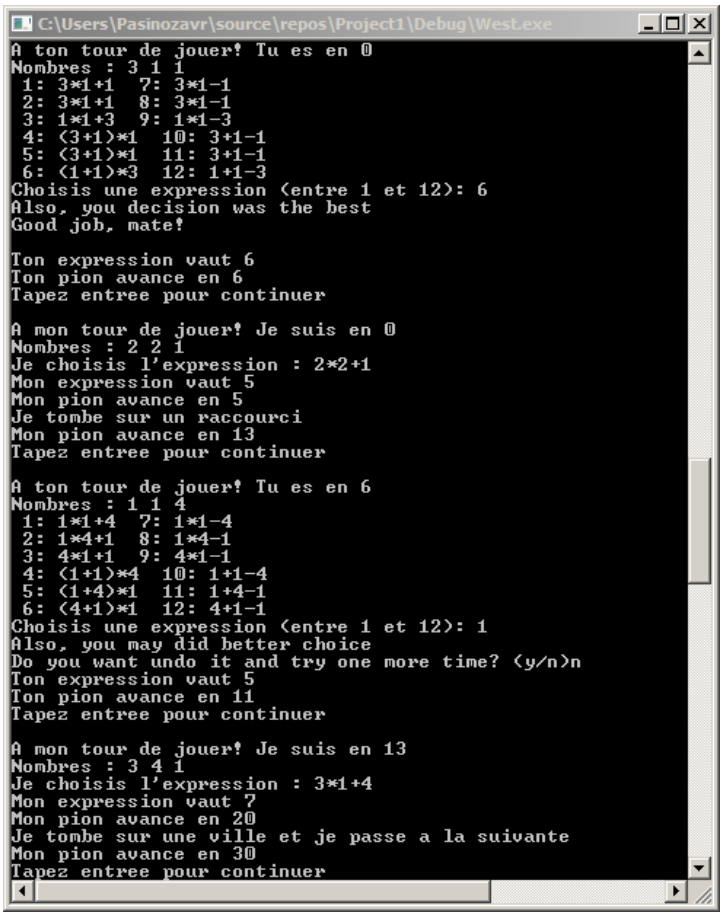
Report for TP “West”



Andriiash Pavlo, *M1 Informatics*

March 20, 2019

User behavior model

During a game, program saves data about each player's decision and forms model frames. I decided to split model into two directions: skill and luck – first is responsible for making good decisions instead of bad ones, second – for using game features such as shortcuts, city bonuses and collisions. Of course, following the logic of the game, player can't make the best decision if he pays no attention to features. But winning also does not always mean choosing the best strategy.

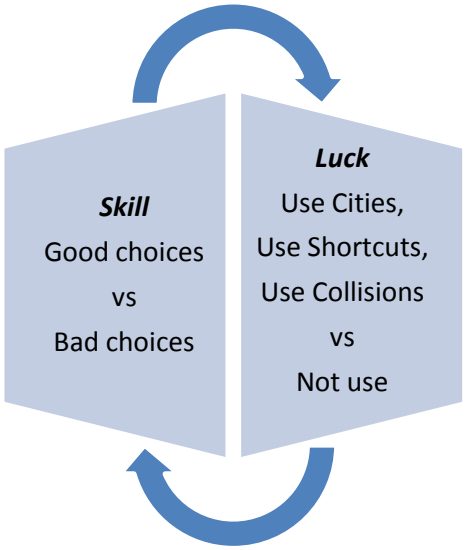


Figure 1. Student model

Advices description

During the game, uses system of advising. It doesn't work if player play its first game (shows notification "Yes, go on, end this first game and I'll show you the truth"). If it's not his first game, it automatically turns on and depending of player's choice shows one of two possible messages:

- "Also, you may did better choice. Do you want undo it and try one more try? (y/n)" for bad decisions (worth than experts)
  In this case player can agree and take or disagree and don't take his second chance
  - If he disagrees, game just goes on
  - If he agrees, system offers new choice showing 12 expressions
  - After this decision system doesn't outputs anything

  If player makes two bad decisions in a row, system simply texts "Also, you may did better choice. But today's without advices, ha-ha". It means it reposes for 1 round when advising is turned off.

- "Good job, mate!" for good decisions (as good as experts)

By this token, system doesn't show "variables" of luck and skill, if it's first game.

Je m'excuse pour l'utilisation de l'anglais

System has storages for number of good and bad decisions (sum of which equals total number) and for number of city visits, shortcuts the way and forced collision. Decidedly collisions are considered bad because of risks.

Values calculating

In this way percentage values of skill and luck are calculated as follows:

$$\boldsymbol{Skill} = \frac{\boldsymbol{GoodDecisionCounter}}{\boldsymbol{TotalDecisionCounter}} * \mathbf{100}$$

Equation 1. Simple and believable formula of skill

$$\boldsymbol{Luck} = \frac{\boldsymbol{CityVisitsCounter}+\boldsymbol{ShortcutsCounter}}{\boldsymbol{CityVisitsCounter}+\boldsymbol{ShortcutsCounter}+\boldsymbol{ColissionsCounter}} * \mathbf{100}$$

Equation 2. Similar formula for luck

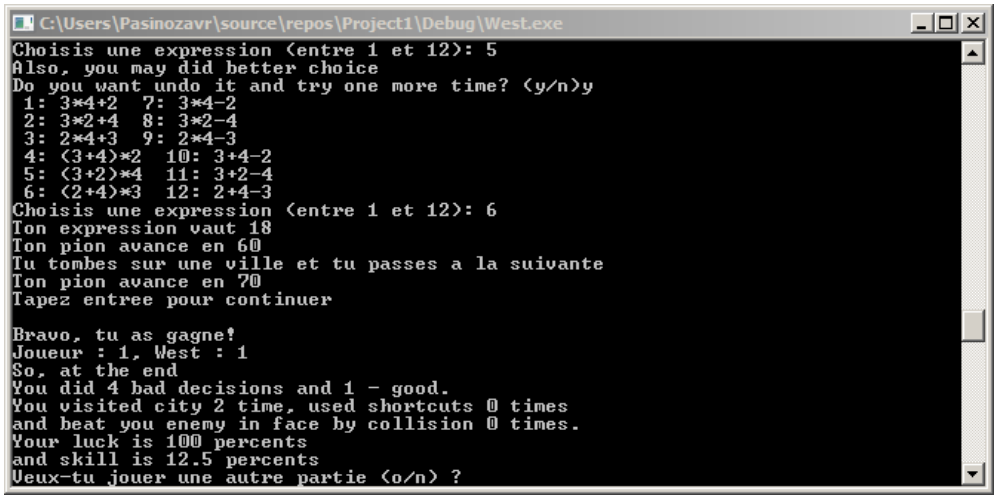


Figure 2. Display values at the end of game loop

Je m'excuse pour l'utilisation de l'anglais