github주소 : https://github.com/freean2468/mask\_vending\_machine

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

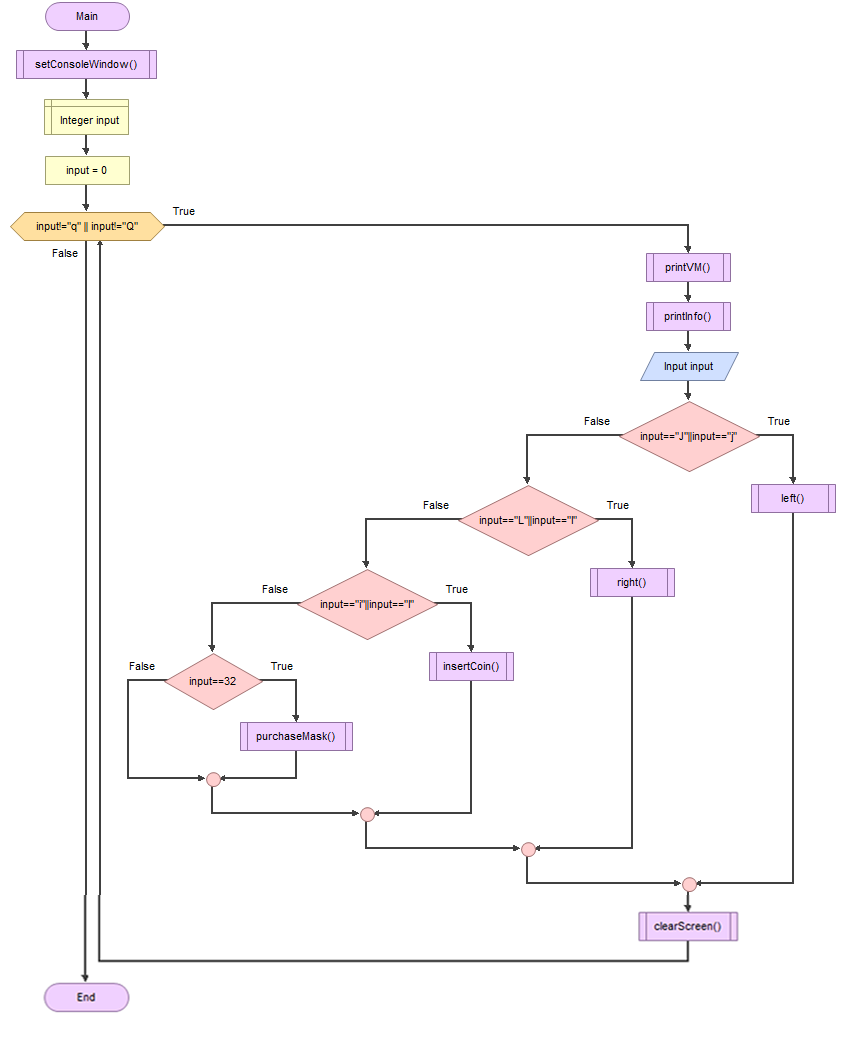
}

화면 초기화

}

* 1. 종료

1. Flowchart



1. Source code

|  |
| --- |
| main.cpp |
| /\*  프로그램 :마스크자판기  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)  {  // 콘솔창화면크기셋팅  setConsoleWindow();  /\*  while (1) {  자판기출력  사용자정보출력  if (input) {  좌우화살표로제품선택(선택된제품색깔변경)  spacebar 로제품구매  구매후제품재고확인  재고가없으면 sold out 출력  i로돈받기  q 로종료  }  화면초기화  }  \*/  while (1) {  printVM();  printInfo();  int c = zeroBufferGetchar();  if (c == 'q' || c == 'Q') {  break;  }  elseif (c == 'J' || c == 'j') {  info.pLeft();  }  elseif (c == 'L' || c == 'l') {  info.pRight();  }  elseif (c == 'i' || c == 'I') {  info.pInsertCoin();  }  elseif (c == 32) { // space  info.pPurchaseMask();  }  clearScreen();  }  return0;  } |

|  |
| --- |
| info.h |
| /\*  \* 프로그램이동작할때필요한모든정보와상수들을정의한곳  \*/  #pragmaonce  #defineLENGTH 64  #defineITEMS 3  #defineITEM\_1 0  #defineITEM\_2 1  #defineITEM\_3 2  #defineINVENTORY 20  #definePRICE\_1 3000  #definePRICE\_2 5000  #definePRICE\_3 20000  #defineuintunsignedint  // 구조체안에함수포인터를통해 OOP의클래스메소드마냥동작하기  typedefstructinfo {  uintinventories[ITEMS];  uintprices[ITEMS];  uintmoney;  uintselected;  charmessage[LENGTH];  uint (\*pGetInventory)(uint);  void (\*pSetInventory)(uint);  uint (\*pGetPrice)(uint);  uint (\*pGetSelected)();  uint (\*pGetMoney)();  void (\*pSetMessage)(constchar\*);  char\* (\*pGetMessage)();  void (\*pLeft)();  void (\*pRight)();  void (\*pInsertCoin)();  void (\*pPurchaseMask)();  } Info;  externInfoinfo;  uintgetInventory(uint);  voidsetInventory(uint);  uintgetPrice(uint);  uintgetSelected();  uintgetMoney();  voidsetMessage(constchar\*);  char\* getMessage();  voidleft();  voidright();  voidinsertCoin();  voidpurchaseMask(); |

|  |
| --- |
| info.cpp |
| #include"info.h"  Infoinfo = {  {INVENTORY, INVENTORY, INVENTORY},  {PRICE\_1, PRICE\_2, PRICE\_3},  0,  ITEM\_1,  {"돈을어서넣어주세요고객님"},  getInventory,  setInventory,  getPrice,  getSelected,  getMoney,  setMessage,  getMessage,  left,  right,  insertCoin,  purchaseMask  };  uintgetInventory(uintn) {  returninfo.inventories[n];  }  voidsetInventory(uintn) {  info.inventories[info.selected] = n;  }  uintgetPrice(uintn) {  returninfo.prices[n];  }  uintgetSelected() {  returninfo.selected;  }  uintgetMoney() {  returninfo.money;  }  voidsetMessage(constchar\* p) {  char\* pMessage = info.message;  // 메세지초기화  while (\*pMessage) {  \*(pMessage++) = '\0';  }  pMessage = info.message;  // 메세지복사  while (\*p) {  \*(pMessage++) = \*(p++);  }  }  char\* getMessage() {  returninfo.message;  }  voidleft() {  if(!info.selected) {  info.selected = ITEM\_3;  return;  }  info.selected -= 1;  }  voidright() {  info.selected += 1;  info.selected %= (ITEM\_3 + 1);  }  voidinsertCoin() {  info.money += 10000;  }  voidpurchaseMask() {  if (info.money>= info.prices[info.selected] &&info.inventories[info.selected] > 0) {  info.money -= info.prices[info.selected];  info.inventories[info.selected]--;  // 구매성공메세지  info.pSetMessage("고객님구매해주셔서감사합니다.");  }  elseif (info.inventories[info.selected] == 0) {  // 안내창에경고메세지  info.pSetMessage("Sold out.");  }  else {  // 안내창에경고메세지  info.pSetMessage("돈이부족합니다고객님, 어서돈을넣어주세요.");  }  } |

|  |
| --- |
| sprites.h |
| #pragmaonce  externcharvendingMachine[]; |

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

|  |
| --- |
| helper.h |
| #pragmaonce  #include<windows.h>  #include<stdio.h>  #include<conio.h>  #include<stdlib.h>  #include"sprites.h"  // 콘솔창의가로세로  #defineWINDOW\_WIDTH 120  #defineWINDOW\_HEIGHT 25  // error codes  enumerrors {  NO\_ERR = 0,  ERR\_SAVE\_TTY\_GETATTR = -2,  ERR\_SAVE\_TTY\_SETATTR = -3,  ERR\_RESTORE\_TTY\_SETATTR = -4  };  // 헬퍼함수들은바깥에서쓸수있도록 extern처리.  externintsetConsoleWindow();  intsave\_tty\_attributes();  intrestore\_tty\_attributes();  externintzeroBufferGetchar();  externvoidsetRed();  externvoidsetCyan();  externvoidsetBlue();  externvoidreset();  externvoidclearScreen(); |

|  |
| --- |
| helper.cpp |
| #include"helper.h"  staticDWORDstdin\_mode;  intsetConsoleWindow()  {  HANDLEhStdout = GetStdHandle(STD\_OUTPUT\_HANDLE);  CONSOLE\_SCREEN\_BUFFER\_INFOcsbiInfo;  SMALL\_RECTsrctWindow;  // Get the current screen buffer size and window position.  if(!GetConsoleScreenBufferInfo(hStdout, &csbiInfo))  {  printf("GetConsoleScreenBufferInfo (%d)\n", GetLastError());  return0;  }  // 변경가능한콘솔창의최대크기정보가져오기  //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());  return0;  }  return1;  }  // 각 OS에따른터미널조작  intsave\_tty\_attributes() {  #ifdef\_WIN32  HANDLEhstdin = GetStdHandle(STD\_INPUT\_HANDLE);  if(!GetConsoleMode(hstdin, &stdin\_mode))  returnERR\_SAVE\_TTY\_GETATTR;  if (hstdin == INVALID\_HANDLE\_VALUE || !(SetConsoleMode(hstdin, 0)))  returnERR\_SAVE\_TTY\_GETATTR; /\* Failed to disable buffering \*/  #elif \_\_APPLE\_\_  structtermiostty\_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  returnNO\_ERR;  }  // 각 OS에따른터미널기본값으로다시설정  intrestore\_tty\_attributes() {  #ifdef\_WIN32  HANDLEhstdin = GetStdHandle(STD\_INPUT\_HANDLE);  if(!SetConsoleMode(hstdin, stdin\_mode))  returnERR\_RESTORE\_TTY\_SETATTR;  #elif \_\_APPLE\_\_  structtermiostty\_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  returnNO\_ERR;  }  // 터미널에서입력버퍼를없애고입력을 screen에표시없이변수에저장해사용  intzeroBufferGetchar() {  int c = 0;  int res = 0;  if ((res = save\_tty\_attributes()) < 0) {  printf("error : %d", ERR\_SAVE\_TTY\_GETATTR);  returnres;  }  c = getchar();  if ((res = restore\_tty\_attributes()) < 0) {  printf("error : %d", ERR\_SAVE\_TTY\_GETATTR);  returnres;  }  returnc;  }  //  // ANSI escape codes  //  voidsetRed() {  printf("\033[1;31m");  }  voidsetCyan() {  printf("\033[1;36m");  }  voidsetBlue() {  printf("\033[1;34m");  }  voidreset() {  printf("\033[0m");  }  voidclearScreen() {  system("cls");  } |

|  |
| --- |
| UI.h |
| #pragmaonce  externvoidprintVM();  externvoidprintInfo(); |

|  |
| --- |
| UI.cpp |
| #include"helper.h"  #include"info.h"  // vendingMachine sprite 출력  voidprintVM() {  for (inti = 0, j = 0; i< WINDOW\_HEIGHT \* WINDOW\_WIDTH; ) {  // 현재선택된아이템에따라색깔을주고  if (i % WINDOW\_WIDTH>= 5 &&i % WINDOW\_WIDTH<= 37  && j >= 7 && j <= 23 &&info.pGetSelected() == ITEM\_1)  setCyan();  elseif (i % WINDOW\_WIDTH>= 39 &&i % WINDOW\_WIDTH<= 73  && j >= 7 && j <= 23 &&info.pGetSelected() == ITEM\_2)  setBlue();  elseif (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();  }  }  // 각종정보출력  voidprintInfo() {  printf("%30s%32s%37s\n", "의료용마스크", "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", "투입된돈", info.pGetMoney(), "", "돈넣기 : 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