RCJA 2025 OnStage Technical Description Paper

Team Information

Division: Open OnStage

Team Name: The Dino Nuggies

School: Lighthouse Christian School

State/Territory: Queensland

Team Mentor: Mr Caesar Al-Samarrie

Mentor Attending Interview (Yes or No)

Mentors are able to attend the RoboCup Junior OnStage Interview in a supportive capacity only (please see OnStage Rules for details). Please indicate if your team mentor wishes to attend.

Team Member Names:

If any team member had a specific role, such as designer, engineer, programmer, prop design or manager please include this below.

Member 1: Allira Fleming – secondary costume designer, set/prop designer

Member 2: Emily Scofield – primary costume designer, set/prop designer, builder

Member 3: Madeleine Van Mello – primary programmer, builder

Collaboration

It is the overall desire of RoboCup Junior events that any technological and curricular developments will be shared with other participants after the event. Any developments including new technology and software examples, may be published on the RoboCup Junior website after the event, furthering the mission of RoboCup Junior as an educational initiative.

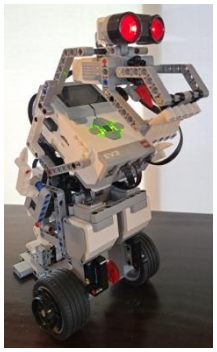
Submission

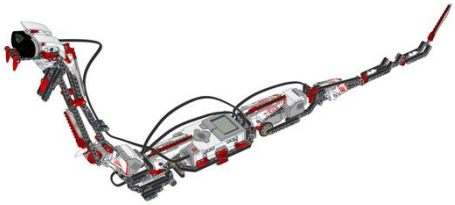
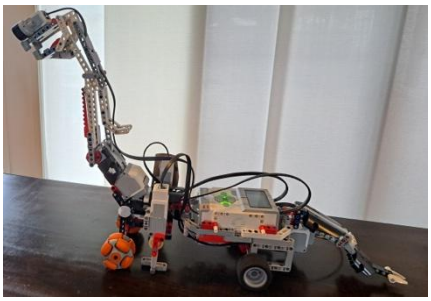
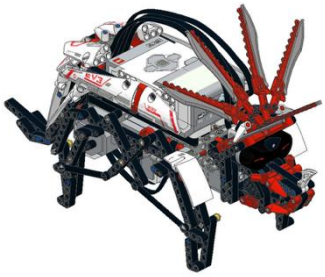
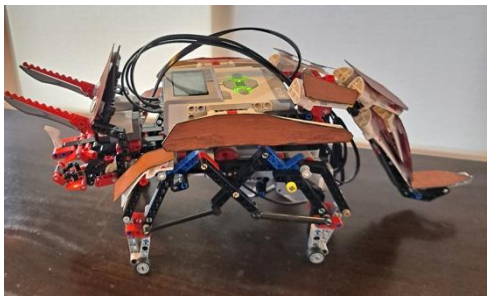
The RoboCup Junior Technical Description document must be able to be submitted as a **PDF file** to the RCJA Registration portal prior to the competition. Please note there is a maximum file size for the portal.

Hardware and Engineering	Showcase & Explain

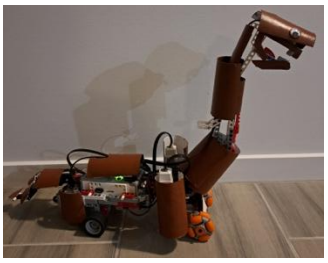
• Design and Construction
• Mechanical Innovation
  ○ How does the design enhance the robot(s) performance?
• Provide images and zoom into special features
	• Showcase your robot(s) design features
	• Include images of your planning of your robot(s) build
	  ○ Is this your own design?
	  ○ Did you gather ideas from other robot builds?
	• Special features: how you made the robot(s) move smoothly, keep balance, communicate, avoiding objects, grasping objects
	• How is your robot(s) stable?
	• Have you added costuming to your robot(s)? How do you make sure your robot(s) still works with the costume?

List of robots, hardware, design, function:

Robot	Hardware	Design	Special Feature/s
Jeep	EV3	Original, Costume designed to frame but not cover sensors, wheels, buttons, screen.	Main robot, Line follower using both reflective light and colours, Master EV3 sending messages to other EV3s over a Mobile Wi-Fi Network.
Tyrannosaurus Rex	EV3	Inspired by 'GyroBoy' (see image below), Costume designed to frame but not cover sensors, wheels, arms, motors, buttons, screen. 	Main robot, Self-balancing using gyroscope, Ultrasonic sensor used to 'attack' Jeep, Balancing code adapted from GyroBoy code. 

Brachiosaurus	EV3	Inspired by 'R3ptar' (see image below), Costume designed to frame but not cover sensors, wheels, actuators, motors, buttons, screen. 	Main robot, Linear actuators used to raise body, Infra-red sensor used to detect an IR beacon. 
Triceratops	EV3	Inspired by 'DinoR3x' (see image below), Costume designed to frame but not cover sensors, legs, motors, buttons, screen. 	Main robot, Walking by using two motors and a touch sensor Infra-red sensor to detect an IR ball. 
Gate	SPIKE Prime	Original, Costume designed to cover tiered sides, doors and arch but not cover sensors, motors or lights.	Moving prop, Master SPIKE Prime sending messages to other SPIKE Primes using Hub Bluetooth broadcast messages Technic Light Cables used to create the illusion of torches.
Nest	SPIKE Prime	Original, Costume designed to hide motors and create the illusion of a dinosaur nest but allow for full shaking movement.	Moving prop, Multi-directional 'shaking' movement, Ultrasonic sensor used to detect the Jeep.
Fossil Dig	SPIKE Prime	Original, Costume designed to hide motors and create the illusion of a dirt mound but allow for full shaking movement.	Moving prop, Multi-directional 'shaking' movement, Ultrasonic sensor to detect the Jeep.
Tree	SPIKE Prime	Original, Costume designed to hide motor and create the illusion of a large tree but allow for full shaking movement.	Moving prop.

Photos of robots with costumes:

Jeep	Tyrannosaurus Rex	Brachiosaurus	Triceratops
			
Gate	**Nest**	**Fossil Dig**	**Tree**
			

Software and Sensors

- Coding language used for robot(s) and sensors
- Annotate to explain any interesting code:
  - Use of sensors
  - Special movements or interactions
- Reference any sources or libraries that you are using
- Provide images and zoom into your code

Showcase & Explain

Coding

- What programming language(s) have you used to move and interact?
- Have you programmed an interaction between a robot and other robot(s), props or with a human performer?

Sensors

- How have you used sensors to enhance your robot(s) movements and actions?
- How do the robot(s) use these sensors?
- Do the performers interact with the robot(s) via sensors?

We have used PyBricks for the SPIKE Primes and EV3Dev Python and PyBricks MicroPython for the EV3s (all text-based programming languages). For the robots to run the PyBricks software, they were all reflashed using the PyBricks website firmware. We are using PyBricks hub-to-hub communication for the SPIKE Primes and a Mobile Wi-Fi Network for the EV3s to allow the robots to exchange messages.

List of robots, software, programming language:

Robot	**Software**	**Programming Language**
Jeep	Visual Studio Code	PyBricks MicroPython
Tyrannosaurus Rex	Visual Studio Code	PyBricks MicroPython
Brachiosaurus	Visual Studio Code	EV3Dev Python
Triceratops	Visual Studio Code	PyBricks MicroPython
Gate	code.pybricks.com	PyBricks
Nest	code.pybricks.com	PyBricks

Fossil Dig	code.pybricks.com	PyBricks
Tree	code.pybricks.com	PyBricks

Photos of code (special features):

Jeep	Tyrannosaurus Rex	Brachiosaurus	Triceratops

Gate

```
# Initialize brick
# Broadcast on channel 1
hub = PrimeHub(broadcast_channel=1)
# Send message
def send_message(message):
    print('Sending message', message,'...')
    # Change button colour, send message
    hub.light.on(Color.RED)
    hub.ble.broadcast(message)
    # Change button colour
    hub.light.on(Color.GREEN)
    wait(wait_time)
```

Nest/Tree

```
# Shake nest & tree at frequency of change
def shake(frequency):
    # Initialize angles, counter, range
    temp_rotate = angle_rotate * -1 / 2
    temp_shake = angle_shake * -1 / 2
    counter = 0
    times = 2
    print('Shaking at', frequency, 'frequency...')
    # Reset nest & tree motors at same time
    nest_right.run_target(speed_motor, temp_shake, wait=False) # Shake motor
    tree_right.run_target(speed_motor, temp_rotate) # Right tree motor
    temp_rotate = angle_rotate
    temp_shake = angle_shake
    # Repeat 2 times
    for counter in range(times):
        # If frequency low, choose low speed in deg/s
        if frequency == 'low':
            speed_shake = randint(50,100)
        # If frequency high, choose high speed in deg/s
        else:
            speed_shake = randint(100,200)
        # Shake nest &/or tree, run motors at same time
        nest_right.run_angle(speed_shake, temp_shake, wait=False) # Shake motor
        tree_right.run_angle(speed_shake, temp_rotate) # Right tree motor
        # Toggle angle to move motors back & forth
        temp_rotate = angle_rotate * -1
        temp_shake = angle_shake
```

Fossil Dig

```
# Shake dig at frequency of change
def shake(frequency):
    # Initialize angles, counter, range
    temp_rotate = angle_rotate * -1 / 2
    temp_shake = angle_shake * -1 / 2
    counter = 0
    times = 2
    print('Shaking at', frequency, 'frequency...')
    # Reset dig motors at same time
    dig_left.run_target(speed_motor, temp_rotate, wait=False) # Rotate motor
    dig_right.run_target(speed_motor, temp_shake) # Shake motor
    temp_rotate = angle_rotate
    temp_shake = angle_shake
    # Repeat 2 times
    for counter in range(times):
        # If frequency low, choose low speed in deg/s
        if frequency == 'low':
            speed_shake = randint(50,100)
        # If frequency high, choose high speed in deg/s
        else:
            speed_shake = randint(100,200)
        # Shake dig
        dig_right.run_angle(speed_shake, temp_shake, wait=False) # Shake motor
        # Rotate dig run motors at same time
        dig_left.run_angle(speed_shake, temp_rotate) # Rotate motor
        # Toggle angle to move motors back & forth
        temp_rotate = angle_rotate * -1
        temp_shake = angle_shake
```

List of robots, sensor, type of interaction, usage:

Robot	Sensor	Type of Interaction	Usage
Jeep	Ultrasonic	Robot-to-prop	Obstacle evasion, T-Rex detection
	Colour	Robot-to-prop	Line following (with both reflective light and colours)
	Touch	N/A	Calibration of colour sensors
Tyrannosaurus Rex	Ultrasonic	Robot-to-prop/Robot-to-human	Following of 'flare' held by human performer
	Ultrasonic	Robot-to-robot	Jeep 'attack'
	Gyroscopic	N/A	Self-balancing
Brachiosaurus	Infra-red (works with IR beacon)	Robot-to-prop/Robot-to-human	'Eating' of branch held by human performer
Triceratops	Infra-red (works with IR ball)	Robot-to-prop/Robot-to-human	Following of 'egg' held by human performer
Gate	Distance	Robot-to-human	Opening of the doors by human performer
	Distance	Robot-to-robot	Closing of doors by Tyrannosaurus Rex
	Force	Robot-to-human	Fail-safe: Opening of the doors by human performer
Nest	Distance	Robot-to-robot	Activation of egg hatching
	Force	Robot-to-human	Fail-safe: Activation of egg hatching by human performer
Fossil Dig	Distance	Robot-to-robot	Activation of Palaeontologist waving
	Force	Robot-to-human	Fail-safe: Activation of Palaeontologist waving by human performer

Performance	Showcase & Explain
• What is the concept or theme of your performance? • Provide a description of your performance	• What are your favourite parts of your performance? • What are you most proud of in your performance, team, design or code?

The concept of our performance is a tour of Jurassic Park, showcasing famous dinosaurs like the Brachiosaurus, Triceratops, and Tyrannosaurus Rex. The tour is suddenly interrupted by a Tyrannosaurus Rex attack. Following the attack, panic ensues despite the tour guide's best efforts to keep everything under control, and, in the end, the T-Rex wins out, ending the performance with the iconic roar from the Jurassic Park movies.

Our favourite parts of the performance are when the T-Rex appears, and the guide and visitors go into 'panic mode' while trying to defend themselves from the oncoming attack and when the easily-distracted visitor gets fed up and throws a 'tantrum' because she doesn't see any dinosaurs.

We are the most proud of the interaction between the EV3s and the SPIKE Primes, especially through the use of mobile Wi-Fi.