Members: Dakota Berbrich, Noah Berkland, Aaron Goff, Nolan Jessen
Advisor: Dr. Doug Jacobson

Client: Information Assurance Center

Curriculums to Use the Sphero RVR in IT-Adventures

Problem Statement
- Some Iowa high schools lack effective coding programs for students who wish to learn how to program.
- Programming is becoming a more needed skill and not having is is putting some students behind the curve of their peers when joining the workforce.
- Existing programs that introduce students to programming like FIRST® LEGO® League can be expensive and difficult to manage for schools with no previous experience.

Solution Statement
- Our team aims to develop a curricula to fulfill the needs of these students
- We have written 2 courses to teach programming at different levels both utilizing the Sphero RVR
- Robotics focuses on using basic building blocks to teach simple programming concepts
- Smart-IT focuses on higher level programming and utilizing the fine controls of the RVR and its sensors

Week of	Robotics Lesson	Smart-IT Lesson
September 7th	Introduction to robotics and the RVR	Introduction to Smart-IT
September 13th	Moving the RVR	Familiarization with the RVR
September 20th	Basic micro:bit Input/Output	Assembly of RVR and introduction to the Pi
September 27th	Variables	First Driven Program and Introduction to Python
October 4th	If Statements and Conditionals	Variables, Inputs, Outputs, and String Conversion
October 11th	While and For Loops	If Statements and Conditionals
October 18th	Arrays and For Loops	Iterations: For and While loops
October 25th	Practice Challenge 1	Monthly python and RVR challenges
November 1st	Button & Buzzer [Digital I/O]	Lists and Strings
November 8th	Servo & Slide Dimmer [Analog I/O]	Functions and Modules
November 15th	Catch up/play week	Objects and Classes
November 22nd	No new material	Catch up week
November 29th	Proximity Sensor	Monthly python and RVR challenges
December 6th	Practice Challenge 2: Factory Robot	Dictionaries and JSONs
December 13th	Remote Control	Catch up week

Standards
Goal is to teach students programming in reliable and accepted ways, using JSON, CSTA K-12 CS, and IEEE Programming standards.

Functional Design Requirements
- Two sets of curriculum, one focusing on introductory coding and one focusing on more advanced coding
- Curricula should be hands-off for teachers; their role should be more supportive
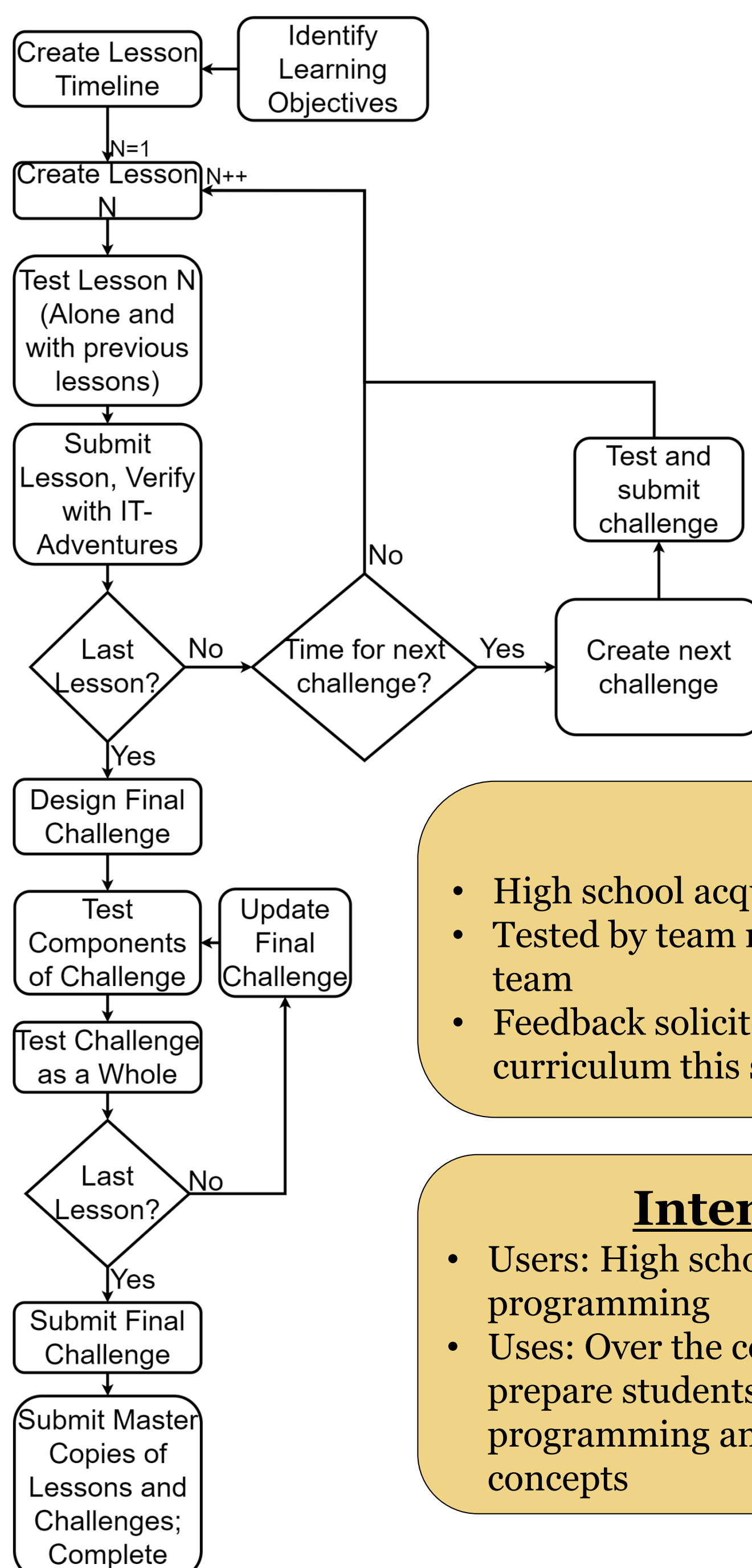
Non-Functional Design Requirements
- The curricula should be accessible to students with any amount of coding experience.
- The challenges must ramp up in difficulty slowly / at a standard pace.
- The final course must be reliable and easy to put together.
- The kits must be cheap and easily supplied.

Operating Environment
- High school classrooms and homeschool locations in Iowa

Technical Details
Hardware
- Sphero RVR
- micro:bit
- littleBits
- Raspberry Pi 3B+

Software
- MakeCode for micro:bit
- Thonny for Raspberry Pi
- Block Code
- Python

Project Resources
- 4x Complete Robot Kits:
  - Sphero RVR, Raspberry Pi 4, Raspberry Pi 3B+, GPIO / microUSB cables, micro:bit, littleBit topper kit

Testing
- High school acquaintance tested instructions for Smart-IT
- Tested by team members before verified by IT-Adventures team
- Feedback solicited from teams running through curriculum this semester

Intended Users and Uses
- Users: High school students who wish to learn programming
- Uses: Over the course of a school year, the program will prepare students to understand the uses and limitations of programming and further bolster their ability to learn the concepts