

# Application of Hill Climbing algorithm to generate solutions for Make N' Break game

Aida Khosroshahi

## Introduction

The NREC ARM-S team has been working on a software to allow a robot play the *Make N' Break* game at the Smithsonian Air & Space museum in Washington, DC.

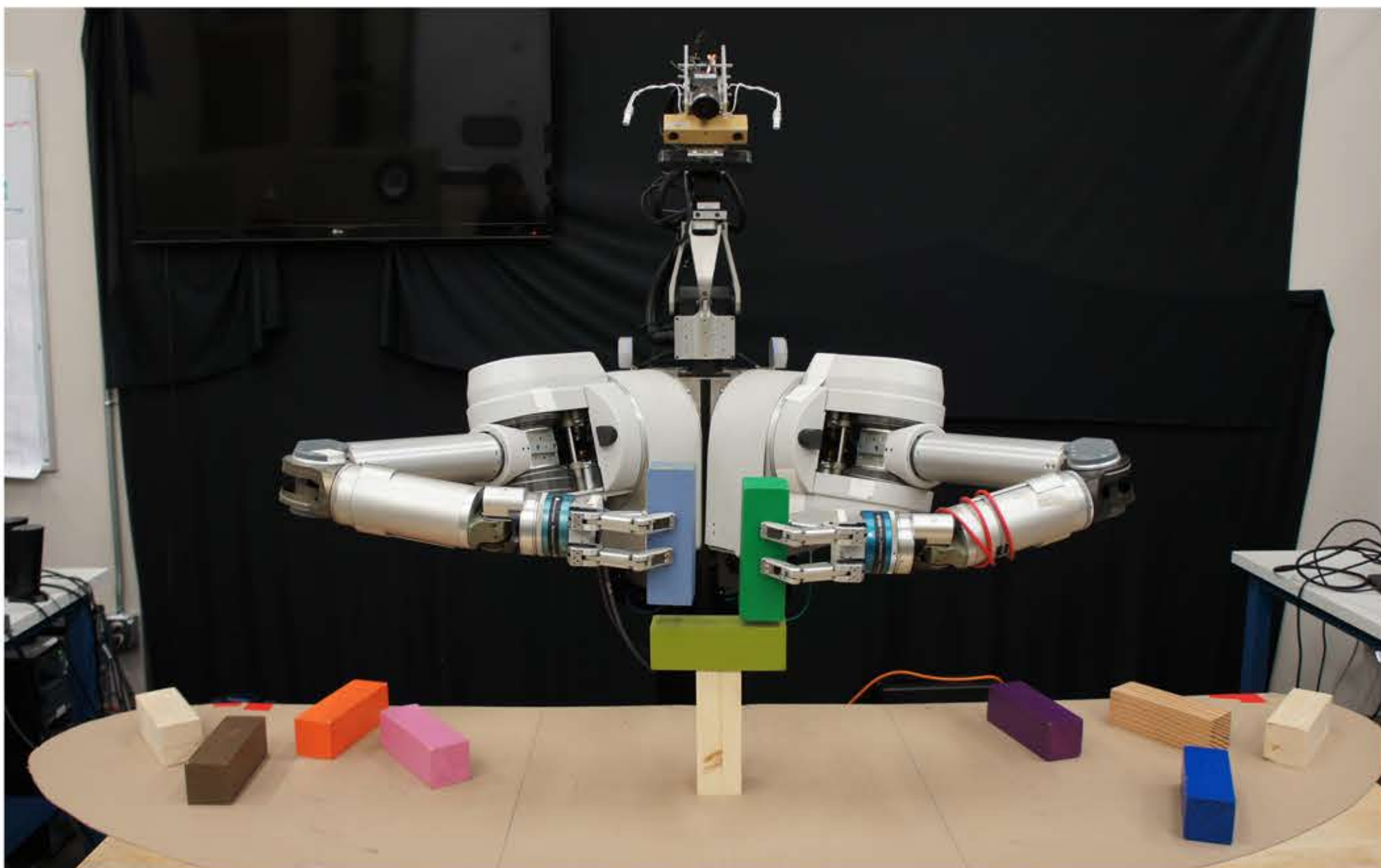


Figure 1. The robot playing Make N' Break. Note the robot needs to use both hands simultaneously, because putting either of green or blue block before the other one, will make the tower to fall over.

Make N' Break is a game played with rectangular blocks that get stacked into patterns. Players are

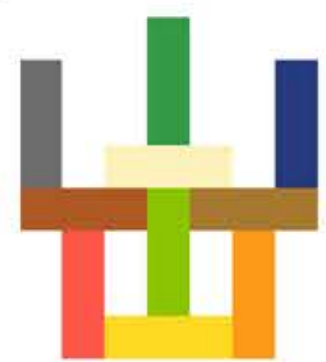


Figure 2. A sample pattern for Make N' Break.

## Objective

- Design an algorithm to generate a sequence of robotic manipulation moves for building a pattern.
- Create a graphical user interface to let the museum visitors design patterns that are later built by the robot. The interface also displays robot telemetry during the building phase.

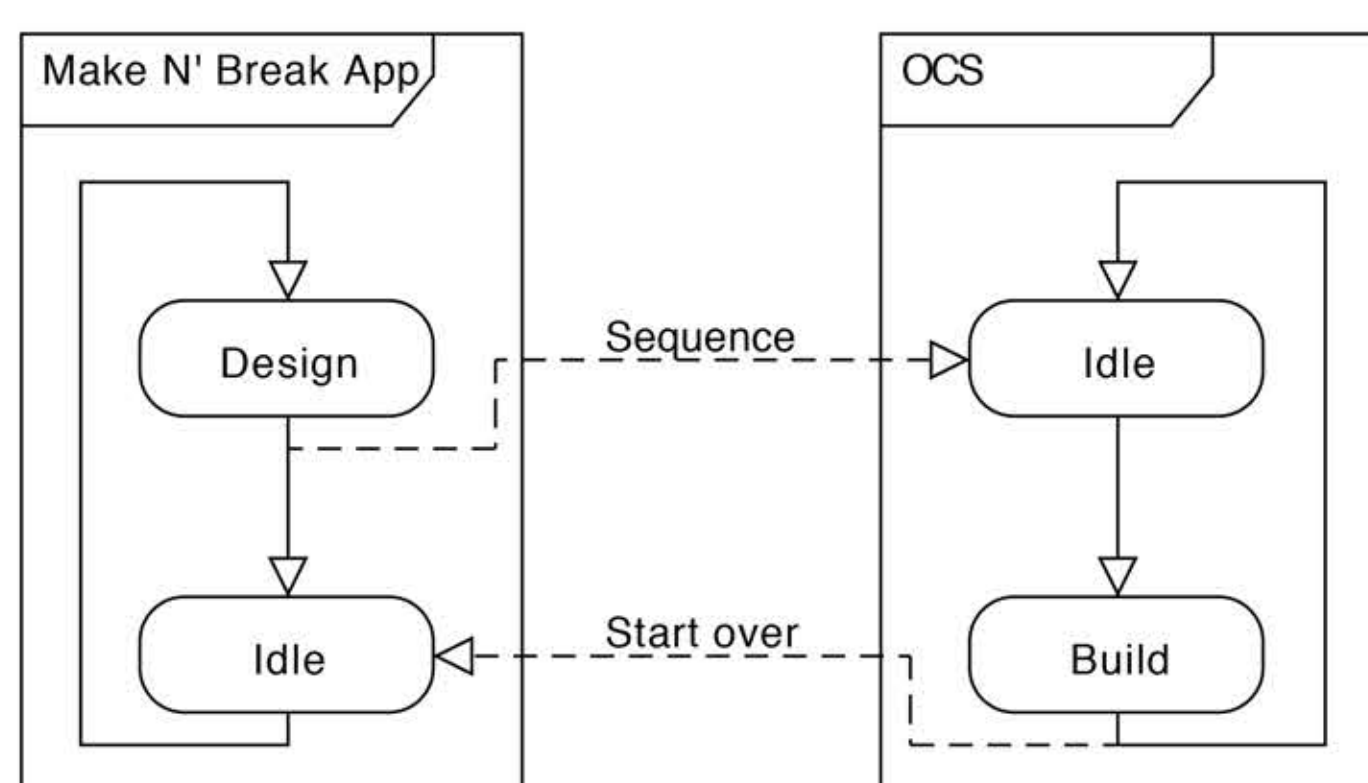


Figure 3. Make N' Break App and robot state diagram.

## Outcomes

The Hill Climbing [1] algorithm was used to generate the sequence of moves to build the pattern. Hill Climbing is a local-search algorithm trying to minimize or maximize a target function. It implements a loop that replaces the current state with the best neighbor (based on the target function) for each iteration, until it cannot improve the solution anymore.

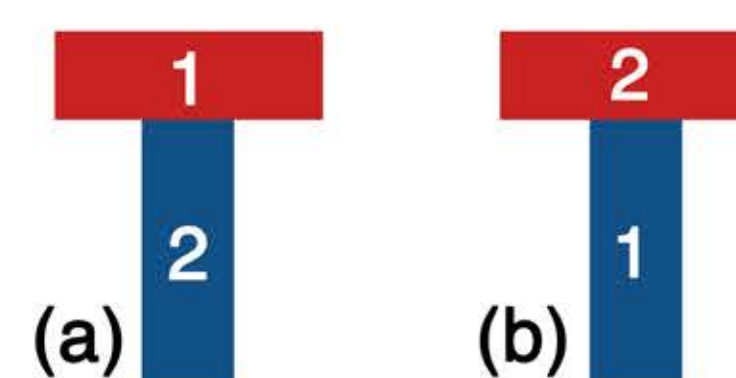


Figure 4. The number on each block shows its position in the sequence. The algorithm will proceed from (a) to (b) to minimize the target function. Target function value for (a) is high because it is trying to put the red block on top of nothing.

The *state* is defined as a complete sequence of moves that will put all the blocks in place, potentially in wrong order/hand configuration.

The *Target function* is computed based on the following criteria:

- Number of steps which will lead to an unstable structure.
- Space left between blocks for the robot's hand.
- Special cases to handle two-handed moves.

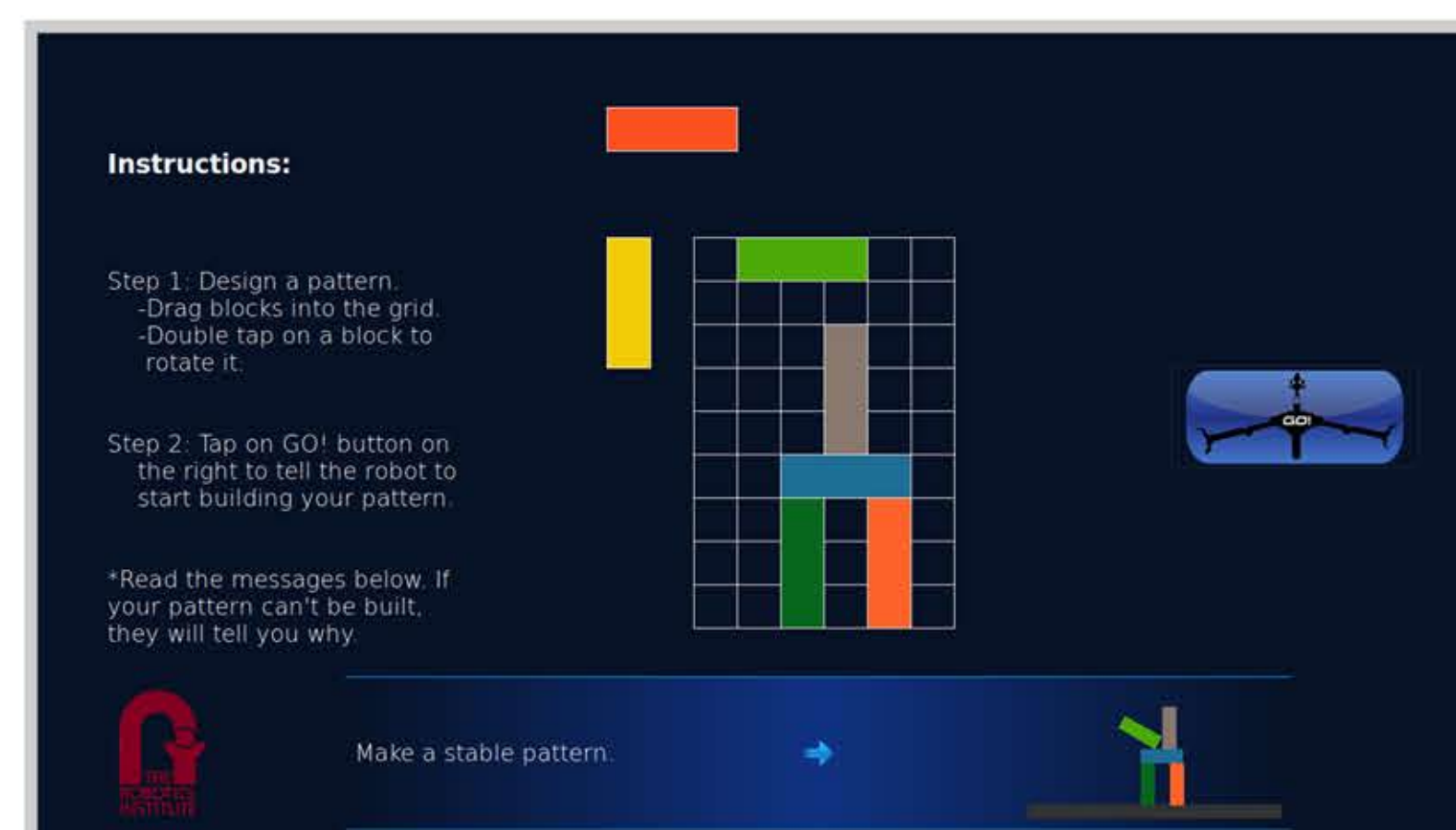


Figure 5. Snapshot of Make N' Break App during design stage. Users build a pattern by dragging blocks into the grid. A simulation of their pattern on the bottom shows if the pattern is stable or not.

*Neighbors* were defined in a way to minimize local maxima while keeping the neighbors' size small enough for efficiency issues. Neighbors are computed by swapping two steps, changing the operative hand, merging two one-handed moves, and separating a two-handed move.

The stability check of structures uses Box2D Physics Engine [2], and graphical user interface was implemented using Qt [3].

The sequence sent to the robot includes the final position, orientation, and color of block(s), as well as which hand(s) to use.

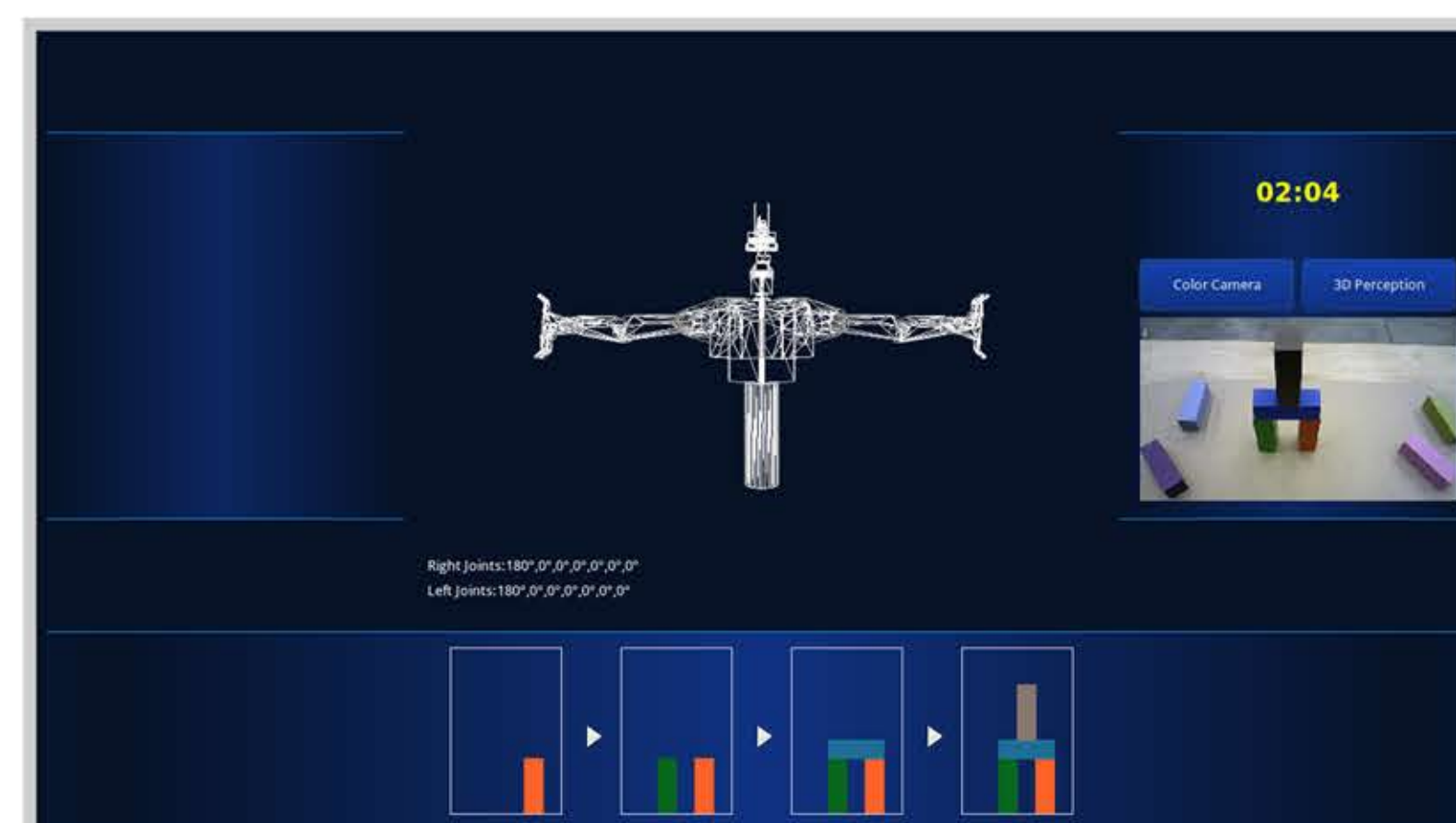


Figure 6. Snapshot of Make N' Break App while the robot is building the pattern. A live model of the robot is in the center, and the sequence of moves is shown in steps on the bottom.

## Acknowledgement

Special thanks to Prof. Drew Bagnell for supporting this work. Thanks to Moslem Kazemi and Jean-Sebastien Valois for their help and support, and to Matt Klingensmith and Arun Venkatraman for valuable suggestions.

## References

- 1 Hill Climbing algorithm  
Russell, Stuart J., and Peter Norvig. "Hill-climbing Search." *Artificial Intelligence: A Modern Approach*. 2003.
- 2 Box2D: A 2D Physics Engine for Games  
<http://box2d.org/>
- 3 Qt Project:  
<http://qt-project.org/>

