



ARCADEMIA Model-C

Alex's Space Adventure

[EFundi URL](#)

1 Project Overview & Objectives

Arcademia is an educational coding game that introduces Grade 5 students to fundamental programming concepts through engaging, story-driven gameplay. It operates entirely offline on a portable arcade console, making it accessible to South African Model-C schools regardless of internet connectivity. The game aligns with the CAPS curriculum by teaching sequencing, loops, conditionals, logical reasoning, and problem-solving in Mathematics and Life Skills.

- Introduce computational thinking through visual block-based coding.
- Teach core programming concepts: sequences, loops, and conditionals.
- Provide equitable access via an offline, low-cost arcade console.
- Develop critical thinking, pattern recognition, and debugging skills.
- Engage learners through narrative and progressive challenges.

2 Game Narrative

Astronaut Alex is stranded on an alien planet after a comet destroys the spacecraft's navigation systems. Players use the Arcademia console to help Alex navigate the planet, collect scattered spaceship parts across four levels, and ultimately repair the ship and return home. Each level unlocks progressively harder coding challenges, reinforcing narrative urgency as the ship's power weakens.

3 System Capabilities & Functionalities

Blockly Visual Interface — Player's construct code sequences using drag-and-drop blocks (move, jump, loops, conditionals) to control Alex through 2D platformer levels.

Four Progressive Levels — Level 1: basic commands; Level 2: loops; Level 3: conditional logic + repeat-until; Level 4: advanced conditionals and while loops — combining all concepts.

Role-Based Access — Separate flows for Students (gameplay, avatar, progress) and Teachers (student management dashboard, progress tracking, account administration).

Persistent Progress Tracking — Player scores, level completion, and block usage statistics are saved between sessions.

Avatar Customisation — Students personalise skin, hair, eyes, mouth, outfit, and accessories before creating their account.

Contextual Help System — A Help button on every level opens a hints popup tailored to the current challenge.

Adaptive Audio — Dynamic sound effects and background music change per level to enhance immersion.

Offline Operation — Runs on an HP EliteDesk PC inside a custom arcade cabinet — no internet required.

4 How the Program Works

The arcade machine presents a split screen: the left panel shows the Blockly workspace where players assemble code blocks, and the right panel displays the 2D game world where Alex executes those instructions.

Step	Description
1. Launch	The Title Screen loads automatically on startup; the player presses Start.
2. Role Selection	Player chooses Student or Teacher; logs in or signs up.
3. Level Selection	Main Menu shows Levels 1–4 with progress scores; player selects a level.
4. Code Construction	Player drags Blockly blocks into the workspace to create a program for Alex.
5. Run & observe	Pressing Run executes the block sequence; Alex moves through the level accordingly.
6. Feedback & Correct	If blocks are incomplete or Alex fails, a feedback popup guides the player to adjust the code. Reset Level restarts without clearing blocks.
7. Collect & Score	Successfully reaching the spaceship part triggers a score summary. Story cutscenes then advance the narrative.
8. Progression	Each completed level unlocks the next, with increasingly complex block types required.

5 Arcade Console Controls

Control	Function
Joystick	Navigate menus, move cursor, scroll Blockly workspace
Left Click Button	Confirm selections; standard left mouse click
Right Click Button	Context actions in Blockly (duplicate/delete block)
Run Button	Execute the current Blockly code sequence
Clear Button	Remove all blocks from the workspace
Power Button	Safely power the arcade machine on/off
Change / Sequence Button	Switch between block sets (basic ↔ repeat/if)
Help Button	Open level-specific hints popup
Back Button	Return to previous menu / screen

6 User Roles

Student / Player — Primary user. Plays levels, customises avatar, tracks personal progress.

Teacher / Facilitator — Monitors class progress, manages student accounts, resets passwords, reviews gameplay data.

Game Administrator — Manages the overall environment, user accounts, and access permissions.

Developer / Sys Admin — Handles bug fixes, feature additions, system maintenance, and updates.

Tester — Validates gameplay correctness and smooth user experience throughout development.

7 Technology & Platform

Arcademia is built on Godot Engine 4 with a Blockly-style visual programming interface. The game runs on an HP EliteDesk desktop PC embedded in a custom arcade cabinet. Minimum requirements for PC operation: 8 GB RAM, dual-core CPU, OpenGL 3.3-capable GPU. Input is handled via joystick and physical arcade buttons no keyboard needed. Output is delivered via monitor (GUI) and speakers (adaptive audio). All data including user accounts, progress, and scores is stored locally with no cloud dependency.

8. Development Team

Project Manager: Melanie Kruger

Anrich Kruger, Batlile Mboweni, Charlton Chivhere, Kyle Ogle, Siphokazi Malesa, Stefan Venter, Justine Sudbury, Dumisani Mlangeni, Sulaiman Sulaman, Natasha Cloete, Dean Dube, Tumelo Maboea

8 Conclusion

Arcademia successfully bridges the gap between game-based learning and structured curriculum delivery. By combining an immersive narrative, progressive coding challenges, and an accessible offline arcade format, the project equips Grade 5 learners with foundational programming skills while remaining practical for under-resourced South African schools. The system serves as both an engaging learning tool for students and a classroom management platform for educators.
