****

Report:

**Batman Vs Joker Game**

**Prepared by**

**Padhuri Samanvitha Reddy**

**23-03-2022**

**LTTS**

# BATMAN VS JOKER GAME

## INTRODUCTION :

## DESCRIPTION:

* This game is created by taking reference from the game X and O which is also called as "Tic Tac Toe".
* There is a city named Gotham City, Where Joker Plants bombs of type X or O and the Batman should Stop it from making a sequence of 4 bombs in a row to prevent Gotham City.

## OBJECTIVE:

* The goal of the game is for players to position their marks so that they make a continuous line of four cells vertically, horizontally, or diagonally.
* An opponent can prevent a win by blocking the completion of the opponent's line.

## SPACE AND TIME COMPLEXITY:

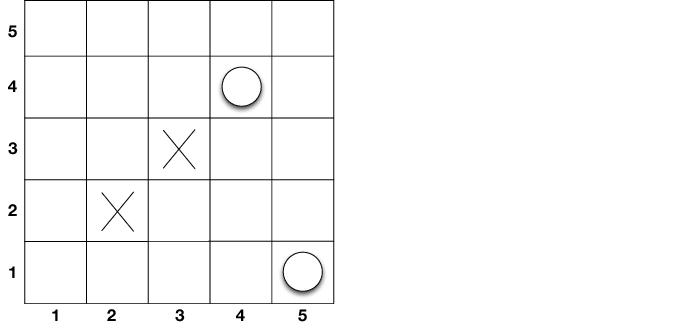
### TIME COMPLEXITY :

* O(1) time complexity that requires O(n) space complexity. it is simpler to write, and debug and more importantly, there are no loops!

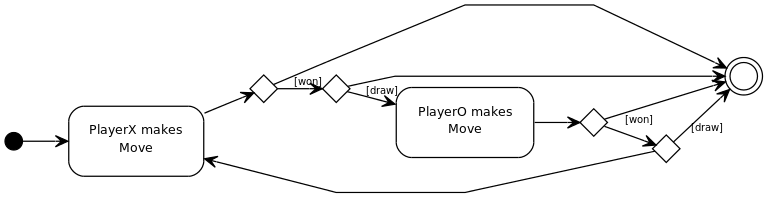
### SPACE COMPLEXITY :

* Combinatorial game theory has several ways of measuring game complexity.
* The state-space complexity of a game is the number of legal game positions reachable from the initial position of the game.
* The game tree size is the total number of possible games that can be played: the number of leaf nodes in the game tree rooted at the game's initial position.
* Decision complexity of a game is the number of leaf nodes in the smallest decision tree that establishes the value of the initial position.
* The game-tree complexity of a game is the number of leaf nodes in the smallest full-width decision tree that establishes the value of the initial position.
* O(n) in space complexity.

## GAME PREVIEW EXAMPLE:

[](https://user-images.githubusercontent.com/62429376/158420699-7ac54a51-717a-417f-8d99-61b849fa0fb6.png)

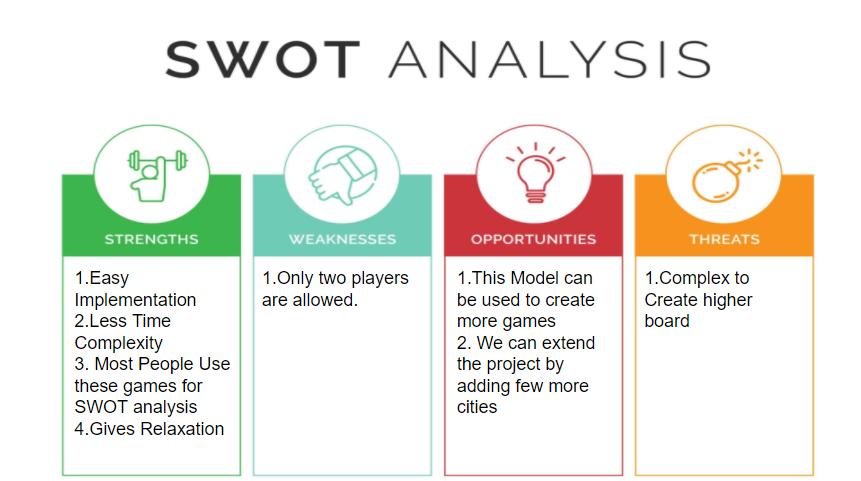
## GAME MODEL DESIGN :

[](https://user-images.githubusercontent.com/62429376/158425634-37f97448-6d95-4695-bac2-518601878d69.png)

## EXPLANATION :

* (X) and (O) are 2 kinds of bombs available in gotham city
* both BATMAN and JOKER has to put both (X) and (O)
* JOKER wins if sequence of 4 bombs is made vertically,horizontally or diagonally with either of the bomb types
* BATMAN wins if JOKER's plan is destroyed and prevents him from making the bomb connections of the same type
* Below is the map of GOTHAM city and the bombs placed

## SWOT ANALYSIS

[](https://user-images.githubusercontent.com/62429376/158436359-0ca1f453-4a7d-4de3-838d-30cbbc616d79.png)

## 4W's and 1'H

## WHO :

* The kids and adults play these kind of games which are made intresting. These games are used in schools for different analysis applications for children.

## WHAT :

* These games are used for different analysis applications for children.Children play these games very often to obtain relaxation.They not only help children in terms of cognitive growth but also personal growth and even meaningful life lessons

## WHEN :

* These games are played very often every day by many people and are used for various analysis methods.

## WHERE :

* This game can be played on a book or a paper or on the computer/system.

## HOW :

* They can install the game on the mobile or laptop.

# High Level Requirements:

| **ID** | **Description** | **Status** |
| --- | --- | --- |
| HR01 | Player should be able to select the cities | Implemented |
| HR02 | Player should be able to select the bombs | Implemented |
| HR03 | Player should be able to match the bombs | Implemented |
| HR04 | Player should have a choice to select bombs | Implemented |
| HR05 | Player should be able to end the game whenever wanted | Future |
| HR06 | Possibility of playing with more than 2 players | future |
| HR07 | Data should not be lost in case of failure | Implemented |

## Low Level Requirements:

| **ID** | **Description** | **Status(Implemented/Future)** |
| --- | --- | --- |
| LR01 | Timed game | Future |
| LR02 | The moves to be undone | Future |
| LR03 | Can a third person spectate the game | Future |
| LR04 | Will any statistics be stored for user | Future |
| LR05 | Will there be any rating change while playing a game | Future |
| LR06 | Can any tournaments be conducted in the game | Future |
| LR07 | Can any other technologies be used in the game | Future |

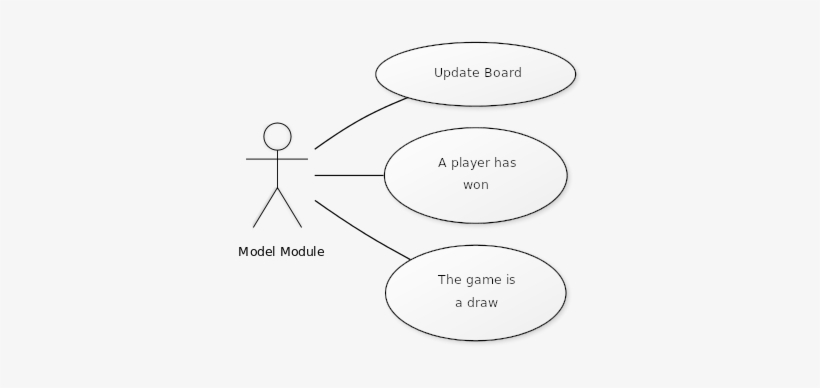
## Best Methods To Be Followed

* Used functions to decrease dependency on main function
* Used structures and arrays to accept the inputs from user and store the values which helped in creating easy design of Employee management system.
* Printf statements have been placed only wherever necessary to avoid confusions
* Created header file so that the fuctions can be used else where ever required without any difficulty
* Unit testing is done to avoid any computational errors.

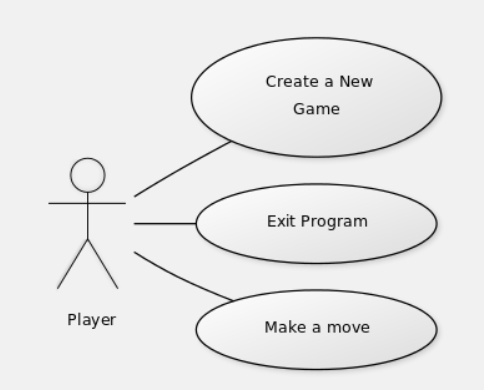
# ARCHITECTURE

## Usecase Diagrams

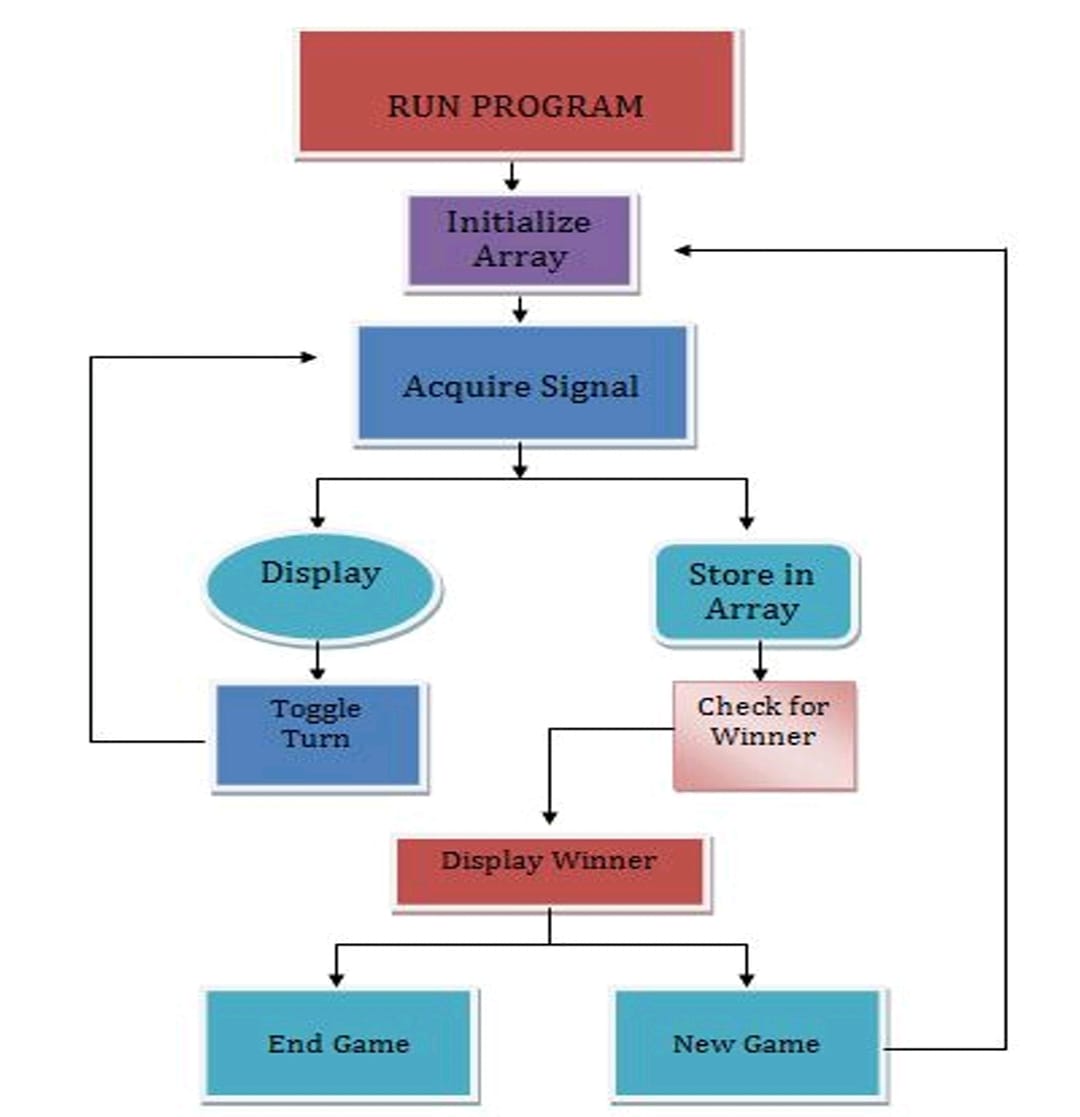
## High Level Design

[](https://user-images.githubusercontent.com/62429376/158599848-ab358f8c-b49c-4ee9-bf27-346b1519f6aa.jpg)

## Low Level Design

[](https://user-images.githubusercontent.com/62429376/158602317-e49c46b7-81ac-476f-91c9-b39287e92946.jpg)

## FlowChart of Batman Vs Joker

[](https://user-images.githubusercontent.com/62429376/158601824-718d7dd4-7b21-421b-8941-0a85274424e6.jpg)

## Here are the below steps to run the code

* For Running and Building the code
  + Type make run in terminal
* For Running the Building tests
  + Type make run\_test in terminal
* For static analysis
  + Type make static\_analysis in terminal
* For dynamic analysis
  + Type make dynamic\_analysis in terminal
* For code coverage
  + Type make coverage in terminal
* For cleaning
  + Type make clean in terminal

## Usage of Pointers

## Pointers:

* The Pointer in C, is a variable that stores address of another variable.
* A pointer can also be used to refer to another pointer function.
* A pointer can be incremented/decremented, i.e., to point to the next/ previous memory location. The purpose of pointer is to save memory space and achieve faster execution time. Like variables, pointers in C programming have to be declared before they can be used in your program. Pointers can be named anything you want as long as they obey C’s naming rules. A pointer declaration has the following form.

syntax: data\_type \* pointer\_variable\_name;

* Types of Pointers in C: 1. Null Pointer 2.Void Pointer 3.Wild pointer 4.Dangling pointer

## Usage of Preprocessor:

* The C preprocessor is a macro processor that is used automatically by the C compiler to transform your program before actual compilation. It is called a macro processor because it allows you to define macros, which are brief abbreviations for longer constructs.

\*The C preprocessor provides four separate facilities that you can use as you see fit:

* Inclusion of header files. These are files of declarations that can be substituted into your program.

## Function Pointer

* In the C function pointer is used to resolve the run time-binding.
* A function pointer is a pointer that stores the address of the function and invokes the function whenever required.
* In C, we can use function pointers to avoid code redundancy.

## Struct

* A structure is a key word that create user defined data type in C. A structure creates a data type that can be used to group items of possibly different types into a single type.
* ‘struct’ keyword is used to create a structure.
* A structure variable can either be declared with structure declaration or as a separate declaration like basic types.
* Structure members cannot be initialized with declaration.

## Typedef

* typedef, which we can use to give a type a new name. Following is an example to define a term BYTE for one-byte numbers.After this type definition, the identifier BYTE can be used as an abbreviation for the type unsigned char
* Syntax: typedef data\_type new\_name

# TEST PLAN

## High Level Test Plan :

| **ID** | **Description** | **Expected Input** | **Expected Output** | **Actual Output** | **Type of Test** |
| --- | --- | --- | --- | --- | --- |
| HR01 | Player should be able to select the cities | User's choice a character | SUCCESS | SUCCESS | Scenario Based |
| HR02 | Player should be able to select the bombs | User's choice a character | SUCCESS | SUCCESS | Scenario Based |
| HR03 | Player should be able to match the bombs | User's choice a character | SUCCESS | SUCCESS | Technical Based |

## Low Level Test Plan :

| **ID** | **Description** | **Expected Input** | **Expected Output** | **Actual Output** | **Type of Test** |
| --- | --- | --- | --- | --- | --- |
| LR01 | Player should have a choice to select bombs | User's choice a character | SUCCESS | SUCCESS | Technical Based |
| LR02 | Data should not be lost in case of faliure | Users Choice | SUCCESS | SUCCESS | Scenario Based |

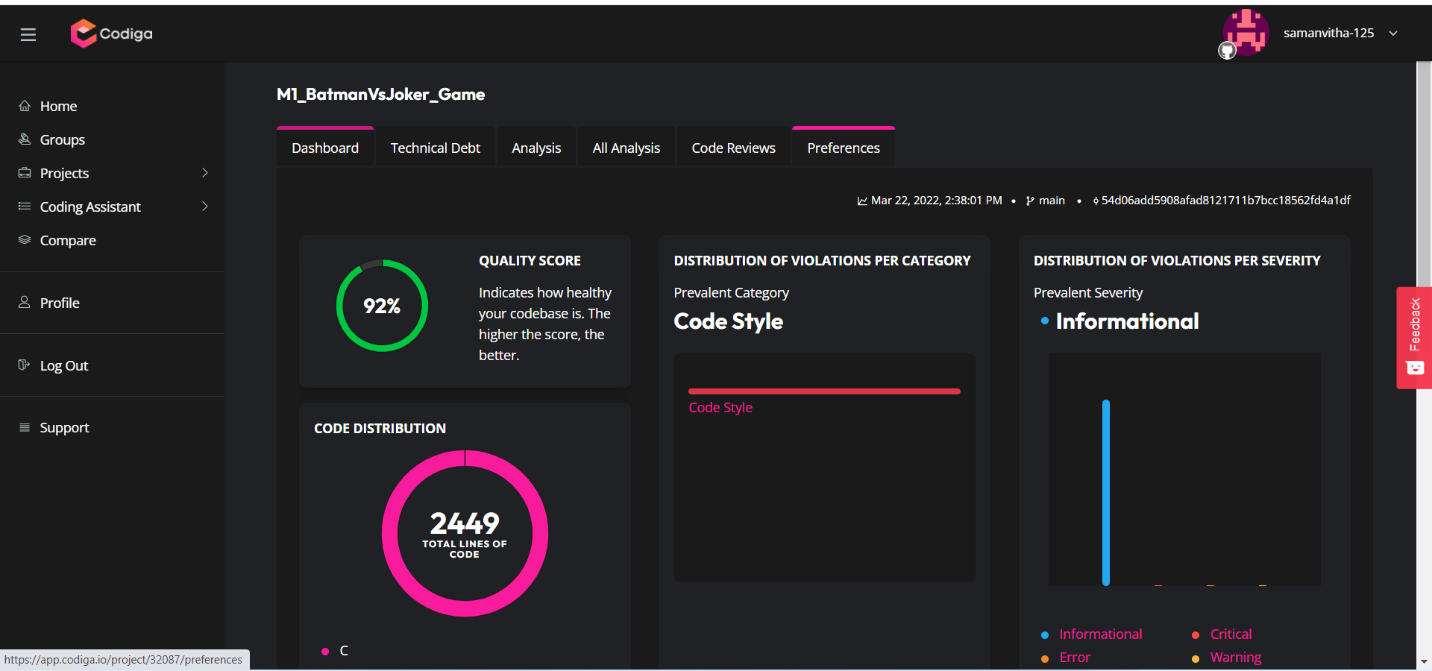
## Usage of Unit Testing-Frameworks

* Unity Test Framework (UTF) enables Unity users to test their code in both Edit Mode and Play Mode, and also on target platforms such as Standalone, Android, iOS, etc.
* UTF uses a Unity integration of NUnit library, which is an open-source unit testing library for .Net languages.
* It is possible to extend the Unity Test Framework (UTF) in many ways, for custom workflows for your projects and for other packages to build on top of UTF.

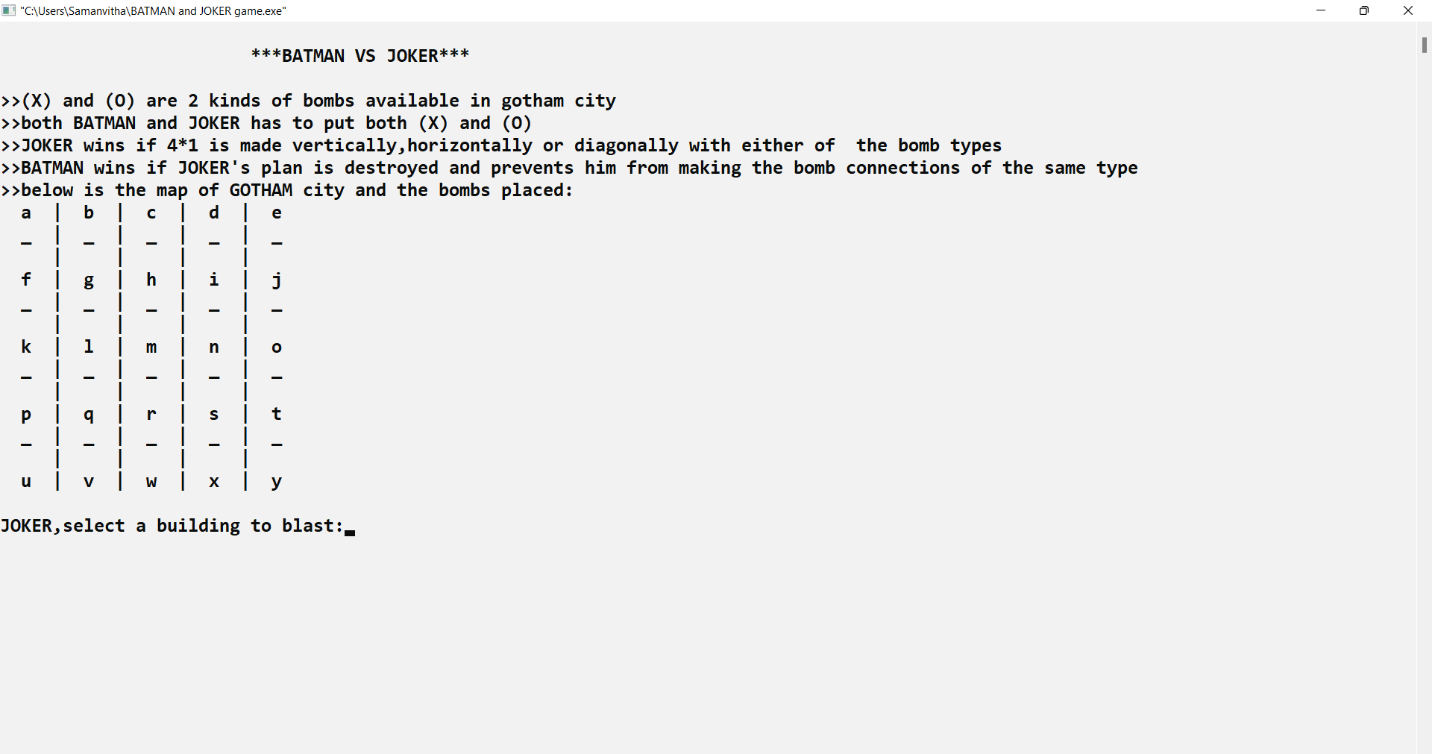
## Best Practices

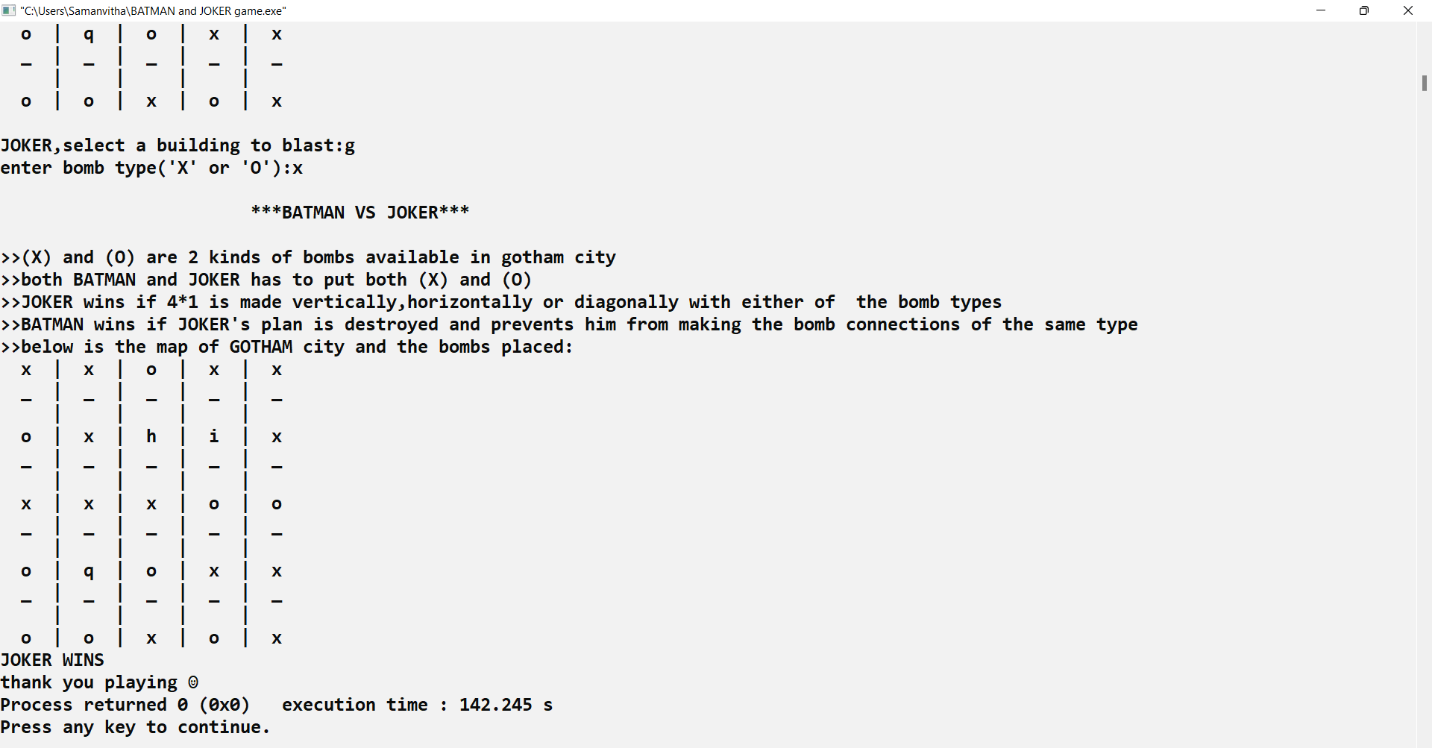
*  Checked all the possibilities of output
*  Mentioned all the inputs for better understanding
*  Presented in tabular form for easy understanding
*  Both High level and low level Test plans are shown

# Code Inspector

[](https://user-images.githubusercontent.com/62429376/159704147-a632136f-3a4a-4bb8-bfa8-45b967b1f720.png)

# Output Images

[](https://user-images.githubusercontent.com/62429376/159704715-63909fcd-f589-4f9c-b8f3-73d4ae74c980.png)

[](https://user-images.githubusercontent.com/62429376/159704734-1c055e53-a370-43da-83cf-601a93354163.png)