Emil Milanov, Boyan Hristov

Clean Room Development

Spezification

Game class

1. Method -> Game start; User selects game mode
2. Method -> Game end;

- Preconditions: None

Verifying inputs and setting game states

1. Methode -> Code Make
   - create code
   - verify guess and create hint
   - change turn
2. Method -> Code Break
   - guess code
   - depending on result, improve hope
   - change turn

AI Methods

1. handleGuess()
2. tryBreak()
3. Method -> User gives input
   - depending on mode, codeMake() or codeBreak()
4. Method -> AI gives input
   - same as above
5. Method (callback) request input from player depending on turn
6. Auto

1. Blackbox

```
Game::init()
Game::start()
```

1. Whitebox

```
game = Game()
game.start()
```

2. Blackbox

- Game::start
- Preconditions: game not started
- Postcondition: game winner is known; game is over ## 2. Whitebox

```
class Game:
def start():

    game_mode = get_game_mode()

    if gamemode == programm_breaking
        self.code_break()
    else:
        self.code_make()
```

3. Blackbox

- Game::code_make
- Preconditions:
  - The user choose to be the code breaker
  - Code is not known
- Postconditions:
  - user broke the code OR
  - game is quit

3. Whitebox

```
def code_make():
    code = Game.generate_code()
    game_over = False

    while not game_over:
        guess = player_guess()
        game_over = compare(guess, code)

    print('gg ez')
```

4. Blackbox

- Game::generate_code
- Pre:
  - no code
- Post:
  - code with:
  - 4 chars length, each char symbolizing a color and being a number from 1 to 6
  - no duplicates

4. Whitebox

```
from random import randint

def generate_code():

    colors = [1,2,3,4,5,6]
    code = []

    while len(code) < 4:
        rnd_idx = randint(0, len(colors))
        code.append(colors[rnd_idx])
        colors.remove(colors[rnd_idx])

    return code
```

5. Blackbox

- Game::player_guess()
- Preconditions
  - Code is fixed, only programm knows it
- Postconditions
  1. Player was given a feedback
  2. Player is requested to input a guess
  3. Guess is verified
  - Either
  1. Input is valid (as of given conditions above) OR input is requested again from user
  2. Input from player was received and returned
  - Or
  - Player chooses to quit the game

5. Whitebox

```
def player_guess():
    user_input = input('Enter code or q to quit: ')

    if user_input.lower() == 'q':
        return None
    elif Game.code_valid(user_input):
        return user_input
    else:
        return player_guess()
```

6. Blackbox

- Game::code_valid(input)
- Preconditions
  - User input as string was given
- Postconditions
  - True if code is valid (as of requirements above)
  - False otherwise

6. Whitebox

```
def code_valid(code):
    allowed = [1,2,3,4,5,6]

    if len(code) != 4:
        return False
    elif len(code) == len(set(code)):
        return False
    else:
        for c in code:
            if c not in allowed:
                return False

    return True
```

7. Blackbox

- Game::compare(guess, code)
- Pre
  - guess is a valid code
  - code and guess are of equal length and contain only 1 to 6 chars

- Post:
  - User is given feedback on the code
  - w is placed for each char in code that is both correct and on the right place
  - b is placed for each char in code that is correct but not on the right place
  - . is placed for all wrong characters (non-existant in code)
  - feedback is of length 4, ALWAYS!
  - returns True if guess = code
  - returns False otherwise

7. Whitebox

```
def compare(guess, code):
    w_s = 0
    b_s = 0
    dots = 0

    for i in range(len(guess)):
        if guess[i] == code[i]:
            w_s += 1
        elif guess[i] in code:  # because unique chars in code and guess
            b_s += 1
        else:
            dots += 1

    print('Feedback: {}'.format(('w' * w_s) + ('b' * 'b_s') + ('.' * dots)))

    if w_s == 4:
        return True

    return False
```

N. Blackbox

- code_break()
- Precoditions:
  - The user choose to be the code maker
  - Code is not known
- Postconditions:
  - code is guessed OR game is quit

N. Whitebox

```
def code_break():
```