BLOCK BLAST GAME

DATA STRUCTURES

Mokshita varma Varada raju   
*Montgomery,Alabama,USA*  
Auburn university at Montgomery   
Matthew Middleton   
*Montgomery,Alabama,USA*  
Auburn university at Montgomery

*Abstract*—

Keywords—

# Introduction

Creating games is an effective way to learn and apply programming concepts. This project is called “Block Blast Game”. Block Blast game is a puzzle game which is creative, attention grabber, and could also be called a brain game as we need creative and critical thinking to solve the game. For this project we are designing this game using C++ programming Language with the concept of Data Structures.

Block Blast game is a grid-based puzzle game where blocks of different colors are placed on the screen. When the game starts, the player selects a group of matching blocks, they are removed, and blocks above fall down to fill empty space. New blocks may also add as the game goes on. For this game to work smoothly, we use Data structures to manage the grid, store player moves and handling the removing and filling of blocks.

The goal of this project is not just building the game but also show how data structures can be used and in addition to that add more features to the game such as timer and reset.

# Literature review

Before diving into the Block Blast game project, it’s important to explore the methods and ideas behind similar puzzle games. Many of these games rely on basic data structures like arrays, lists, or trees to manage the game and control how blocks are removed or fall into place. These techniques are well established, but they often stick to the basics. By exploring how these games are built, we can spot patterns, strengths, and opportunities to introduce something new.

In our version of the game, we want to push beyond the standard approach. We are using data structures not just to keep the grid organized, but also to support extra features like a timer, a reset button, and possibly even more advanced gameplay mechanics. These additions make the game more interactive and enjoyable, and they give us a chance to apply programming concepts in a creative way.

Looking at how similar games have been made helps us understand what direction to take. It gives us a clear picture of the usual techniques and shows us where we can bring something new to the table. This project isn’t just about making a fun puzzle game; it’s about using data structures in smart ways to make the game run better and offer more flexibility for new features.

Here are some perspectives on puzzle games. Puzzle games like Block Blast have stayed popular because they are simple to understand but still challenge players to think ahead. Games that use grids and shapes, such as Tetris (1984) and 1010!, make players plan how to fit different block patterns into limited space (Juul, 2005). This kind of gameplay encourages problem-solving and spatial reasoning, which keeps people engaged without needing complex rules.

Researchers have found that games like these are effective at promoting focus and logical thinking. Csikszentmihalyi (1990) described this type of engagement as “flow,” where players lose track of time because the task is challenging but still achievable. Puzzle games work especially well for this since players can see their progress instantly as rows or columns clear.

From a design perspective, block puzzle games rely on clear visual structure and immediate feedback. Every move changes the grid and affects what options come next, creating a loop of decision-making and reward. This structure makes them both relaxing and mentally stimulating. Because the goal is always clear, filling rows or columns, the challenge comes from planning how each piece fits into the grid.

Studies on educational gaming also suggest that puzzle-style games help people practice critical thinking and strategy in a fun, low-pressure way (Papastergiou, 2009). These types of games are accessible to all ages and are easy to adapt to different platforms, which is why they remain one of the most common styles of mobile and desktop games.

# methodology

# Limitations

# conclusion

##### References

[1] J. Blanchette and M. Summerfield, C++ GUI Programming with Qt 4, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2008

[2] M. Csikszentmihalyi, Flow: The Psychology of Optimal Experience. New York, NY, USA: Harper & Row, 1990.

[3] J. Juul, Half-Real: Video Games Between Real Rules and Fictional Worlds. MIT Press, 2005.

[4] M. Papastergiou, “Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation,” Computers & Education, vol. 52, no. 1, pp. 1–12, Jan. 2009, <https://doi.org/10.1016/j.compedu.2008.06.004>