1. Pseudo code
   1. 실행환경을 윈도우10이라고 가정하고 콘솔 화면 크기를 세팅한다. (가로 120, 세로 25)
   2. While (1) {

User Interface 출력

선택된 아이템 색깔 변경

각종 정보 출력 (돈, 가격, 잔량, UI, 메시지)

사용자로부터 입력을 받고

If (입력값 == ‘q’) {

종료

} else if (입력값 == ‘j’) {

좌측 아이템 선택

If (더 이상 좌측 아이템이 없으면 가장 우측 아이템 선택)

} else if (입력값 == ‘l’) {

우측 아이템 선택

If (더 이상 우측 아이템이 없으면 가장 좌측 아이템 선택)

} else if (입력값 == ‘I’) {

돈++

} else if (입력값 == ‘space’) {

아이템 구매 (잔량 -1, 돈 -1)

If (돈이 부족하면)

메시지 출력

If (재고가 없으면)

메시지 출력

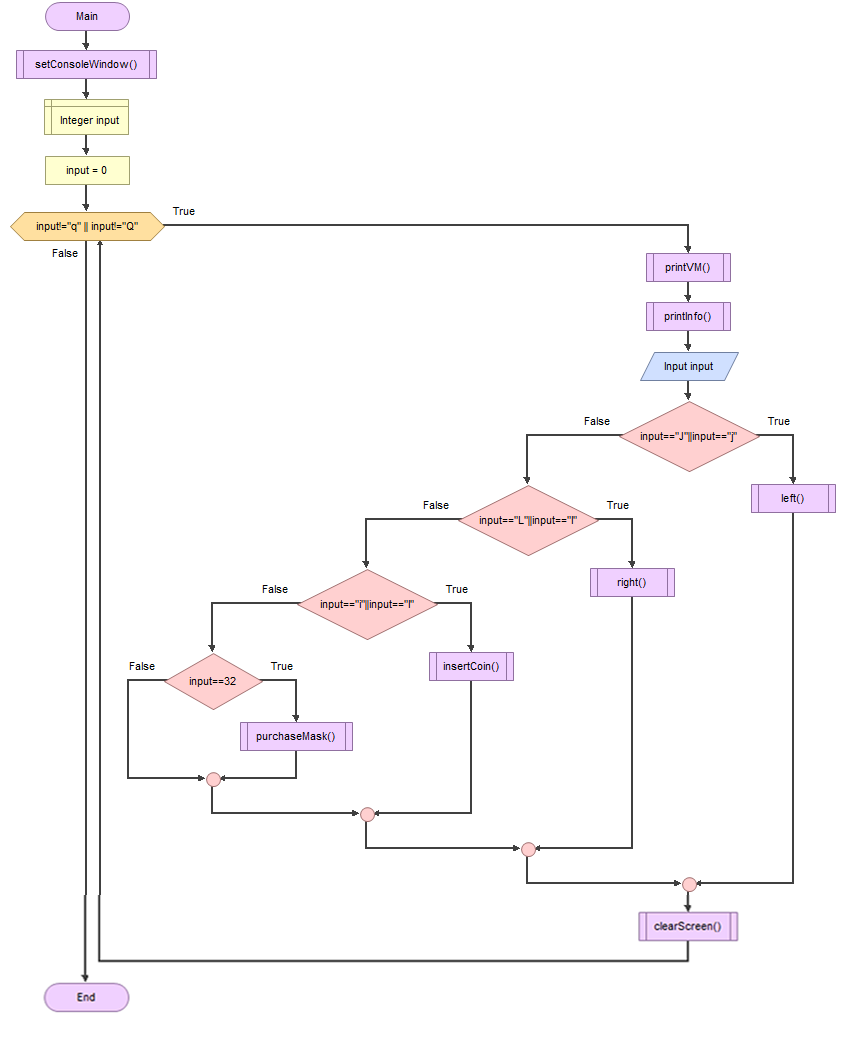
}

화면 초기화

}

* 1. 종료

1. Flowchart



1. Source code

|  |
| --- |
| main.cpp |
| /\*  프로그램 : 마스크 자판기  개발자 : 송훈일  개발기간 : Dec 16, 2020 ~ Dec 18, 2020  문의 : https://www.youtube.com/channel/UC5yZGtDDMZDe3jmCDUB\_rJA  \*/  #include "helper.h"  #include "UI.h"  #include "info.h"  int main(void)  {  // 콘솔창 화면크기 셋팅  setConsoleWindow();  /\*  while (1) {  자판기 출력  사용자 정보 출력  if (input) {  좌우 화살표로 제품 선택(선택된 제품 색깔 변경)  spacebar 로 제품 구매  구매 후 제품 재고 확인  재고가 없으면 sold out 출력  i 로 돈 받기  q 로 종료  }  화면 초기화  }  \*/  while (1) {  printVM();  printInfo();  int c = zeroBufferGetchar();  if (c == 'q' || c == 'Q') {  break;  }  else if (c == 'J' || c == 'j') {  info.pLeft();  }  else if (c == 'L' || c == 'l') {  info.pRight();  }  else if (c == 'i' || c == 'I') {  info.pInsertCoin();  }  else if (c == 32) { // space  info.pPurchaseMask();  }  clearScreen();  }  return 0;  } |

|  |
| --- |
| info.h |
| /\*  \* 프로그램이 동작할 때 필요한 모든 정보와 상수들을 정의한 곳  \*/  #pragma once  #define LENGTH 64  #define ITEMS 3  #define ITEM\_1 0  #define ITEM\_2 1  #define ITEM\_3 2  #define INVENTORY 20  #define PRICE\_1 3000  #define PRICE\_2 5000  #define PRICE\_3 20000  #define uint unsigned int  // 구조체 안에 함수포인터를 통해 OOP의 클래스 메소드마냥 동작하기  typedef struct info {  uint inventories[ITEMS];  uint prices[ITEMS];  uint money;  uint selected;  char message[LENGTH];  uint (\*pGetInventory)(uint);  void (\*pSetInventory)(uint);  uint (\*pGetPrice)(uint);  uint (\*pGetSelected)();  uint (\*pGetMoney)();  void (\*pSetMessage)(const char\*);  char\* (\*pGetMessage)();  void (\*pLeft)();  void (\*pRight)();  void (\*pInsertCoin)();  void (\*pPurchaseMask)();  } Info;  extern Info info;  uint getInventory(uint);  void setInventory(uint);  uint getPrice(uint);  uint getSelected();  uint getMoney();  void setMessage(const char\*);  char\* getMessage();  void left();  void right();  void insertCoin();  void purchaseMask(); |

|  |
| --- |
| info.cpp |
| #include "info.h"  Info info = {  {INVENTORY, INVENTORY, INVENTORY},  {PRICE\_1, PRICE\_2, PRICE\_3},  0,  ITEM\_1,  {"돈을 어서 넣어주세요 고객님"},  getInventory,  setInventory,  getPrice,  getSelected,  getMoney,  setMessage,  getMessage,  left,  right,  insertCoin,  purchaseMask  };  uint getInventory(uint n) {  return info.inventories[n];  }  void setInventory(uint n) {  info.inventories[info.selected] = n;  }  uint getPrice(uint n) {  return info.prices[n];  }  uint getSelected() {  return info.selected;  }  uint getMoney() {  return info.money;  }  void setMessage(const char\* p) {  char\* pMessage = info.message;  // 메세지 초기화  while (\*pMessage) {  \*(pMessage++) = '\0';  }  pMessage = info.message;  // 메세지 복사  while (\*p) {  \*(pMessage++) = \*(p++);  }  }  char\* getMessage() {  return info.message;  }  void left() {  if (!info.selected) {  info.selected = ITEM\_3;  return;  }  info.selected -= 1;  }  void right() {  info.selected += 1;  info.selected %= (ITEM\_3 + 1);  }  void insertCoin() {  info.money += 10000;  }  void purchaseMask() {  if (info.money >= info.prices[info.selected] && info.inventories[info.selected] > 0) {  info.money -= info.prices[info.selected];  info.inventories[info.selected]--;  // 구매 성공 메세지  info.pSetMessage("고객님 구매해주셔서 감사합니다.");  }  else if (info.inventories[info.selected] == 0) {  // 안내창에 경고 메세지  info.pSetMessage("Sold out.");  }  else {  // 안내창에 경고 메세지  info.pSetMessage("돈이 부족합니다 고객님, 어서 돈을 넣어주세요.");  }  } |

|  |
| --- |
| sprites.h |
| #pragma once  extern char vendingMachine[]; |

|  |
| --- |
| sprites.cpp |
| // 1 2 3 4 5 6 7 8 9 0 1 2  //3456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890  // s e s e s e  char vendingMachine[] = "\  ====================== # # # ##### # # # # # # ======================\  ====================== ## ## # # # # # # # # ## ## ======================\  ====================== # # # # # # # # # # # # # # # ======================\  ====================== # # # # # ##### ### # # # # # ======================\  ====================== # # ####### # # # # # # # ======================\  ====================== # # # # # # # # # # # # ======================\  ====================== # # # # ##### # # # # # ======================\  xxxxxxx \  x........x \  x;..........x \  x. .....x \  x.. ....x \  X X x.. .....x \  XXXX XXXX X. X x x x.. .......x \  x xx xXXXxxxxxX X x x.. xxxx x X x x.........x x \  x x x x .........x x x...........xxxxxxxxxxxxx x x x........x x \  x x x x ............. X x x .................. . x x x x....xx x \  x xx x ............ X x X........................x x x ........x x \  x x x .............. x x x: ....................x x xx..........x x \  x X ............ x x Xx.................., X x x.............x x \  x X .......... x X x xx ............... :x x x...........x x \  x X x xx x X x x....... ..' : x x x.......;x x \  x x xxxxxxxxxxxx x xxxxxxx x \  x x x x \  xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx \  "; |

|  |
| --- |
| helper.h |
| #pragma once  #include <windows.h>  #include <stdio.h>  #include <conio.h>  #include <stdlib.h>  #include "sprites.h"  // 콘솔창의 가로 세로  #define WINDOW\_WIDTH 120  #define WINDOW\_HEIGHT 25  // error codes  enum errors {  NO\_ERR = 0,  ERR\_SAVE\_TTY\_GETATTR = -2,  ERR\_SAVE\_TTY\_SETATTR = -3,  ERR\_RESTORE\_TTY\_SETATTR = -4  };  // 헬퍼 함수들은 바깥에서 쓸 수 있도록 extern처리.  extern int setConsoleWindow();  int save\_tty\_attributes();  int restore\_tty\_attributes();  extern int zeroBufferGetchar();  extern void setRed();  extern void setCyan();  extern void setBlue();  extern void reset();  extern void clearScreen(); |

|  |
| --- |
| helper.cpp |
| #include "helper.h"  static DWORD stdin\_mode;  int setConsoleWindow()  {  HANDLE hStdout = GetStdHandle(STD\_OUTPUT\_HANDLE);  CONSOLE\_SCREEN\_BUFFER\_INFO csbiInfo;  SMALL\_RECT srctWindow;  // Get the current screen buffer size and window position.  if (!GetConsoleScreenBufferInfo(hStdout, &csbiInfo))  {  printf("GetConsoleScreenBufferInfo (%d)\n", GetLastError());  return 0;  }  // 변경 가능한 콘솔창의 최대 크기 정보 가져오기  //printf("maximum window size x : %d \n", csbiInfo.dwMaximumWindowSize.X);  //printf("maximu window size y : %d \n", csbiInfo.dwMaximumWindowSize.Y);  srctWindow = csbiInfo.srWindow;  srctWindow.Bottom = WINDOW\_HEIGHT + 10;  srctWindow.Right = WINDOW\_WIDTH - 1;  //SetConsoleScreenBufferSize(hStdout, csbiInfo.dwSize \* 2);  // Set srctWindow to the current window size and location.  if (!SetConsoleWindowInfo(  hStdout, // screen buffer handle  TRUE, // absolute coordinates  &srctWindow)) // specifies new location  {  printf("SetConsoleWindowInfo (%d)\n", GetLastError());  return 0;  }  return 1;  }  // 각 OS에 따른 터미널 조작  int save\_tty\_attributes() {  #ifdef \_WIN32  HANDLE hstdin = GetStdHandle(STD\_INPUT\_HANDLE);  if (!GetConsoleMode(hstdin, &stdin\_mode))  return ERR\_SAVE\_TTY\_GETATTR;  if (hstdin == INVALID\_HANDLE\_VALUE || !(SetConsoleMode(hstdin, 0)))  return ERR\_SAVE\_TTY\_GETATTR; /\* Failed to disable buffering \*/  #elif \_\_APPLE\_\_  struct termios tty\_attr;  if (tcgetattr(STDIN\_FILENO, &tty\_attr) < 0)  return ERR\_SAVE\_TTY\_GETATTR;  tty\_attr.c\_lflag &= ~ICANON;  tty\_attr.c\_lflag &= ~ECHO;  if (tcsetattr(STDIN\_FILENO, 0, &tty\_attr) < 0)  return ERR\_SAVE\_TTY\_SETATTR;  #endif  return NO\_ERR;  }  // 각 OS에 따른 터미널 기본값으로 다시 설정  int restore\_tty\_attributes() {  #ifdef \_WIN32  HANDLE hstdin = GetStdHandle(STD\_INPUT\_HANDLE);  if (!SetConsoleMode(hstdin, stdin\_mode))  return ERR\_RESTORE\_TTY\_SETATTR;  #elif \_\_APPLE\_\_  struct termios tty\_attr;  if (tcgetattr(STDIN\_FILENO, &tty\_attr) < 0)  return ERR\_SAVE\_TTY\_GETATTR;  tty\_attr.c\_lflag |= ICANON;  tty\_attr.c\_lflag |= ECHO;  if (tcsetattr(STDIN\_FILENO, 0, &tty\_attr) < 0)  return ERR\_RESTORE\_TTY\_SETATTR;  #endif  return NO\_ERR;  }  // 터미널에서 입력 버퍼를 없애고 입력을 screen에 표시 없이 변수에 저장해 사용  int zeroBufferGetchar() {  int c = 0;  int res = 0;  if ((res = save\_tty\_attributes()) < 0) {  printf("error : %d", ERR\_SAVE\_TTY\_GETATTR);  return res;  }  c = getchar();  if ((res = restore\_tty\_attributes()) < 0) {  printf("error : %d", ERR\_SAVE\_TTY\_GETATTR);  return res;  }  return c;  }  //  // ANSI escape codes  //  void setRed() {  printf("\033[1;31m");  }  void setCyan() {  printf("\033[1;36m");  }  void setBlue() {  printf("\033[1;34m");  }  void reset() {  printf("\033[0m");  }  void clearScreen() {  system("cls");  } |

|  |
| --- |
| UI.h |
| #pragma once  extern void printVM();  extern void printInfo(); |

|  |
| --- |
| UI.cpp |
| #include "helper.h"  #include "info.h"  // vendingMachine sprite 출력  void printVM() {  for (int i = 0, j = 0; i < WINDOW\_HEIGHT \* WINDOW\_WIDTH; ) {  // 현재 선택된 아이템에 따라 색깔을 주고  if (i % WINDOW\_WIDTH >= 5 && i % WINDOW\_WIDTH <= 37  && j >= 7 && j <= 23 && info.pGetSelected() == ITEM\_1)  setCyan();  else if (i % WINDOW\_WIDTH >= 39 && i % WINDOW\_WIDTH <= 73  && j >= 7 && j <= 23 && info.pGetSelected() == ITEM\_2)  setBlue();  else if (i % WINDOW\_WIDTH >= 75 && i % WINDOW\_WIDTH <= 108  && j >= 7 && j <= 23 && info.pGetSelected() == ITEM\_3)  setRed();  // vendingMachine[i++]  printf("%c", \*((char\*)vendingMachine + i++));  if (!(i % WINDOW\_WIDTH)) {  j++;  printf("\n");  }  reset();  }  }  // 각종 정보 출력  void printInfo() {  printf("%30s%32s%37s\n", "의료용 마스크", "KF-95마스크", "복면마스크");  printf("<-J%117s\n", "L->");  printf("%6s\  %6s%2d%6s%6s%4d\  %6s\  %8s%2d%6s%9s%4d\  %7s\  %8s%2d%7s%8s%5d\n",  "",  "잔량 : ", info.pGetInventory(ITEM\_1), "", "가격 : \\", info.pGetPrice(ITEM\_1),  "",  "잔량 : ", info.pGetInventory(ITEM\_2), "", "가격 : \\", info.pGetPrice(ITEM\_2),  "",  "잔량 : ", info.pGetInventory(ITEM\_3), "", "가격 : \\", info.pGetPrice(ITEM\_3));    printf("\n%15s : \\%d%5s%83s", "투입된 돈", info.pGetMoney(), "", "돈 넣기 : i, 구매 : spacebar");  printf("%20s%60s%40s", "", info.pGetMessage(), "");  } |

1. 배포
   1. 실행 영상 : <https://youtu.be/UUxt6wQskPI>
   2. 이미지

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명