Introduction to Programming (2)

Term Project

Dongwann Kang

Assistant Professor
Department of Computer Science and Engineering

Squid Game

- Design and implement games based on given classes
  - `class Game`
    - Abstract base class
  - `class RedLightGreenLight : public Game`
    - Class for the red light green light game
  - `class RPS : public Game`
    - Class for the rock paper and scissors game
  - `class Player`
    - Class for game player
  - `class PlayerRLGL : public Player`
    - Class for player of the red light green light game
  - `class PlayerRPS : public Player`
    - Class for player of the rock paper and scissors game

https://en.wikipedia.org/wiki/Squid_Game

Squid Game

- Players have their personal abilities
  - `Player::agility` : [0:100]
  - `Player::fearlessness` : [0:100]

- The red light green light game
  - In the game, a ground of which size is 1000 is given
    - `const unsigned int RedLightGreenLight::distance`
  - Initially, the game gives players 20 turns
    - `const unsigned int RedLightGreenLight::turn`

Squid Game

- The red light green light game (cont.)
  - Players move forward on each turn
    - The distance that a player moves depends on player's ability (`agility and fearlessness`)
    - Moving distance = agility + random distance + fearless bonus distance
      - random distance: [0:10]
      - fearless bonus distance
        - » If a player has higher fearlessness, that player will have higher possibility for getting bonus chance
        - » When a player get the bonus chance
          - The player would fall down with 10 percent probability
            - `RedLightGreenLight::fallDownRate`
            - If a player fell down, that player is going to die immediately
          - The player would move further with 75 percent probability
            - fearless bonus distance = agility * (fearlessness * 0.01)
        - » So, fearless player can move further, but it's risky!

Squid Game

- The red light green light game (cont.)
  - When a player moved more than 1000 (distance), that player will escape from the ground and will be alive.
  - After last turn (20, initially), players who are still on the ground will die.

- The rock, paper, scissors game
  - Each player plays the game with (imaginary) staff
    - This game does not require a pair of players
    - You don’t need to implement the staff
  - The players who won the game will be alive

Squid Game

- Fill the blank area of given code to make the program work appropriately
  - `bool PlayerRLGL::act()`
  - `void RedLightGreenLight::play()`
  - Refer to sample runs

- Design and implement more games
  - Original code uses three games

```
std::vector<Game*> games;
games.push_back(new RedLightGreenLight(20));
games.push_back(new RPS());
games.push_back(new RedLightGreenLight(10));
```

  - Instead of the last game (`RedLightGreenLight(10)`) in the given code, use your own game!
    - The `Player` class is designed for single games, so I recommend you to design single-player-based games
    - Amend given codes if required

- File Upload
  - Please upload your **zipped** assignment files to the e-class
    - Include only **.h** and **.cpp** files that you have personally authored
      (Do NOT upload your entire project; Only the relevant source code files are required)
    - Include an executable file (for example, **.exe** file)
    - Include a report (**.pdf** or **.docx**)
      - Source code with comments
      - Captured results and brief analysis

- Deadline
  - 2023/11/18 23:59
  - Late submissions will be rejected

- Academic Integrity
  - Violations of academic honesty, such as plagiarism, will be dealt with seriously
  - While discussions among students are permitted, direct **copying** of another person's work is not
  - Using materials sourced from the internet (e.g., via Google search) and presenting them as your own is also considered **plagiarism**