```
/*
* Author:   Keith Bush
*           UALR
*
* Date:     December 2, 2014
*
* File:     hmwk0.cpp
*
* Purpose:  The purpose of this homework is to review Programming I topics covering
*           standard input/output, types, declaration, selection constructs, loop
*           constructs, and functions using a very simple Single-User Dungeon environment.
*/

#include<iostream>

using namespace std;

/* GUI Functions */
void clearConsole();
void pauseConsole();
void splashScreen();
void displayGameState(int, int, bool);
void displayGameDone(int, int);
void displayIllegalMove(int, char, bool);
char getAction();

/* Engine Functions*/
bool changeGameState(char, int &, int &, bool &);
bool gameIsNotDone(int, int, bool);

int main(){

    /*State variables*/
    int location = 0;
    char action = '\n';
    int health = 10;
    bool key = false;

    /*Splash Screen*/
    clearConsole();
    splashScreen();

    /*Until Game Termination Condition Do: */
    do{

        /*Display Game State*/
        clearConsole();
        displayGameState(location, health, key);

        /*Collect Player Action*/
        action = getAction();

        /*Update Game State*/
        if(changeGameState(action, location, health, key)){
            health--;
        }else{
            displayIllegalMove(location,action,key);
        }

        /*Check Game Termination*/
    }while(gameIsNotDone(location, health, key));

    /*Display Termination Game State*/
    displayGameDone(location, health);

    return 0;

}
```

```
void clearConsole(){
    system("cls");
}

void pauseConsole(){
    system("PAUSE");
}

void splashScreen(){
    cout << "DUNGEON ADVENTURE" << endl;
    cout << endl;
    cout << "Your name here (2015)" << endl;
    cout << "CPSC 2377, Game Programming, Homework 0" << endl;
    cout << "UALR, Computer Science Dept." << endl;
    cout << endl;
    cout << "INSTRUCTIONS:" << endl;
    cout << endl;
    cout << "Find the key and get out of the dungeon!" << endl;
    cout << endl;
    cout << "        (North)          " << endl;
    cout << "           w             " << endl;
    cout << "           |             " << endl;
    cout << "(West) a --+-- d (East)  " << endl;
    cout << "           |             " << endl;
    cout << "           s             " << endl;
    cout << "        (South)          " << endl;
    cout << endl;
    pauseConsole();
}

void displayGameState(int location,int health, bool key){

    cout << "View:   ";
    switch(location){
    case 0:
        cout << "A large, torchlit room with doors South and East." << endl;
        break;
    case 1:
        cout << "A jailer's barracks room with doors West, South, and East." << endl;
        cout << "        There is daylight entering under the door to the East." << endl;
        break;
    case 3:
        cout << "A small, dark prison cell with doors North and East." << endl;
        break;
    case 4:
        cout << "A store room with doors West and North." << endl;
        if(!key){
            cout << "        There is a key on the floor. You pick it up." << endl;
        }
        break;
    default:
        break;
    }

    cout << "Health: " << health << endl;
    cout << "Equip:  ";
    if(key){
        cout << "1 jailer's key" << endl;
    }else{
        cout << endl;
    }

}

void displayGameDone(int location, int health){

    clearConsole();
    if(health > 0){
        if(location==2){
            cout << "YOU FOUND THE KEY AND ESCAPED!" << endl;
```

```
        }
    }else{
        cout << "YOU DIED...RIP." << endl;
    }

    cout << endl;
    pauseConsole();
}


void displayIllegalMove(int location, char action, bool key){

    clearConsole();

    if(location==1 && !key && action=='d'){
        cout << "The door is locked." << endl;
    }else{
        cout << "You cannot go that way." << endl;
    }
    cout << endl;

    pauseConsole();

}

char getAction(){
    char action;
    cout << endl;
    cout << "Select action: ";
    cin >> action;
    return(action);
}

//Apply the game's logical rules to the current location, possession of the
//key, movel legality, and health.
bool changeGameState(char action, int & location, int & health, bool & key){

    //Assume the move is illegal
    bool legalMove = false;

    //Need to change key here to make gui work
    if(location == 4){
        if(!key){
            key = true;
        }
    }

    //Encode the transition function from the specification
    //Switch on the action.  For each room that the action is legal,
    //update the location and identify the move's legality (o/w,
    //the move is illegal.
    switch(action){
    case 'a': //Moving West
        if(location == 4){
            location = 3;
            legalMove = true;
        }else{
            if(location == 1){
                location = 0;
                legalMove = true;
            }
        }
        break;
    case 'd': //Moving East
        if(location == 0){
            location = 1;
            legalMove = true;
        }else{
            if(location == 3){
                location = 4;
```

```
                legalMove = true;
            }else{
                if(location == 1 && key){
                    location = 2;
                    legalMove = true;
                }
            }
        }
        break;
    case 'w':  //Moving North
        if(location == 3){
            location = 0;
            legalMove = true;
        }else{
            if(location == 4){
                location = 1;
                legalMove = true;
            }
        }
        break;
    case 's': //Moving South
        if(location == 0){
            location = 3;
            legalMove = true;
        }else{
            if(location == 1){
                location = 4;
                legalMove = true;
            }
        }
        break;
    default:  //Handle garbage keystrokes
        break;
    }

    return(legalMove);

}

//Check the end-of-game conditions.  If the player is out
//of health or the player has reached room 2 (i.e., outside)
//then the game is over.
bool gameIsNotDone(int location, int health, bool key){

    bool gameNotDone = true;

    if(health<=0 || location == 2){
        gameNotDone = false;
    }

    return(gameNotDone);
}
```