

# Summary (Testing)

## Table of Contents

- [Summary](#)
- [Terms, Notations, Acronyms](#)

## Summary

The testing documentation for "A Blocky Start" encompasses a thorough and multifaceted approach to quality assurance, integrating unit testing, integration testing, validation testing, and system testing to ensure the highest standards of software quality and reliability. Following the detailed unit testing of individual components to verify their standalone functionality, the testing strategy progresses to integration testing. This phase is crucial for assessing the interactions and data flow between integrated components, ensuring seamless operation and communication within the game's architecture. It addresses potential integration issues such as data exchange errors and dependency conflicts, which are critical for maintaining a smooth and cohesive gameplay experience.

Validation testing serves as the next critical step, focusing on ensuring that the game meets all its functional and non-functional requirements as defined in the development phase. This testing phase is designed to ensure the game fulfills its intended purposes, adhering to the specifications and satisfying the target audience's needs and expectations. It's a pivotal process that confirms the game's alignment with user requirements, verifying its readiness for the intended educational and entertainment objectives.

The culmination of the testing process is system testing, where the game is evaluated as a complete and integrated system to ensure it operates correctly in its intended environment. This includes assessing the software's performance, compatibility, and overall behaviour under different conditions. System testing is essential for identifying any issues that might not have been apparent during unit or integration testing, providing a final assurance of the game's quality and readiness for release.

Through this comprehensive testing documentation, our team ensures that every aspect of the game, from individual components to the complete integrated system, is rigorously tested and verified. This approach guarantees not only the functional excellence of the game but also a reliable and engaging user experience, fully showcasing the educational potential of this innovative programming game.

Middle school students of today have already lived a life with technology all around them. The internet has been present in their life since they were born, they were likely introduced to it and other software as soon as they were born. Many students today use software in school, using apps like Google Drive to store their school work. Very few assignments are even written by hand these days, students have to know how to use a computer and the software on it from a young age. Even after school, they'll rewind by scrolling through TikTok, watching a video on YouTube, or perhaps playing Minecraft with their friends from school. These examples illustrate how large of a role software will play in these students' life, it will be constantly present from the time they're born. If software is going to be such an important part of these students' lives, then they must learn the foundational skills needed to give them an understanding of how software is created. Coding is already becoming a part of the basic curriculum in public schools across Canada. Having a strong foundation of the skills that software engineers use every day when designing software can only be beneficial as children grow up in a world increasingly centred around tech. Software isn't going anywhere, as current middle school students graduate school and enter the professional world they do so in a world that is beginning to see artificial intelligence play a large role. The emergence of AI is only going to increase humanity's reliance on software, more job fields are going to require a base understanding of software as it becomes a larger part of all fields.

Our game is a cartesian puzzle-solving game. One where middle school and secondary school students will be challenged to put their logic and math skills to the test to solve a series of mazes. In doing so, they will be taught the basics of programming and have to put these newfound skills to the test to complete our game. The skills they learn in our game will equip the students with skills that will help them navigate a world that will only continue to see tech become an even more integral part of our lives.

## Terms, Notations, Acronyms

Terms	Meaning
Level	Level of difficulty. Several stages may be used to teach & test the same concept at different levels of difficulty.
GUI	Graphical user interface.
CLI	Command line interface.
AI	Artificial Intelligence.
Menu	An overlay that can be displayed on top of any screen.
Action Chain	An ordered list of action blocks that is executed from left to right by the gameplay logic.
ASCII	American Standard Code for Information Interchange, which is a character encoding standard.
Stage	A self-contained puzzle that has a set of starting conditions, available action blocks, and objectives. Completion of earlier stages is required to access later stages.
Screen	A self-contained GUI area that has a unique set of buttons/ text boxes.
Action Block	A rectangular GUI element can be chained together to make the character perform a series of actions to complete the objective of a stage.

Popup	A menu that opens in a new window.
-------	------------------------------------