```
#include <iostream>
#include <string>
#include <ctime>

using namespace std;

/*
Example of using arrays and for loops to create a mini dungeon crawler

*/


const int LEVEL_WIDTH = 60;
const int LEVEL_HEIGHT = 15;
const int NUM_ITEMS = 5;

const int ITEM_FOOD = 1;
const int ITEM_GOLD = 2;
const int ITEM_TRAP = 3;

const int START_HEALTH = 100;
const int START_FOOD = 20;
const int START_GOLD = 0;

int LevelData[LEVEL_WIDTH][LEVEL_HEIGHT];

int charPos[] = {0,0};
int charStatus[3];

int itemsLocations[NUM_ITEMS][3];

bool IsPlaying;

void PrintPlayerStatus()
{
	cout << "\n1: move left, 2: move up, 3: move down, 4: move right, 5: quit\n";
	cout << "Current status: ";
	cout << "Health: " << charStatus[0];
	cout << " Food: " << charStatus[1];
	cout << " Gold: " << charStatus[2];
	cout << "\n";
}

void PrintLevel()
{
	for (int i = 0; i < LEVEL_HEIGHT; i++)
	{
		for (int j = 0; j < LEVEL_WIDTH; j++)
		{
				if (j == charPos[0] && i == charPos[1])
				{
					cout << "@";
				}
				else
				{

					switch (LevelData[j][i])
					{
					case 1:
						cout << "*";
						break;
					case 3:
						cout << "%";
						break;
					default:
						cout << ".";
					}
				}

			}
			cout << endl;
		}

	}

	void InitLevel()
	{
		srand(time(NULL));

		charPos[0] = 1; // LEVEL_WIDTH / 2;
		charPos[1] = 4; // LEVEL_HEIGHT / 2;

		charStatus[0] = START_HEALTH;
		charStatus[1] = START_FOOD;
		charStatus[2] = START_GOLD;

		IsPlaying = true;

		for (int i = 0; i < LEVEL_HEIGHT; i++)
		{
			for (int j = 0; j < LEVEL_WIDTH; j++)
			{
				LevelData[j][i] = 0;
				if (j == 0 || j == (LEVEL_WIDTH - 1) || i == 0 || i == (LEVEL_HEIGHT - 1))
				{
					LevelData[j][i] = 1;
				}
			}
		cout << endl;
	}

	for (int i=0; i < NUM_ITEMS; i++)
	{
		int itemType = (rand() % 3) + 1;

		int itemPosX = (rand() % (LEVEL_WIDTH - 2)) + 1;
		int itemPosY = (rand() % (LEVEL_HEIGHT - 2)) + 1;

		// int itemPosX = 3 + i;
		// int itemPosY = 3 + i;

		itemsLocations[i][0] = itemType;
		itemsLocations[i][1] = itemPosX;
		itemsLocations[i][2] = itemPosY;

		LevelData[itemPosX][itemPosY] = 3;
	}

}

void CheckForItemPickup()
{
	for (int i=0; i < NUM_ITEMS; i++)
	{
		// cout << i << ":" << itemsLocations[i][1] << "," << itemsLocations[i][2] << endl;

		if (itemsLocations[i][1] == charPos[0] &&
			itemsLocations[i][2] == charPos[1])
		{
			// remove the item from the board
			LevelData[charPos[0]][charPos[1]] = 0;

			switch (itemsLocations[i][0])
			{
				case ITEM_FOOD:
					cout << "You found some food!\n";
					charStatus[1] += rand() % 10 + 2;
					break;
				case ITEM_GOLD:
					cout << "You found some gold!\n";
					charStatus[2] += rand() % 5 + 1;
					break;
				case ITEM_TRAP:
					cout << "It's a trap!\n";
					charStatus[0] -= rand() % 20 + 5;
					break;
			}
		}
	}
}


void ProcessInput(int i)
{
	switch (i)
	{
		case 1:
			charPos[0] -= 1;
			if (charPos[0] < 1) charPos[0] = 1;
			break;
		case 4:
			charPos[0] += 1;
			if (charPos[0] > LEVEL_WIDTH-2) charPos[0] = LEVEL_WIDTH - 2;
			break;
		case 3:
			charPos[1] += 1;
			if (charPos[1] > LEVEL_HEIGHT-2) charPos[1] = LEVEL_HEIGHT-2;
			break;
		case 2:
			charPos[1] -= 1;
			if (charPos[1] < 1) charPos[1] = 1;
			break;
		case 5:
			IsPlaying = false;
			break;

	}

	CheckForItemPickup();
}

int main()
{
	int input;
	InitLevel();

	while (IsPlaying)
	{
		PrintPlayerStatus();
		PrintLevel();
		cin >> input;
		ProcessInput(input);
	}

	// cin >> input;
	return 0;
}
```