T1A3 Terminal App

App introduction

Harvest items from farms and cook at home!

Three different locations to explore

- Fruit farm
- Grain farm
- Home

Key components

- Harvest
- Bag
- Storage
- Recipes
- Cook

Landing

```
QQQQQQQQQQQQQQQQQQQQQQQQQQQRNB$OEd4gE&$N0QQQQQQQQQQQQQQQQQQQQQQQQQQQQQQ
QQQQQQQQQQQQQQQQQQQQQN$PS5fJC5Fw32C3#36hgb$0QQQQQQQQQQQQQQQQQQQQQQQQ
    00000000000000$GV5w#3ggmgEbZPPE4dESpwnTfgRQQQQQQQQQQQQQQQQQQQQQ
QQQQQQQQQQQQQQQQQQXJw4XXhqgXbYZYYbYkGGPEXSpfLwpg2VKQQQQQQQQQQQQQQQQQ
QQQQQQQQQQQQQUd#TnyFpp5mXYA&g4AkkkYYkkE53pGgV4hEgg2fd0QQQQQQQQQQQQQQ
0000000000000ULLCFghhdVFyTfEbbYkkkkYZYXTTmgXXXgPZES2yfN0000000000000
QQQQQQQQQQQ0Mw!aT6GEg6mfy2CFh4gGY00kbZPVdgEEdmyCw2h5oLq8QQQQQQQQQQQQQ
QQQQQQQ000RWH#}leu#JfwpdEPEgy5EGkOAOYGXE4dh67]rI7#wCLuopRQQQQQQQQQQQ
QQQQQQQ00RMWK8g!l1*{*107zTygEdVgPYOAkbGgdg63w2m3Ljeauotjn$QQQQQQQQQQQQ
QQQQQQ00RMWD@&kVj}reyqFJnTae36gEGkOAkbPdm5#f6F5yC23u[sIobRQQQQQQQQQQQQ
00000000RMWD@80YZEgGbbGXV2ooy4gEZkAAkZ4mgg6CwFh4gXdfL1JhW000000000000
QQQQQ00RRMNBH@8&AA&&&&ObXqp#mhgbOAOkPhSP4p2GYYYbbbbYAKWMR0QQQQQQQQQQ
QQQQ0000RMMWBDKK@@@@KK@U&YPE62SgbkAOYXhFqSPY8UUU8&&8$KBNR00QQQQQQQQQQ
QQQQQ000RRRMNWWBBBBBBBBK$&kPm5hgZkkkZES2qZUKHDDDHHDDBNMR00QQQQQQQQQQQQ
QQQQQQ00000RRRMMMNNNWBH@8X4hh4XZYYbPEggh&DBWNNNNMMRR00QQQQQQQQQQQQQQ
QQQQQQQ00000000RRRRMMNWDKUdgPGPGbYYbP4qF6DMMRRRRR00000QQQQQQQQQQQQQQQ
QQQQQQQQQQQQQQQQQQQQARMNB@bP4SdXZYk0kbGPXgZMQQQQQQQQQQQQQQQQ
Welcome to farming game!
Harvest ingredients from 2 different farms and cook them at home!
Hit enter to continue
           ted number to make a decision.
Enter a re
(1)Go to Fruit farm
(2)Go to Grain farm
(3)Quit the game
```

Fruit farm / Grain farm

```
🔛 lmh4686@이지혁: ~/assessment/JihyukLee T1A3/src
QQQQQQQQQQQQQQQQQQQQRWD@UAYZg5VXYO&8U@KBMQQQQQQQQ<del>QQQQQQQQQQQQQQQQQQQQQ</del>
QQQQQQQQQQQQQQQQQQW8AA&8&&Ab4pfmydbkOA&AAAkbb&WQQQQQQQQQQQQQQQQQQQQQ
QQQQQQQQQQQQQQQQQDkbbYOOOOkkkOAOA&&A&&AkbGPXbBQQQQQQQQQQQQQQQQQQQQQQ
000000000000000000NbXXPGGZbbYkkkkkk0AAAkYbGgEEggYM000000000000000000
000000000000000000Ug4EggEEEgPPPXPGPXPPPXE4dVhggd4X@000000000000000000
QQQQQQQQQ000000RRRRMMNNWBDDHK@$$$$@KHDBWNMMRR0000Q0QQQQQQQQQQQQQQQQQQ
You are at the Fruit farm
Hit enter to continue
You found Orange(s)!!!
Enter a represented number to make a decision.
(1)Harvest (2)Skip (3)Check bag (4)Check recipes
(5)Go to Grain farm (6)Go to home (7)Ouit the game
```

Harvest

You obtained 1 Orange(s). You have 29 space left. Hit enter to continue

Show what just harvested & amount

Leftover bag space

Joint template

Check bag

Check Recipes

```
Enter a represented number to make a decision.

(1)Harvest (2)Skip (3)Check bag (4)Check recipes

(5)Go to Grain farm (6)Go to home (7)Quit the game

4

DISH NAME | RECIPE |

Apple porridge | {'Apple': 7, 'Wheat': 5, 'Oat': 3, 'Corn': 3} |

Plum porridge | {'Plum': 5, 'Wheat': 4, 'Oat': 2, 'Corn': 1} |

Orange porridge | {'Orange': 8, 'Wheat': 3, 'Oat': 6, 'Corn': 4} |

Mixed porridge | {'Apple': 4, 'Plum': 2, 'Orange': 3, 'Wheat': 2, 'Oat': 3, 'Corn': 1} |

Hit enter to continue
```

Bag is full!!

```
You obtained 1 Wheat(s). You have 1 space left.
Hit enter to continue
You found Oat(s)!!!
Enter a represented number to make a decision.
(1)Harvest (2)Skip (3)Check bag (4)Check recipes
(5)Go to Fruit farm (6)Go to home (7)Quit the game
1
You obtained 1 Oat(s).
Your bag is full! Directing home to empty the bag.
Hit enter to continue
```

Left over space == Obtained amount of item

Empty the bag

```
You are in home
All your items in the bag have been transferred to the storage.
Storage :
   ITEM
           AMOUNT
   Plum
             20
 Apple
             17
             23
 Orange
             33
  Wheat
   Oat
             28
   Corn
             29
Hit enter to continue
You can cook :
2 of Apple porridge
4 of Plum porridge
2 of Orange porridge
4 of Mixed porridge
Enter a represented number to make a decision.
(1)Choose dish to cook (2)Cook later go to farm to harvest (3)Quit the game
```

Cook

```
Enter a represented number to make a decision.

(1)Choose dish to cook (2)Cook later go to farm to harvest (3)Quit the game 1

Enter a represented number to make a decision.

(1)Cook Apple porridge
(2)Cook Plum porridge
(3)Cook Orange porridge
(4)Cook Mixed porridge
1

How many Apple porridge do you want to cook? Max: 2

Congrats!! You cooked 2 Apple porridge!
Now your storage has:
```

After cook

```
Congrats!! You cooked 2 Apple porridge!
Now your storage has :
   ITEM
          AMOUNT
   Plum
             20
 Apple
 Orange
             23
 Wheat
             23
   Oat
             22
             23
   Corn
Hit enter to continue
You can cook:
4 of Plum porridge
2 of Orange porridge
Enter a represented number to make a decision.
(1)Choose dish to cook (2)Cook later go to farm to harvest (3)Quit the game
```

Not enough ingredients, no cook

```
Congrats!! You cooked 1 Orange porridge!
Now your storage has :
   ITEM
           AMOUNT
  Apple
  Orange
  Wheat
             10
   0at
             14
   Corn
Hit enter to continue
You don't have enough ingredients to cook.
Go back to farm and harvest more ingredients.
Enter a represented number to make a decision.
(1).Go to Fruit farm
(2).Go to Grain farm
(3).Quit the game
```

Code walk through

Crucial code

Harvest

- bag_add()
- bag_full()

Cook

get_available_dish()

Common

- get_user_choice()

Harvest process

```
def fruit_farm():
    ascii magic.to terminal(fruit landing)
    print(fruit arv msg)
    joint prompt()
    while True:
        harvested_amount = randint(1, 3)
        discovered_item = choice(fruit_obj)
        main_farm(discovered_item, harvested_amount, grain)
```

Harvest process

```
def main_farm(item, amount, other_farm):
   while True:
        try:
            decision = get user choice(
                f"{obj_discover_msg(item)}\n"
                f"(1)Harvest (2)Skip (3)Check bag (4)Check recipes\n"
               f''(5){other farm} (6)Go to home (7){off}\n",
                ['1', '2', '3', '4', '5', '6', '7'])
        except InputError as err:
            print(err)
        except KeyboardInterrupt:
            keyboard itr msg(7)
        else:
            sub_farm_options(decision, item, amount, other_farm)
            if decision == '1' or decision == '2':
                break
```

Harvest process

```
def sub farm options(
                     user input,
                     harvested item,
                     harvested amount,
                     other farm):
   if user input == '1' and sum(bag.values()) \
                     + harvested_amount < bag_limit:
        bag add(harvested item, harvested amount)
    elif user input == '1' and sum(bag.values()) \
                       + harvested amount >= bag limit:
        bag full(harvested item)
```

```
def bag add(item, amount):
    bag[item] += amount
    if <u>name</u> == ' main_':
        print(f"You obtained {c.CYAN}{amount} {item}(s){r}. "
              f"You have {c.GREEN}{bag space()}{r} space left.")
        joint prompt()
    else:
        return bag[item]
def bag full(item):
    print(f"You obtained {c.CYAN}{bag space()} {item}(s){r}.\n"
          f"Your {c.RED}bag is full{r}! "
          f"{c.GREEN}Directing home{r} to empty the bag.")
    bag[item] += bag space()
    joint prompt()
    return home()
```

```
def get available dish():
    for item in recipes:
       for name, recipe in item.items():
           for grocery, amount in recipe.items():
               if storage[grocery] < amount:</pre>
                   available dish[name] = 0
                   break
           else:
               for grocery, amount in recipe.items():
                   grocery_quotients.append(storage[grocery] // amount)
               available dish[name] = min(grocery quotients)
               grocery quotients.clear()
recipes = (
    {"Apple porridge": {
        "Apple": 7, "Wheat": 5, "Oat": 3, "Corn": 3}},
    {"Plum porridge": {
        "Plum": 5, "Wheat": 4, "Oat": 2, "Corn": 1}},
    {"Orange porridge": {
        "Orange": 8, "Wheat": 3, "Oat": 6, "Corn": 4}},
    {"Mixed porridge": {
        "Apple": 4, "Plum": 2, "Orange": 3, "Wheat": 2, "Oat": 3, "Corn": 1}}
```

String input

```
try:
    decision = get user choice(
        f"{decision temp}\n"
        f"(1)Choose dish to cook (2)Cook later go to farm to harvest "
        f"(3){off}\n",
        ['1', '2', '3'])
except KeyboardInterrupt:
    keyboard itr msg(3)
except InputError as err:
    print(err)
```

get_user_choice()

```
def get_user_choice(prompt, options):
    user_input = input(prompt)
    if isinstance(options, list) and user_input not in options:
        raise InputError(user_input)
    elif isinstance(options, range) and user_input not in options:
        if int(user_input) > max(options):
            raise ExcessError
        elif int(user_input) < 1:
            raise RangeError
    return user_input</pre>
```

Int input

```
try:
    food amount = int(
        get user choice(
            f"How many {c.GREEN}"
            f"{printed dish[food num]}{r} "
            f"do you want to {c.GREEN}cook{r}? "
            f"Max: {c.RED}{max_dish_num}{r}\n",
            range(1, \max dish num + 1, 1))
except (ValueError, RangeError):
    print(f"{b.RED}{c.WHITE}Please enter a positive "
         f"integer bigger than zero.{r}")
except KeyboardInterrupt:
    print(key itr msg)
except ExcessError:
    print(f"{b.RED}{c.WHITE}The maximum available amount "
          f"for this dish is {max_dish_num}.{r}")
```

