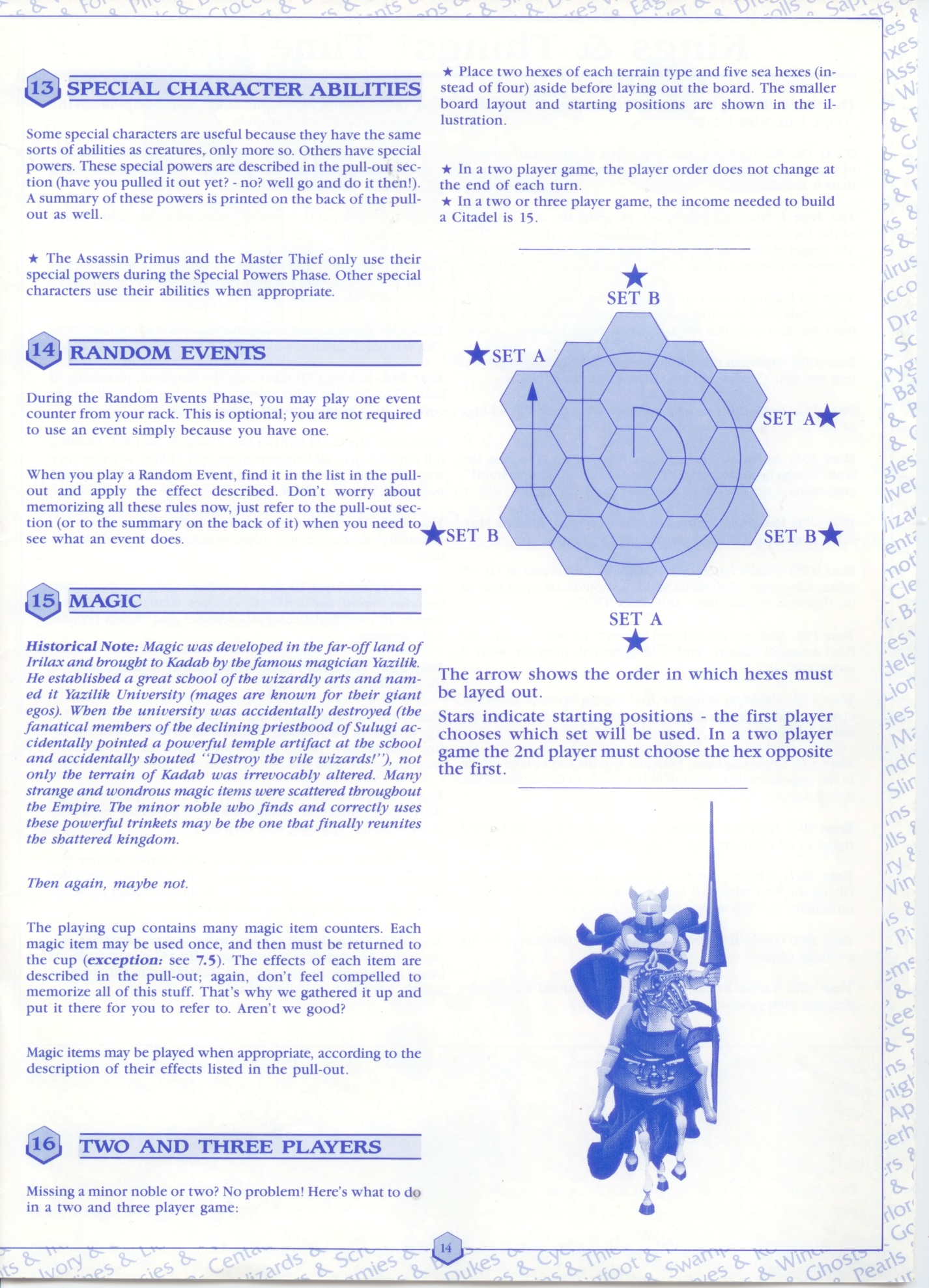
GR-FP-03

GR-FP-02

GR-FP-01

GR-CP-16

GR-CP-15



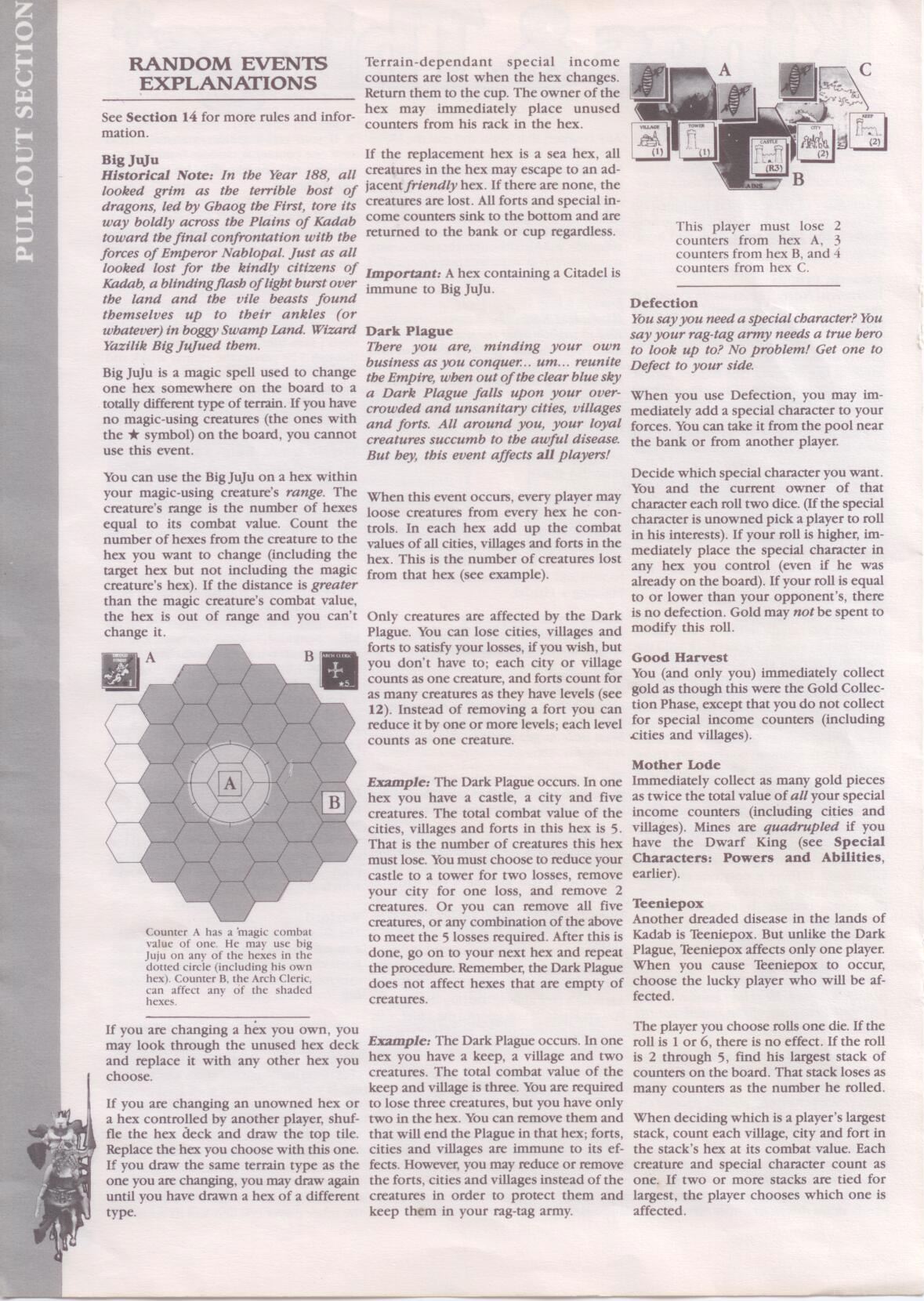
GR-ME-01

GR-RE-01

GR-SCA-01



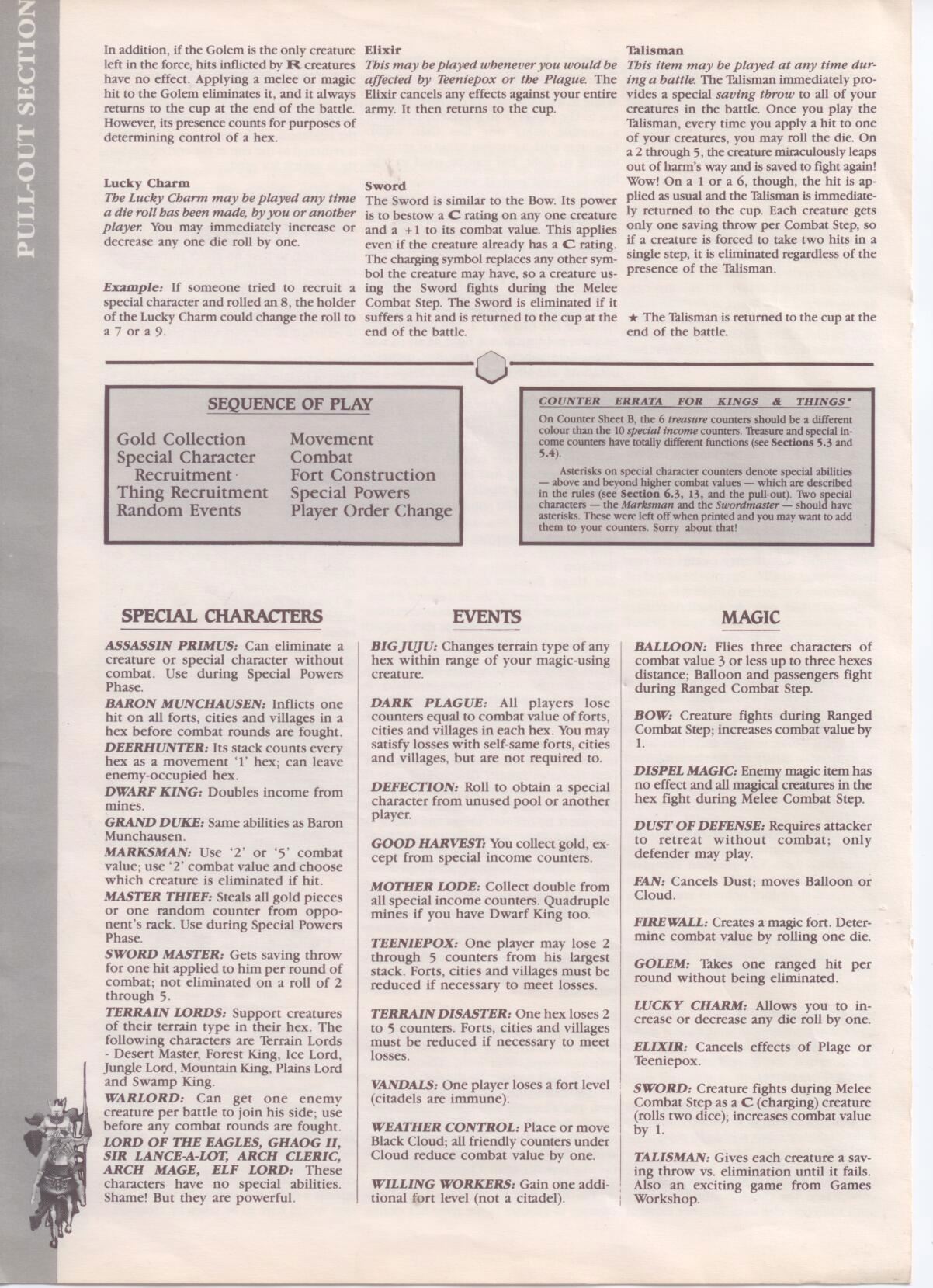
GR-LL-01



GR-LL-02



GR-LL-03



Section 2 - Requirements

2.1 - Functional Requirements

Functional Requirements define the functionality that the system must have, and have been categorized below with relevant traceability to the Game Rules listed in the previous section.

|  |  |  |
| --- | --- | --- |
| ID | Functional Requirement | Traceability |
|  | **Game Setup** |  |
| FR-GS-01 | The System doesn’t allow for games consisting of less than or more than 4 players. | A-01 |
| FR-GS-02 | Players can choose their starting position. | GR-SG-05 |
| FR-GS-03 | Players can choose their two other initial tiles. | GR-SG-06 |
| FR-GS-04 | Players can place a tower on one of the tiles they selected in FR-GS-03 and FR-GS-02. | GR-SG-07 |
| FR-GS-04 | Players can take their initial ‘Things’ from the cup. | GR-SG-07 |
|  | **Turn Sequence** |  |
| FR-TS-01 | Players automatically receive gold based on their current income. | GR-TS-01 |
| FR-TS-02 | Players have the option of trying to recruit a Special Character. | GR-TS-02 |
| FR-TS-03 | Players can draw from the cup and trade in unwanted ‘Things’. | GR-TS-03  A-02 |
| FR-TS-04 | Players may use one of their Random Event counters. | GR-TS-04 |
| FR-TS-05 | Players may move their Creature and/or Special Character ‘Things’ around the board. | GR-TS-05 |
| FR-TS-06 | Players may resolve any combat caused by the movement phase (FR-TS-05). | GR-TS-06 |
| FR-TS-07 | Players may construct additional Forts on tiles they control. | GR-TS-07 |
| FR-TS-08 | Players may use the Special Abilities of Assassin Primus and Master Thief (if they are controlled). | GR-TS-08  A-03 |
| FR-TS-09 | The Player Order must change. | GR-TS-09 |
|  | **Winning A Game** |  |
| FR-WG-01 | The System detects when a Citadel has been built. | GR-GV-01 through GR-GV-04 |
| FR-WG-02 | The System detects when a Player should win the game. | GR-GV-05 through GR-GV-07 |
|  | **Gold Collection** |  |
| FR-GC-01 | The System automatically calculates a Player’s income. | GR-GC-01 |
| FR-GC-02 | The System displays each Player’s gold so every Player can see it. | GR-GC-02 |
| FR-GC-03 | The System automatically distributes gold any Player who uses a Treasure counter. | GR-GC-03 |
|  | **Special Character Recruiting** |  |
| FR-SC-01 | Players have the choice of recruiting one Special Character per turn. | GR-SC-01  A-04  A-05 |
| FR-SC-02 | The System knows when a Special Character’s ability should be activated. | GR-SC-02 |
| FR-SC-03 | The System correctly returns defeated Special Characters to the Bank. | GR-SC-03 |
|  | **Normal Recruiting** |  |
| FR-NR-01 | The System will automatically calculate the number of Free Recruits the Player will receive. | GR-RT-01 |
| FR-NR-02 | The System will prevent the Player from paying for more recruits than they can afford. | GR-RT-02 |
| FR-NR-03 | The System will calculate the amount of recruits a Player receives when they perform a trade. | GR-RT-03 |
| FR-NR-04 | The System will not allow a Player to exceed the Rack Limit. | GR-RT-04 |
|  | **Movement** |  |
| FR-MT-01 | The System calculates the amount of moves each Creature can make per turn. | GR-MP-01 through  GR-MP-03 |
| FR-MT-02 | The System allows the use of Flying Creatures. | GR-MP-06 |
|  | **Exploration** |  |
| FR-EX-01 | The System knows when a Player has entered an unexplored hex. | GR-EP-01 |
| FR-EX-02 | The System determines if the hex is guarded or unguarded. | GR-EP-02  A-06  A-04 |
|  | **Combat** |  |
| FR-CB-01 | The System keeps track of battles that need to be resolved from the Movement or Exploration phase. | GR-CP-01 |
| FR-CB-02 | The System performs the phase using the current Player Order. | GR-CP-04 |
| FR-CB-03 | The Battle phase is broken up into different rounds. | GR-CP-05 through GR-CP-09 |
| FR-CB-04 | Each Player is allowed to choose which of their Creatures receive combat hits. | GR-CP-12 |
| FR-CB-05 | Attacking Players are allowed to retreat from combat. | GR-CP-13 |
|  | **Forts** |  |
| FR-FT-01 | The System automatically subtracts 5 gold from the Player when they build or upgrade a fort. | GR-FP-01  GR-FP-02 |
| FR-FT-02 | Forts can engage in combat during the Combat Phase. | GR-FP-05, GR-FP-06 |
|  | **Interface Requirements** |  |
| FR-IF-01 | A user is allowed to see all of the pieces and tiles he/she owns, as well as the current gold of all other Players. | GR-GC-02 |
| FR-IF-02 | The System displays a roll-able die. | A-07 |
| FR-IF-03 | The interface allows the user to play a local (offline) game, or a networked game. | A-08 |

2.2 - Non-Functional Requirements

The following are non-functional requirements that deal with accessibility, usability, code maintainability, and overall system quality.

|  |  |  |
| --- | --- | --- |
| ID | Non-Functional Requirement | Traceability |
| NFR-01 | The System responds instantaneously when a user performs any actions, such as drawing creatures from the cup, placing creatures on the board, and making general mouse actions. | A-9 |
| NFR-02 | The System knows whether it should connect the users to a network or not. | A-08 |
| NFR-03 | The System follows a precise order when executing the different phases. | A-10 |

2.3 - Assumptions

While developing this system, certain assumptions needed to be made in order to allow the user to interact with the system as smoothly as possible. Many of the functional/non-functional requirements trace back to the following assumptions. Each assumption is listed with an ID as well as the justification for making said assumption.

|  |  |  |
| --- | --- | --- |
| ID | Assumption | Justification |
| A-01 | Games must be played with no less or no more than 4 players. | The 2-3 player game vastly differs in terms of tile setup and Citadel construction, so the system only handles games with exactly 4 players. |
| A-02 | Players can only play Creatures/Special Income during the recruiting normal things phase. | It is very intuitive to be able to play things during the phase that you acquire them. |
| A-03 | A Player who controls Master Thief or Assassin Primus must use the special ability during the Special Ability Phase | The abilities are powerful enough that players should not want to skip using them. |
| A-04 | Only one die is used for rolling | This increases performance since the system does not have to generate more than one random number at a time. |
| A-05 | Players play their newly acquired Special Characters in the preceding phase along with other Creatures. | This allows for a central phase where anything can be played on the board at once (similar to A-02). |
| A-06 | The system rolls a die instead of a Player during the exploration phase. | This allows for a faster exploration phase because the system will not have to wait for another user to roll a die. |
| A-07 | Players will want to see the values being rolled. | Players want to easily be able to see if their roll is better than their opponent’s roll. |
| A-08 | Users want to play on one machine, or across a network. | Sometimes users will all want to take turns on one machine, so they should be given the option to play locally or across a network. |
| A-09 | Players wish to play a game without performance issues. | Modern systems require instantaneous response to prevent angry users. |
| A-10 | The system correctly executes the different phases and waits for any user input. | The different phases must be performed according to the rules. |

Section 3 - Use Cases

3.1 - Use Case Diagram

The following diagram corresponds to the use cases of this system. A larger version of the image is attached as a PDF. The ellipses represent specific use cases, the dashed lines with a <<participates in>> stereotype represents a “participates in” relationship, the dashed lines with an <<enables>> stereotype represents an “enables” relationship, the dashed lines with an <<includes>> stereotype represents an “includes” relationship, and the dashed lines with a <<precedes>> stereotype represents a use case preceding another.



3.2 - Use Cases

Each use case below is given its own ID, as well as a description.

|  |  |
| --- | --- |
| UC-01 | Player Joins Game |
| Description | This use case describes a Player joining a hosted game of Kings n’ Things |
| Actors | Player |
| Triggering Event | Player accesses relevant server |
| Pre-Condition | The host has setup a game and is waiting for *others* players to join |
| Main Sequence | 1. Player selects a hosted game of Kings N Things 2. Player joins the hosted game 3. Game notifies player that s/he has successfully joined the game |
| Post-Condition | The player views the game view |
| Resulting Event | The player waits until game starts |
| Alternative Scenarios | 3a. Player fails to connect to game and its client terminates |
| NFRs | N/A |
| Policies | A-01 |

|  |  |
| --- | --- |
| UC-02 | Player Plays Game |
| Description | This use case describes 4 Players playing a game of Kings n’ Things |
| Actors | 4 Players |
| Triggering Event | Final player has joined the game |
| Pre-Condition | 4 Players have joined a Game |
| Main Sequence | 1. Players roll for player order 2. Players are given their initial gold 3. Players place initial control markers in player order 4. Players place their initial tower in player order 5. All Players receive initial things 6. Players play initial things in player order 7. Repeatedly:    1. Players Perform a Turn of Kings N’ Things    2. Game determines if a Player has won       1. If so, Player wins       2. else loop back to 7a |
| Post-Condition | Game has declared a winner |
| Resulting Event | Game ends |
| Alternative Scenarios | **System unexpectedly terminates**:   * Notify players that the game is over. * Terminate system for all players. |
| NFRs | NFR-03, NFR-01 |
| Policies | All FR-TS  All FR-GS  All Assumptions |

|  |  |
| --- | --- |
| UC-03 | Player Plays a Turn |
| Description | This use case describes 4 players playing a turn within a game of Kings n’ Things |
| Actors | 4 Players |
| Triggering Event | First turn OR previous turn has ended without a winner |
| Pre-Condition | Player order is set |
| Main Sequence | 1. Players acquire gold 2. Players recruit Special Characters 3. Players get things 4. Players play things 5. Players use Random Events 6. Players move things 7. Players enter combat 8. Players construct Forts 9. Players use Special Character Abilities 10. Play order is updated |
| Post-Condition | All players have played another turn |
| Resulting Event | Players are ready to play another turn |
| Alternative Scenarios | N/A |
| NFRs | NFR-03, NFR-01 |
| Policies | All FR-TS |

|  |  |
| --- | --- |
| UC-04 | Players Acquire Gold |
| Description | This use case describes the steps for the players to get gold |
| Actors | 4 Players |
| Triggering Event | Players begin a new turn |
| Pre-Condition | A new turn sequence has begun |
| Main Sequence | 1. In play order, each Player collects gold based on his/her income |
| Post-Condition | Players have received their gold |
| Resulting Event | The system advances to the next phase. |
| Alternative Scenarios | N/A |
| NFRs | NFR-03, NFR-01 |
| Policies | All FR-GC |

|  |  |
| --- | --- |
| UC-05 | Players Recruit Special Characters |
| Description | This use case describes the steps for players to recruit Special Characters |
| Actors | 4 Players |
| Triggering Event | Players have just acquired their gold. |
| Pre-Condition | The collection of Special Characters not in play is not empty. |
| Main Sequence | 1. In player order, each Player selects the Special Character they would like to recruit. 2. In player order, each Player rolls a die. If the value is larger than the combat value of the Special Character, they receive the Special Character. |
| Post-Condition | Players have either succeeded or failed in recruiting a Special Character |
| Resulting Event | Recruiting Special Character phase ends. |
| Alternative Scenarios | **The Collection of Available Special Characters is Empty:**   * Report to all players that no more Special Characters are available for acquisition. * Proceed to UC-07 and carry out game as normal. |
| NFRs | NFR-03, NFR-01 |
| Policies | A-03, A-04, A-05, A-07  All FR-SC |

|  |  |
| --- | --- |
| UC-06 | Players Get Things |
| Description | This use case describes the steps for the players to get things |
| Actors | 4 Players |
| Triggering Event | Players have just tried recruiting Special Characters |
| Pre-Condition | “The cup” is not empty |
| Main Sequence | 1. In play order, each Player obtains things from “the cup” based on the number of hex tiles they control.   1. In play order, each Player can optionally recruit things at the cost of gold. 2. In player order, each Player can return unwanted things to the cup in exchange for new ones. |
| Post-Condition | Players have received their things |
| Resulting Event | Players are ready to play things |
| Alternative Scenarios | **The Cup Is Empty:**   * The System notifies the players that the game is over. |
| NFRs | NFR-03, NFR-01 |
| Policies | All FR-NR  A-02 |

|  |  |
| --- | --- |
| UC-07 | Players Play Things |
| Description | This use case describes the steps for the players to play things. |
| Actors | 4 Players |
| Triggering Event | Players have recruited their things |
| Pre-Condition | UC-07 has been completed but system is still in Recruit phase |
| Main Sequence | 1. In player order, each player takes things from his/her rack and places them on any tiles that s/he owns. |
| Post-Condition | Players’ things have been placed on their tiles. |
| Resulting Event | Players are ready to use their Random Events |
| Alternative Scenarios | N/A |
| NFRs | NFR-01 |
| Policies | A-02 |

|  |  |
| --- | --- |
| UC-08 | Players Use Random Events |
| Description | This use case describes the steps for players to use their Random Events |
| Actors | 4 Players |
| Triggering Event | Players have recruited their things and played relevant pieces on the board. |
| Pre-Condition | At least one player has at least one Random Event that s/he would like to play |
| Main Sequence | 1. In Player order, is allowed to play one Random Event (if s/he has any on his/her rack). 2. The corresponding event happens, based on what Random Event was played. |
| Post-Condition | At least zero Random Events were played. |
| Resulting Event | Players are ready to move their things. |
| Alternative Scenarios | **No player has a Random Event that they would like to play:**   * The Random Event phase ends and the system enters the Movement Phase. |
| NFRs | NFR-01, NFR-03 |
| Policies | FR-TS-04  GR-RE-01  GR-LL-02 |

|  |  |
| --- | --- |
| UC-09 | Players Move Things |
| Description | This use case describes 4 Players moving their things |
| Actors | 4 Players |
| Triggering Event | Players have completed placing their things and playing Random Events |
| Pre-Condition | Things are in play |
| Main Sequence | In Player order,   1. The current player selects things (if any) to move from the currently selected tile. 2. The current player moves the selected things to the designated tile. 3. The system accepts the move to a hex owned by the current player |
| Post-Condition | Things have been moved |
| Resulting Event | System continues to Combat Phase |
| Alternative Scenarios | 2a. The current player moved to an unexplored tile   * System rolls die to see if defenders appear * If defenders appeared, players resolve combat in the Combat Phase   2b. The current player moved to an enemy player’s tile   * Combat will be resolved in the Combat Phase |
| NFRs | NFR-01, NFR-03 |
| Policies | All FR-MT  All FR-EX  A-06 |

|  |  |
| --- | --- |
| UC-10 | Players Enter Combat |
| Description | This use case describes players resolving Combat from the Movement Phase |
| Actors | 4 Players |
| Triggering Event | Combat from the Movement Phase must be resolved |
| Pre-Condition | A Player has unresolved Combat |
| Main Sequence | In each tile with an unresolved battle, repeat the following until the system signals the battle to be complete:   1. If more than two players are involved in the current battle, then the system asks each participant to select another participant of this battle as target. 2. Each participant rolls for each of his/her creatures involved in the current round of combat to determine how many hits s/he inflicts to his/her selected target. 3. For each hit received by a participant, the game asks that participant to select a thing to discard for its army involved in that combat. 4. The game offers each relevant participant the possibility of retreating. 5. The game declares combat to be over in this tile if none or a single participant is left in this tile, in which case:    1. the status of any fort in that tile is resolved |
| Post-Condition | All combats have been resolved |
| Resulting Event | The system continues to the Construction Phase |
| Alternative Scenarios | 2a. Tile has no fort so no fort is placed on the tile  2b. Battle had no victor, so the tile is set to neutral territory |
| NFRs | NFR-01, NFR-03 |
| Policies | All FR-CB  FR-FT-02  FR-IF-02  A-04 |

|  |  |
| --- | --- |
| UC-11 | Players Construct Forts |
| Description | This use case describes 4 Players constructing Forts |
| Actors | 4 Players |
| Triggering Event | Players have resolved all battles |
| Pre-Condition |  |
| Main Sequence | 1. Each player is given the option of either building a fort on a tile that they own, or upgrading an existing fort. |
| Post-Condition | Players have constructed forts |
| Resulting Event | 1. Player wins 2. Players Use Special Character Abilities |
| Alternative Scenarios | 1b. **Player upgrades fort to Citadel**   * System determines if the Player should win |
| NFRs | NFR-01, NFR-03 |
| Policies | FR-FT-01 |

|  |  |
| --- | --- |
| UC-12 | Players Use Special Character Abilities |
| Description | This use case describes 4 Players using their Special Character abilities |
| Actors | 4 Players |
| Triggering Event | Any player owns either Master Thief, Assassin Primus, or both |
| Pre-Condition | Players have constructed Forts and a victor was not found |
| Main Sequence | 1. Each player involved uses the Special Ability as per the rules |
| Post-Condition | Special Abilities have been used |
| Resulting Event | Special Abilities phase ends |
| Alternative Scenarios | **No Player owns Master Thief or Assassin Primus**   * The system continues as normal |
| NFRs | NFR-01, NFR-03 |
| Policies | FR-TS-08  A-03  A-04  GR-SCA-01  GR-LL-01 |

|  |  |
| --- | --- |
| UC-13 | Player Wins |
| Description | This use case describes a Player winning the game |
| Actors | 4 Players |
| Triggering Event | The system has determined that a player has won |
| Pre-Condition | N/A |
| Main Sequence | 1. The system notifies each player who the victor is. 2. The system asks each player if they would like to play again. |
| Post-Condition | The game receives the response from each player |
| Resulting Event | 1. A new game is played as per UC-03 2. Each player disconnects from the system |
| Alternative Scenarios | N/A |
| NFRs | All NFR |
| Policies | All FR-WG |

3.4 – Responsibilities, Unbound Use Case Maps

The following are the Responsibilities of each use case and will be used in each of the following Use Case Maps. The codes found in the ID column can be interpreted as Triggering Event, Responsibility and Resulting Event; for TE, RESP, and RE, respectively.

UUCM-01

Mac:Users:schurman93:Documents:School:COMP3004:kingsandthings:Docs:Iteration Two Documentation:UUCM:UUCM-01.pdf

|  |  |  |
| --- | --- | --- |
| ID | Responsibility | Use Case |
| TE-01 | System starts | UC-01 |
| RESP-01 | User decides whether the game will be hosted locally or across a network. | UC-01 |
| RESP-02 | User connects to the game server | UC-01 |
| RESP-03 | System notifies player that the game is beginning | UC-01 |
| RESP-04 | System reports an error connecting to the server | UC-01 |
| RE-01 | System continues to game | UC-01 |

UUCM-02

Mac:Users:schurman93:Documents:School:COMP3004:kingsandthings:Docs:Iteration Two Documentation:UUCM:UUCM-02.pdf

|  |  |  |
| --- | --- | --- |
| ID | Responsibility | Use Case |
| TE-02A | System begins game | UC-02 |
| TE-02B | System just exited the game due to an unexpected error | UC-02 |
| RESP-05 | The system determines the Player order. | UC-02 |
| RESP-06 | The system gives players initial gold. | UC-02 |
| RESP-07 | The system lets players choose their initial starting kingdoms. | UC-02 |
| RESP-08 | The system lets players place their initial towers. | UC-02 |
| RESP-09 | The system gives players their initial Things. | UC-02 |
| RESP-10 | The system lets players place their initial Things. | UC-02 |
| RESP-11 | The system declares a winner. | UC-02 |
| RESP-12 | System notifies user that an unexpected error has occurred | UC-02 |
| RESP-13 | System terminates the game | UC-02 |
| RE-02 | Player exits the game | UC-02 |

UUCM-03

Mac:Users:schurman93:Documents:School:COMP3004:kingsandthings:Docs:Iteration Two Documentation:UUCM:UUCM-03.pdf

|  |  |  |
| --- | --- | --- |
| ID | Responsibility | Use Case |
| TE-03 | User plays a turn | UC-03 |
| RESP-14 | System updates player order | UC-03 |
| RE-03 | Players are ready to play another turn | UC-03 |

UUCM-04

Mac:Users:schurman93:Documents:School:COMP3004:kingsandthings:Docs:Iteration Two Documentation:UUCM:UUCM-04.pdf

|  |  |  |
| --- | --- | --- |
| ID | Responsibility | Use Case |
| TE-04 | Player begin a new turn | UC-04 |
| RESP-15 | Player collects gold | UC-04 |
| RE-04 | System advances to the next phase | UC-04 |

UUCM-05

Mac:Users:schurman93:Documents:School:COMP3004:kingsandthings:Docs:Iteration Two Documentation:UUCM:UUCM-05.pdf

|  |  |  |
| --- | --- | --- |
| ID | Responsibility | Use Case |
| TE-05 | Players have just acquired gold | UC-05 |
| RESP-16 | Player selects a Special Character to recruit | UC-05 |
| RESP-17 | Player rolls die in attempt to recruit the Special Character | UC-05 |
| RESP-18 | System notifies Players that there are no more Special Characters | UC-05 |
| RE-05 | System advances to the next phase | UC-05 |

UUCM-06Mac:Users:schurman93:Documents:School:COMP3004:kingsandthings:Docs:Iteration Two Documentation:UUCM:UUCM-06.pdf

|  |  |  |
| --- | --- | --- |
| ID | Responsibility | Use Case |
| TE-06 | Players have just finished recruiting Special Characters | UC-06 |
| RESP-19 | Player obtain Things from The Cup | UC-06 |
| RESP-20 | Player purchases Things from the Cup | UC-06 |
| RESP-21 | System notifies player that The Cup is empty | UC-06 |
| RE-06 | System advances to the next phase | UC-06 |

UUCM-07

Mac:Users:schurman93:Documents:School:COMP3004:kingsandthings:Docs:Iteration Two Documentation:UUCM:UUCM-07.pdf

|  |  |  |
| --- | --- | --- |
| ID | Responsibility | Use Case |
| TE-07 | Players have just finished recruiting Things from The Cup | UC-07 |
| RESP-22 | Player places Things on the board | UC-07 |
| RE-07 | System advances to the next phase | UC-07 |

UUCM-08

Mac:Users:schurman93:Documents:School:COMP3004:kingsandthings:Docs:Iteration Two Documentation:UUCM:UUCM-08.pdf

|  |  |  |
| --- | --- | --- |
| ID | Responsibility | Use Case |
| TE-08 | Players just finished placing Things on the board | UC-08 |
| RESP-23 | Player plays a Random Event | UC-08 |
| RESP-24 | System adjusts games state for played Random Event | UC-08 |
| RESP-25 | No players held Random Events | UC-08 |
| RE-08 | System advances to the next phase | UC-08 |

UUCM-09

Mac:Users:schurman93:Documents:School:COMP3004:kingsandthings:Docs:Iteration Two Documentation:UUCM:UUCM-09.pdf

|  |  |  |
| --- | --- | --- |
| ID | Responsibility | Use Case |
| TE-09 | Players just finished playing Random Events | UC-09 |
| RESP-26 | Player selects Things to move | UC-09 |
| RESP-27 | Player selects a tile to move Things to | UC-09 |
| RESP-28 | System moves Things to selected tile | UC-09 |
| RESP-29 | Player moves to an unexplored tile | UC-09 |
| RESP-30 | System rolls die to determine if combat will be declared | UC-09 |
| RESP-31 | Player moves to an enemy tile, system declares combat | UC-09 |
| RE-09 | System advances to the next phase | UC-09 |

UUCM-10Mac:Users:schurman93:Documents:School:COMP3004:kingsandthings:Docs:Iteration Two Documentation:UUCM:UUCM-10.pdf

|  |  |  |
| --- | --- | --- |
| ID | Responsibility | Use Case |
| TE-10 | Players just finished movement phase | UC-10 |
| RESP-32 | Combat is declared, player selects an enemy to attack | UC-10 |
| RESP-33 | Player rolls die for each Creature to determine if they will attack | UC-10 |
| RESP-34 | System asks defending player to select a defending creature | UC-10 |
| RESP-35 | Player retreats from combat | UC-10 |
| RESP-36 | System ends combat, remaining player is given ownership of tile | UC-10 |
| RE-10 | System advances to the next phase | UC-10 |

UUCM-11Mac:Users:schurman93:Documents:School:COMP3004:kingsandthings:Docs:Iteration Two Documentation:UUCM:UUCM-11.pdf

|  |  |  |
| --- | --- | --- |
| ID | Responsibility | Use Case |
| TE-11 | Players just finished combat phase | UC-11 |
| RESP-37 | Player constructs or upgrades a Fort | UC-11 |
| RESP-38 | Player upgraded to a citadel | UC-11 |
| RE-11 | System continues to the next phase | UC-11 |

UUCM-12Mac:Users:schurman93:Documents:School:COMP3004:kingsandthings:Docs:Iteration Two Documentation:UUCM:UUCM-12.pdf

|  |  |  |
| --- | --- | --- |
| ID | Responsibility | Use Case |
| TE-12 | Players just finished construction phase | UC-12 |
| RESP-39 | Player uses a Special Character’s ability | UC-12 |
| RE-12 | System advances to the next phase | UC-12 |

UUCM-13

Mac:Users:schurman93:Documents:School:COMP3004:kingsandthings:Docs:Iteration Two Documentation:UUCM:UUCM-13.pdf

|  |  |  |
| --- | --- | --- |
| ID | Responsibility | Use Case |
| TE-13 | Players have finished taking their turns | UC-13 |
| RESP-40 | System notifies players of the victor | UC-13 |
| RESP-41 | System asks players if they would like to play again | UC-13 |

Section 4 - Design Decision

|  |  |
| --- | --- |
| ID | Design Decision |
| DD-01 | **Use of the Singleton Pattern**  Some classes only needed one instance. Using this pattern allows other classes to access the same instance of a singleton class, without storing it in a local variable, or retrieving it from another class’s local variable. |
| DD-02 | **Multi-threaded dedicated server using the Reactor pattern**  By using a dedicated server, our game allows for many simultaneous online games at the same time. Users do not have to know who they will play with to start a game of Kings and Things, but can join any game, at any time, with random players.  Using a dedicated server and database allows for scalability. For example, the game may be easily ported to a web or mobile application without making any changes to the server. |
| DD-03 | **Special Characters displayed in a grid-like fashion**  This allows for a centralized, intuitive spot for players to see the available Special Characters. |
| DD-04 | **Factory Pattern for Special Characters and various other piece classes**  Certain classes have lots of specific types of instances, needed a central place to create them. |
| DD-05 | **The cup reads in its initial members from a file.**  This method of initializing the contents of the cup was used so that the code would look a little cleaner, and all of the Creatures are located in one file which can be easily modified. |
| DD-06 | **The Player Rack uses the Observer Pattern**  The backend of the Player Rack is a subject and the GUI portion observes the backend so it can be updated whenever a piece is added/removed from the rack. |
| DD-07 | **All playable Things inherit from an abstract Piece class**  Code simplification, ease of maintainability. |
| DD-08 | **The server stores game and user information in a central database**  The server supports a database, which saves the entire state of any game, as well as some user information, while only sending the relevant information to clients. For example, a user’s Player Rack is stored online in the database, but its contents are only sent to the corresponding user, disallowing other users to have any access to that private information.  We understand that a single game of Kings and Things may take a long time to finish. By storing the game state in the online database, users are able to quit the client application, and then resume the same instance of that game at any time. |
| DD-09 | **Use of Combatable and Performable interfaces.**  Some functionality is common among classes that cannot be shared through the use of inheritance. For example, some Piece subclasses like Creatures and Forts can engage in combat while other cannot. Those classes may implement the same Combatable interface. |