# VG101 — Introduction to Computer and Programming

*Project 2*

Manuel — UM-JI (Fall 2018)

- Include simple comments in the code
- Split the code over several functions
- Extensively test your code and impove it
- Start early and respect the milestones
- Update the `README` file for each milestone
- Update the `Changelog` file between two milestones
- Archive the files (`*.{zip|tar}`) and upload on Canvas

## 1 Project setup

After successfully taking the very easy MATLAB midterm Haruka, Kana, and Chiaki, want to know more about programming. So they are thinking of the best way to learn and practice C, when Haruka proposes to implement the game *Legends of the Three Kingdoms* (三国杀)[1] in C.
However as Chiaki feels this task is a bit too hard for beginners, Kana suggests to go with a simplified version. Then Haruka runs to her bedroom and brings back the game. They start by defining simplified rules to implement their game. Lets the fun begin!

### Overview

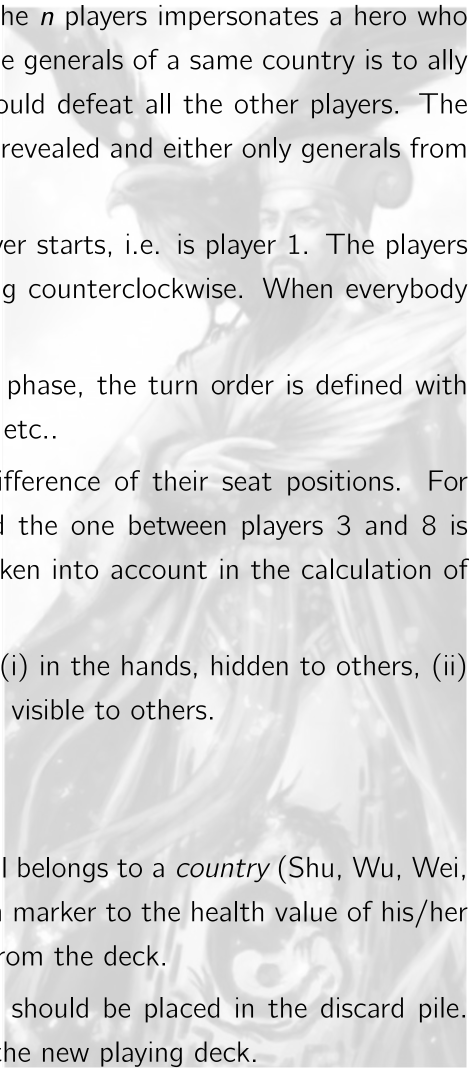
In the variant of the game imagined by the three sisters each of the $n$ players impersonates a hero who can either be *general* of a country or a *mercenary*. The goal for the generals of a same country is to ally together and defeat all other countries, while the mercenaries should defeat all the other players. The game stops as soon as all the generals and mercenaries have been revealed and either only generals from a country are alive or only one mercenary is alive.

The $n$ players are expected to form a circle and the youngest player starts, i.e. is player 1. The players are numbered in increasing order, starting from player 1 and going counterclockwise. When everybody has played, the next round starts, and any dead player is skipped.

For any action during the whole game, including the preparation phase, the turn order is defined with respect to the player's number, i.e. player 1, followed by player 2, etc..

The *distance* between two players is defined as the minimum difference of their seat positions. For example, initially the distance between players 1 and $n$ is 1, and the one between players 3 and 8 is $\min\{5, n-5\}$. If a hero dies his seat is removed and he is not taken into account in the calculation of the distance.

At any time of the game a player can have cards in three places: (i) in the hands, hidden to others, (ii) in the *equipment zone*, visible to others, and (iii) in the *fate zone*, visible to others.



### Preparation

Each player randomly picks a *general* and keeps it secret. A general belongs to a *country* (Shu, Wu, Wei, or Qun) and has an initial *health* value. Each player sets his health marker to the health value of his/her general. After shuffling all the cards, deal each player four cards from the deck.

Once a card is played and its corresponding action is resolved it should be placed in the discard pile. When the deck is empty the discard pile is shuffled and becomes the new playing deck.

[1] Chinese official website: http://www.sanguosha.com/, English official website: http://www.sanguosha.us/

## Gameplay

Each player's turn is divided into four phases:

1. *Fate:* resolve any delayed strategy card in the player's fate zone;
2. *Draw:* draw two cards from the deck;
3. *Play:* play cards;
4. *Discard:* the number of cards in the hands cannot exceed the health level;

During the third phase various actions can be taken in any order by the player: reveal the general, attack another player, recover some health points, and perform special actions.

Until the general is revealed a player does not belong to any country. When a general is revealed the corresponding country is claimed, unless the total number of generals for that country exceeds $n/2$. In that case the player does not belong to any country and becomes a *mercenary* whose goal is to defeat all the other players. Upon claiming a country a player can draw three extra cards from the deck.

A player can attack others following three main strategies: (i) direct using a `Strike` or `Duel` card; (ii) global using an `Arrow barrage` or a `Barbarian invasion` card; and (iii) sneaky using a `Dismantle`, `Snatch`, or `Borrowed sword` card.

When attacked with a `Strike`, `Borrowed sword`, or `Arrow barrage`, card a player should immediately *dodge* the attack using a `Dodge` card or decrease his/her health level by one unit. If following an attack the health of a hero reaches 0, then he is *dying* but can still ask for help: starting from the current player each one announces whether or not he/she wants to play a `Peach` card to save the dying character.

While only a `Peach` can be used to increase the health level of another hero, a player can decide to play a `Peach` or `Wine` card in order to recover one point on his own health. When a hero effectively dies, i.e. nobody helped him and he could not play a `Wine` or `Peach` card, the corresponding player loses. His/Her cards are placed on top of the discard pile and his position seat is removed.

Besides the cards played on other heroes, some cards belong to the fate zone (`Lightning`, Drown in Happiness, and `Starvation`). This special cards are processed at the beginning of each turn.
By default a general can only attack a player at distance one, however this can be adjusted through the use of special cards displayed in the equipment zone (`Bow`, `Binoculars`, and `Horse`).

## Cards

### Basic cards

There are four types of normal cards in the game:

- `Strike:` strike another player; can be avoided using a `Dodge`, otherwise causes −1 health damage; Can only be played once in a turn;
- `Dodge:` dodge a strike; can only be played to counter a `Strike`;
- `Peach:` recover one health point; can be played on another hero only if he is dying;
- `Wine:` recover one health point when dying but incurs +1 damage on the next strike in this turn; Can only be played once in a turn;

### Scroll cards

In the play phase a hero can play any number of strategy cards. They split into three main categories.

Cards that can only be played on another hero:

- `Duel:` challenge another player into a duel during which the two heroes have to play `Strike` cards; The first one not playing `Strike` gets a health damage of one;
- `Dismantle:` discard any card from the equipment zone of another hero;
- `Snatch:` seize any equipment card of another general within the player's range of action;
- `Borrowed sword:` order a hero to strike a player within his range of action; If he refuses he should give his weapon to the current player;

Cards that take effect immediately:

- `Arrow barrage:` all heroes, expect the current player, must play a `Dodge` or receive +1 damage;
- `Barbarian invasion:` all heroes, expect the current player, must play a `Strike` or receive +1 damage;
- `Peach garden:` all heroes, including the current player, regain one unit of health; The new health level cannot exceed its original value'
- `Something for nothing:` draw two cards;
- `Bountiful harvest:` reveal the top $m$ cards from the deck where $m$ is the number of alive heroes. The current hero picks one, followed by all others in the turn order.

Cards are played in the reverse order of appearance in the fate zone:

- `Lightning:` place in your fate zone and draw a card. If it features a spade and a number between 2 and 9 receive +3 damage; otherwise place `Lightning` in the fate zone of the next player;
- `Drown in happiness:` place in the fate zone of another player. During his fate phase this hero draws a card; If it features a heart the hero is not affected; otherwise his loses his play phase;
- `Starvation:` place in the fate zone of a player whose distance is within a range of 1. During his fate phase this hero draws the top card of the deck; If it features a club, the hero is not affected; otherwise he loses his draw phase;

### Equipment cards

When played equipments card must be placed in the equipment zone of the hero playing them.

Equipment cards impact the distance calculation or the range of action:

- `Binoculars:` allows a hero to better observe other players by decreasing the distance by one unit; The minimum distance remains 1;
- `Bow:` weapon allowing a hero to target a player at a distance +2;
- `Horse:` if a hero rides a horse other players see him at distance regular distance +1;

A hero can ride a horse and hold both a bow and binoculars.

## 2 Project tasks and milestones

The project features three milestones, the last one corresponding to the final submission. Each milestone should take about a week to complete. For each milestone students must submit their current code with all the usual relevant files attached as well as with a short `Changelog.txt` file that describes the progress done since the last submission.

### Milestone 1

Tasks to be completed:

- Define the card structure;
- Define the player structure;
- Read the deck composition from a text file or from the standard input:
  - File: use a command line argument (`--file`) to specify the input file; Each line of the file is composed of two fields: the name of the card and its number (e.g. `Snatch 5`, meaning 5 `Snatch` cards);
  - Standard input: display the card name and ask the user for its number; Ask the user if the result should be saved in a file;
- Write a function to randomly initialize the game, i.e. all the parameters are randomly generated;
- Write a function to initialize the game based on the user's choices, i.e. prompt the user for every decision;
- Write a function to calculate the distance between two players; The equipment cards should be taken into account;

### Milestone 2

Tasks to be completed:

- Write a function to shuffle the deck of cards;
- Write a function to deal the cards;

- Write a function to handle cards in the fate zone card;
- Given a player, write a function returning all the cards in his hands, fate zone, and equipment zone;
- List all the other functions you feel should be implemented and implement them;

### Milestone 3

Tasks to be completed:

- Implemented a double linked list to handle all the players;
- Write a demo-mode, i.e. a whole game is played, all the decision are randomly taken, and the screen displays all the details of the game;
- Write a multi-player mode: one-by-one each player should be shown a screen with his cards and the visible cards of the others; The current player should go through all the turn phases, and prompted for some actions until he/she decides to end the turn; Then refresh the screen to hide the current player's information and switch to the next one.
  *Note:* redrawing the screen is done differently on Linux, Mac OSx, and Windows; ensure the program compiles and behaves properly on all systems;
- Proof-read the project and adjust any part that can be improved, e.g. if some parts of milestone 1 could be improved using the linked list from milestone 3 adjust the code accordingly;

### Bonuses

Optional tasks bringing a substantial reward:

- Add new cards with special features to the game and document each addition;
- Implement a stack and use it to handle the deck and fate zone cards;
- Draw the card using ASCII art;
- Split the code into a server and a client; Run the server, allow a number of clients to connect and play together;

## 3 Project submission

Before submitting the project on Canvas ensure the project compiles on JOJ.

- A project that has not been submitted to JOJ will not be graded;
- JOJ will compile the code using the flags `-Werror -pedantic -Wall -Wextra -std=c11`;
- A project that is not compiling on JOJ will not be graded;
- No test case is offered for the project, the goal of JOJ is only to ensure the written code compiles and complies with the C standard;
- If the Canvas submission includes a GUI, please contact the teaching team when uploading the code since it will not compile on JOJ;

# 4 FAQ

This section lists Frequently Asked Questions (FAQ).

1. I have no idea where to start and what to do. Log on Piazza and discuss with other students and the teaching team. Clearly explain what you do not understand, and why you feel stuck, **do not ask for a solution**. If several opinions appear to be valid determine which ones is the best and most reasonable. Document your choices in the README file. Feel free to edit or refine others' questions and answers. To ensure everybody benefits from the question and its answer **no question will be answered if not asked on the project discussion**;
2. I am very busy with the project and do not have time to work on the assignments.
   Change your work strategy: **first solve the assignments and then move on to the project**. Several exercises from the assignments can be partially reused in the project. Directly starting with a hard task is a waste of time. Assignments are designed to help you progress, and milestones have been organised with the assignments in mind;
3. Is claiming a country the same as revealing a general?
   A country is only claimed by the first revealed general;
4. I am expected to get my program to run on all the common Operating Systems (OS), but I am running only one of them. How can I check?
   The only feature that require to call of an OS specific function is the redrawing of the screen. Search how to do it for each of the most common OS (Windows, Mac OSx, and Linux), then use the "family" of `#ifdef`, `#define` instructions to detect the running environment and ensure the correct function is used;
5. How many cards compose a deck?
   There is no special requirement on the number of cards. Just ensure the number of cards "make sense", e.g. one card of each type is not enough for six players. Having a total number of card in the range 70–100 appears as a reasonable choice;
6. Are there any rules to assign the suite and number to a card (top left corner symbols)?
   They should be randomly assigned;
7. How should I provide the location of the file containing the definition of the deck?
   A file location can be expressed using either a relative or an absolute path. As the absolute path is "computer specific" it is not a good idea to defined any absolute path in the program. Therefore in this project only relative path should be used inside the program. A user should however be able to use either an absolute or relative path when providing a file location as a command line argument;
8. Each general has a health value, what should be its range?
   The initial health value should be randomly generated, Commons values are in the range 3–5;