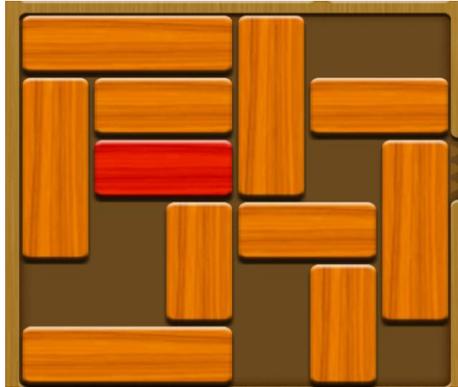


# PROJECT REPORT

## *PRISON BREAK*



### **Intoduction**

The main goal of PRISON BREAK is to move the PRISONER out of the Prison through the hole shown by the arrow

One has to move the BLOCKERS to achieve the goal.

The vertical BLOCKERS can move vertically only & same applies for the horizontal BLOCKERS.

<b>Sr.no</b>	<b>Important terms</b>	<b>Defination</b>
1.	Prisoner	Red Block
2.	Blocker	Yellow coloured blocks
3.	Prison	Square shaped boundary

## **Problem1**

- We were in a dilemma whether to have PRE-DEFINED LEVELS or AUTO-GENERATED RANDOM LEVELS.
- After a brief discussion with our TA, we decided upon the latter.
- We also had a problem, as the code was becoming too large.
- We divided the team into two for:
  - 1.USER INTERFACE : Shreerang Kaore , Nitin Chaudhary.
  - 2.AUTO-GENERATION OF LEVELS: Pratik Babhulkar , Amit Patil.

## **Solution(Random generation)**

- The code for the auto-generation was difficult. So we all discussed the same & came with a solution:
  - We first randomly arranged the BLOCKERS & the PRISONER.
  - Then Pratik developed an auto-solver to solve a randomly arranged pattern. He was assisted by us (mainly by Amit).
  - Using files, minimum number of moves required to solve a randomly generated level was calculated.
  - Difficulty level thus generated using minimum number of moves needed to solve the game was stored in files.
- We wrote a lot of functions. This saved our workload by a lot of extent.

## **Problem2(Graphics part)**

- Opening a second canvas while the first one is open was a major problem during the initial coding.
- This problem was occurring as the function "closeCanvas( );" did not allow new rectangles to be drawn. Even the function "cleardevice( );" did not solve our problem.
- Initially, we had a problem in using the Click function.
- We did not want to use the " Repeat(int i){ }" function. We wanted the code to look more perfect.

## **Solution**

- We used hide and show functions. As it was a gruesome task, we wrote a function for it and used it every time.
- We applied do{ }while( ); statement for solving our click dilemma.

## **About the Game**

- We decided to have 36 levels in this game. However, we can even include upto 1000 levels! However, due to time constraint and the amount of time required to run the code, we decide upon 36 levels.
- We also decided to include 4 themes that would change the colour of the BLOCKERS, background, Home (not that of the PRISONER), etc.
- We decide to use mouse for selecting and keyboard for moving the blockers and the prisoner.

- The number of moves made by a player are shown on the game window.
- The game also include undo, restart options.
- One can choose any level at any time.

## **Modifications**

- We can change the orientation,length of prisoner at any time.
- We can also change the handling of the game(by using drag function).
- We can increase the size of Canvas.
- We can also include two prisoners at a time!
- We can increase the speed for generation of random levels.

## **Constraints**

- We also wanted to include a timer. However, due to time constraint, we were not able to do so.
- We also wanted to show the"least number of moves until now". However, due to shortage of time, we decided not to venture upon that and concentrate on other important aspects.

### **Self Evaluation marks stage1:**

1. Pratik Babhulkar-140100013-**5** Marks.
2. Shreerang kaore-140100028-**4.5** marks
3. Amit Patil -140100043-**3** marks
4. Nitin Choudhary - 140100048-**3.5** marks

### **Self Evaluation marks stage2:**

1. Pratik Babhulkar-140100013-**5** Marks.
2. Shreerang kaore-140100028-**4** marks
3. Amit Patil -140100043-**3.5** marks
4. Nitin Choudhary - 140100048-**3** marks