

Introduction (Testing)

Table of Contents

- [Overview](#)
- [Objectives](#)
- [References](#)

Overview

"A Blocky Start" is an educational game, uniquely designed to make programming accessible and engaging for elementary and secondary school students. Set within an amazing block-based universe, this game provides the thrill of interactive gameplay while teaching foundational principles of coding, creating an exciting learning environment. By transforming complex programming concepts into hands-on challenges within an immersive game environment, "A Blocky Start" simplifies the learning process of many Assembly-style programming languages, making it enjoyable and educational.

During the development of "A Blocky Start", unit tests played a crucial role in ensuring the quality and functionality of the game. These tests allowed the development team to verify individual parts of the game code, ensuring that each function performed as expected. This methodical approach to testing helped to identify and rectify bugs early, contributing to a smoother, more reliable gameplay experience. This application of unit tests underscores our commitment to delivering an educational tool of the highest quality.

When the game is launched, the player is prompted to enter their name, and if they are a teacher or a developer, their password. This takes them to the main menu, with options to play a new game, continue a saved game, change settings, and view the tutorial and leaderboard. The progress is automatically saved, and players can continue the next time they log in. There are special modes for teachers and developers that are unlocked with a password. The teacher mode allows the user to access students' progress data. The developer mode additionally has bug reports and cheat codes. As players journey through the world of "A Blocky Start", they encounter a series of progressively complex cartesian puzzles and obstacles that require the manipulation of blocks while being introduced to coding concepts such as variables, control structures, loops, and functions.

The game screen has a maze in the centre and blocks like "left", "right", and "loop" at the bottom of the screen that can be dragged to the side to make an algorithm that when run, will traverse through the maze. This gradual learning approach ensures that students build on their knowledge incrementally, solidifying their understanding before moving on to more challenging tasks. The game's unique narrative places these challenges within engaging storylines and quests, giving players full control throughout the adventure they embark on. This approach not only boosts player involvement but also places programming skills within a wider context of solving problems, illustrating how coding can be applied in real-life situations. Collaboration features are also included in this game, which allows players to share solutions and strategies while working through challenges together. This social aspect encourages a communal learning experience, where players can learn from each other and collectively overcome the challenges found in this game.

"A Blocky Start" includes scenarios that reflect actual software development challenges, demonstrating the versatility and necessity of coding skills in the real world. This connection displays the value of the skills being developed and inspires players to explore further studies and career opportunities in this field. This game represents more than just an educational game, as it is a learning experience that aims to inspire a new generation of programmers. Through its engaging gameplay and educational content, "A Blocky Start" shows the capability and usefulness of programming, equipping students with the skills, confidence, and curiosity to explore the endless possibilities of technology.

Objectives

- Effectively integrate educational content
 - Develop the game to provide educational lessons through gameplay, utilizing a block-based universe to teach coding fundamentals such as variables, control structures, loops, and functions.
- Promote interactive learning
 - Ensure the game encourages active learning through coding challenges that progressively increase in complexity, allowing students to apply and reinforce their knowledge in engaging ways.
- Narrative-driven engagement
 - Create an exciting narrative that incorporates the educational content, ensuring players are fully immersed in the storyline, which enhances engagement and places programming skills within a wider context of solving problems.
- Create a collaborative environment
 - Design the game with features that encourage collaboration among players, promoting a shared learning experience and teamwork.
- Ensure accessibility and inclusivity
 - Make the game accessible to a wide audience by incorporating diverse learning styles and abilities, ensuring everyone can participate and benefit from the game.
- Provide learning support
 - Provide additional learning resources and information within the game to increase understanding and exploration of programming concepts.
- Build confidence in players' coding abilities
 - Design the game to gradually build players' confidence in their coding abilities, providing a supportive environment for experimentation where they can learn from their mistakes.
- Encourage creative application of skills
 - Allow players to use programming concepts in creative ways within the game, encouraging innovation and demonstrating the flexible nature of coding.
- Include assessments
 - Implement an assessment and feedback system within the game to monitor and support the player's learning progress, offering feedback to reinforce understanding.
- Follow proper software engineering practices
 - Ensure the development of the game follows the principles of software engineering covered in this course, including the use of design patterns, writing clean and efficient code, reflection on design decisions and more.
- Utilize appropriate development tools

- Develop the game using Java, and ensure compatibility with the necessary development environments and that we have adhered to all other requirements of the project.
- Demonstrate project management skills
 - Utilize project management tools like Jira, BitBucket, Confluence and more to ensure we efficiently navigate through the project as a strong team.
- Demonstrate technical skills and creativity
 - Display a high level of technical and creative quality in the game, and go above and beyond basic requirements to create an experience that is both educational and genuinely enjoyable for players.
- Ensure the game runs smoothly and provides the user with a bug-free experience
 - Implement comprehensive unit testing to ensure each game function behaves as expected.

References

- Project Specification
 - <https://owl.uwo.ca/access/content/group/aa2311c9-4cef-497f-8d9c-b502023be21c/project/CS2212%20Group%20Project%20Specification.pdf>
- Scratch Games
 - <https://scratch.mit.edu/search/projects?q=game>
- Are Games Effective Learning Tools? A Review of Educational Games
 - <https://www.jstor.org/stable/26388380?seq=1>
- JavaX Swing Complete Tutorial
 - <https://youtu.be/Kmgo00avvEw>
- Ways To Make An Engaging Game
 - <https://youtu.be/9vJRopau0g0>
- Test-Driven Development (Process)
 - <https://www.oreilly.com/library/view/test-driven-development/0321146530/>
- Model-View-Controller (UI Pattern)
 - <https://www.artima.com/articles/the-dci-architecture-a-new-vision-of-object-oriented-programming>
- JavaX Swing: Draggable GUI component
 - <https://stackoverflow.com/questions/874360/swing-creating-a-draggable-component>
- Zero Player Game
 - <https://youtu.be/N-BbgqOjIqk?si=hVPG4bRj5GMgloxW>
- Educational Video Game
 - https://www.researchgate.net/publication/238444705_Educational_Video_Game_Design_A_Review_of_the_Literature
- The DCI Architecture: A New Vision of Object-Oriented Programming
 - <https://www.artima.com/articles/the-dci-architecture-a-new-vision-of-object-oriented-programming>
- Test-Driven Development: By Example (book)
 - [https://github.com/clarabez/SoftwareTestingBooks/blob/master/Test-Driven%20Development%20By%20Example%20\(Kent%20Beck\).pdf](https://github.com/clarabez/SoftwareTestingBooks/blob/master/Test-Driven%20Development%20By%20Example%20(Kent%20Beck).pdf)