UNIVERSITY CATEGORY

FINAL ROUND

Task Specifications

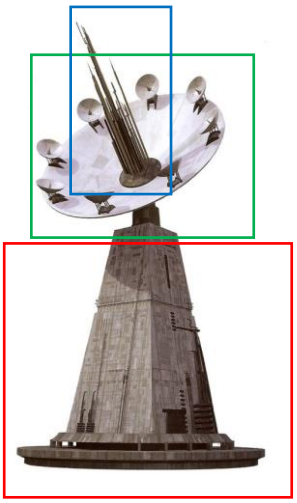
University Category

After the battle of D'Qar with the First Order, the resistance is fleeing across the galaxy in search of secure hiding places. After finding two planets, resistance admirals have decided to use invisibility shields to protect the planets. In order to generate a shield, one would need a base (red box), a shield generator (green box) and a power source (blue box). The parts must be placed exactly in this order, from the bottom to top. Resistance pilots Poe Dameron and Kazuda "Kaz" Xiono have volunteered to search the black market for smuggled parts to avoid unnecessary attention.

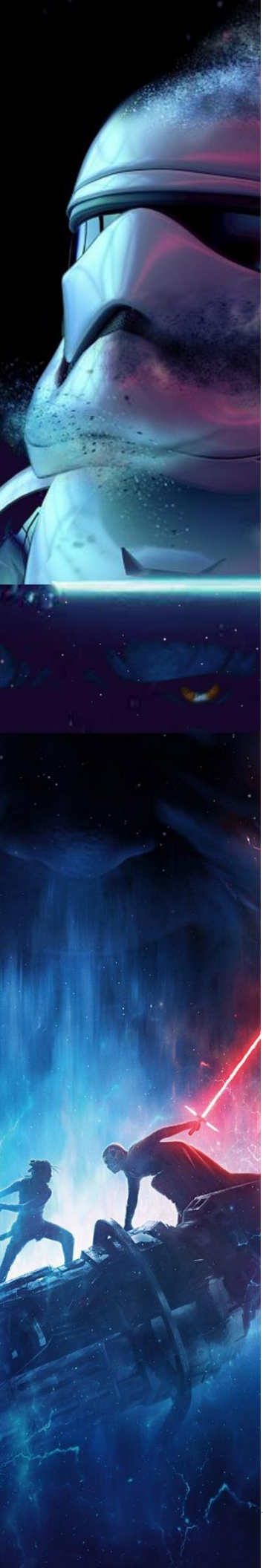
The First Order has placed automated turrets (inverted patches) at locations which they suspect of resistance travels. Luckily, the resistance has found two keys (white boxes) which disable the turrets at the gates and allow safe passage. Smugglers might also have some keys that the pilots can use.

Once again, the resistance needs your help to secure their hideouts and live to fight another day.

May the Force be with you!



The shield



Introduction

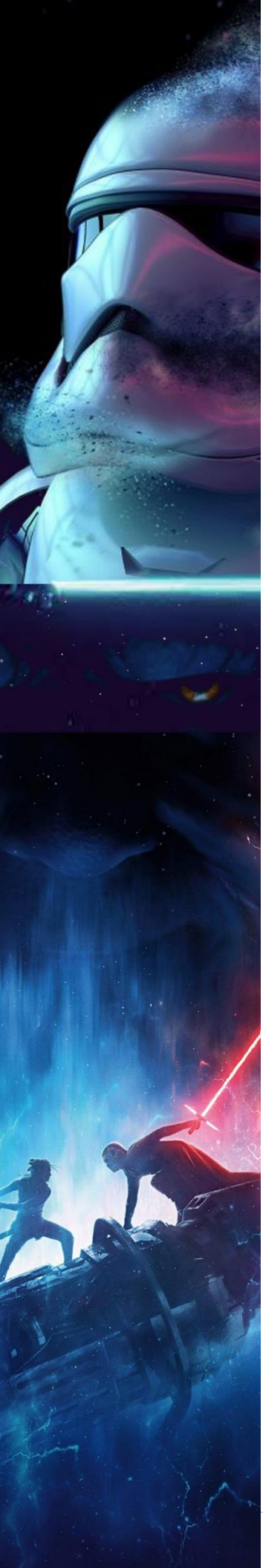
For this task, you are expected to design two virtual robots within the limits specified, using the Webots Open-Source Simulator (https://cyberbotics.com). This round will account for 100% of your total mark for the final round.

Task Description

- The virtual round will be conducted using Webots R2021a.
- Task of the final round is to be completed using two virtual robots which have the following capabilities.
    - Line following
    - Box detecting
    - Having an arm capable of moving and carrying boxes
    - Detecting the color of the boxes
    - Communicating between the two robots (Optional)
- First, the robots will be placed on the starting white squares.
- Then, the robots must proceed to the maze area by following a white line on a black surface. The maze is a grid consisting of white and red line segments on a black surface. Robots must follow only white lines, and they are not allowed to move along red lines.
- While exploring the maze, the robots must collect **red** (**R), green (G)** and **blue (B) colored boxes** that are placed in the maze, carry them to the two starting squares, and stack them. Order of the stack should be the red color box (bottom), green color box (middle) and blue color box (top) as shown in *Fig 1*. At the end, there should be two stacks in the two starting squares.

Fig 1. Final Expected Stack

- R, G, B boxes are placed in the ***white patches*** (*Fig 3*) in the junctions, and there may be one box, or two boxes stacked in the *white patches* (not necessarily in the center). That is, if there are two boxes in a *white patch*, they will be stacked on top of each other in a way such that the smaller box will always be on the top. As shown in example *Fig 4*, the green box **must** be on top of the red box as it is smaller than the red box. No boxes will be placed on ***inverted patches*** (*Fig2*). Penalties will be given, each time the R, G, B boxes touch the black surface during the task.



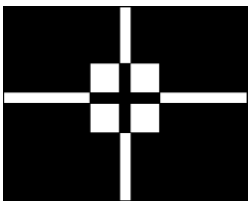


Fig 2. Inverted patch

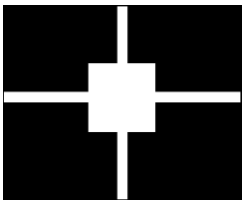


Fig 3. White patch

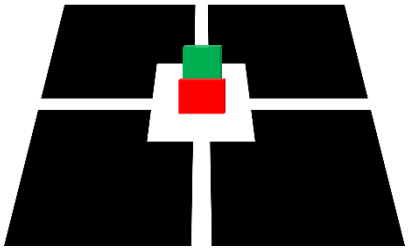


Fig 4. Two boxes in one patch

- While exploring the grid and searching the boxes, robots can **only** pass-through the **middle** of the inverted patches under one condition. That is, to pass through them, the robot should move a white box into one of the four adjacent cells to the *inverted patches,* which are bounded by red dotted lines (The robots can only pass the middle of an *inverted patch* as long as there is a white box placed in one of four adjacent cells to the patch as shown in *Fig 6*). The robot is allowed to take along that white box after passing the *inverted patch*. Then the *inverted patch* no longer can be passed since there is no box in a cell (*Fig 5*). If a robot passes through the middle of an *inverted patch* violating the above rules, **serious penalty marks** will be given, and you will **not receive any time bonus.**

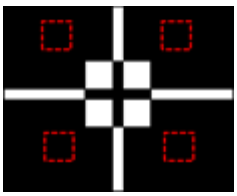


Fig 5. Not allowed to pass-through

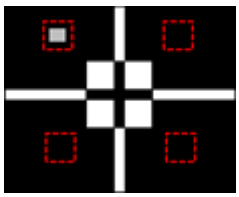
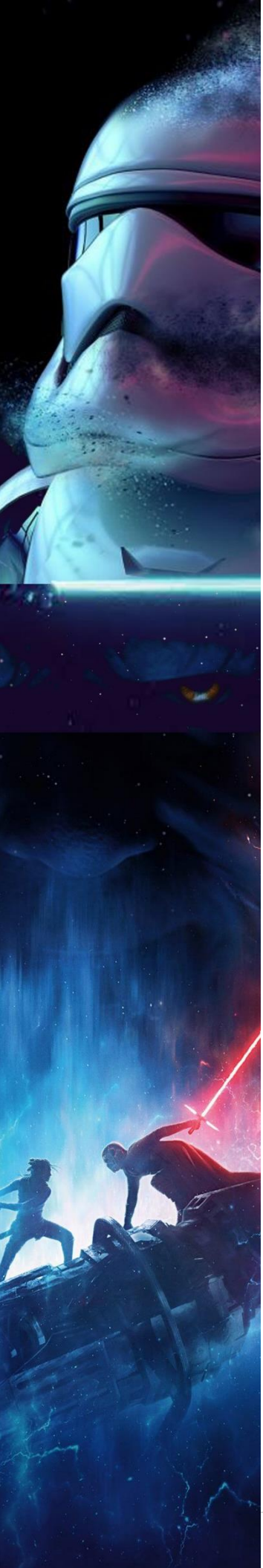


Fig 6. Allowed to pass-through

- Two white boxes will be placed in two predetermined places (*Fig 9*) near the starting squares for the two robots. Other white boxes can be found in the *white patches* as similar as the R, G, B boxes scenario.
- One robot is allowed to travel **only with one box** at one time. That is, a robot cannot travel carrying multiple boxes. This condition is for both RGB and white boxes.
- It is not mandatory to carry the white boxes that are placed near the starting squares.
- The two robots are allowed to communicate with each other while completing the task.
- The task is considered to be completed when the two robots complete both the stacks in the two starting squares.



Arena Details

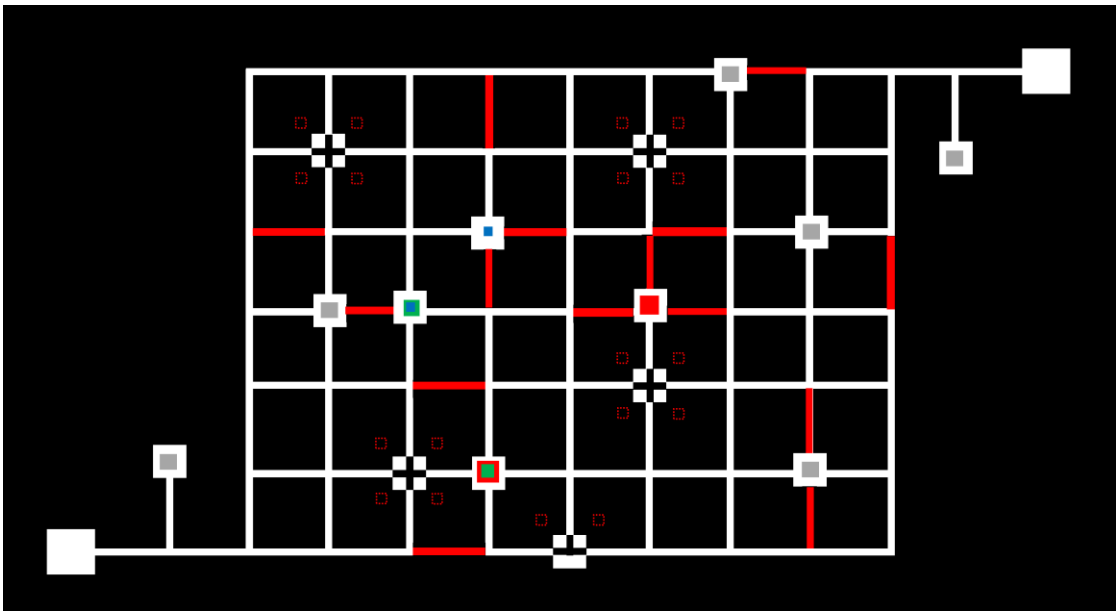


Fig 7. Sample arena
(This is for illustration purposes only. Actual arena can be different than this.)

- The arena contains two white starting squares, white line following segments that resemble a maze, red line segments, *white patches*, *inverted patches*, and dotted line cells. (See *Fig 7*)
- Both starting squares are 25cm × 25cm white color squares.
- All the white and red line segments are 3cm wide.
- The maze grid consists of 8 × 6 black squares and the dimensions of a square are 60cm × 60cm.
- The white patches that contain boxes have the dimensions of 20cm × 20cm. Some patches can be empty at the game play.
- The inverted patches are 20cm × 20cm and the width of the black line segments in those inverted patches is 3cm.
- The dotted line cells have the dimensions of 10cm × 10cm, and the red dotted lines that enclose the cells have 1cm thickness.
  (Refer to the *Fig 9* to get more understanding about the dimensions of the arena)
- There will be four types of boxes (Blue, Green, Red, White) to be collected. The weight of a box will not exceed 150g and their dimensions are given in *Fig 8*.

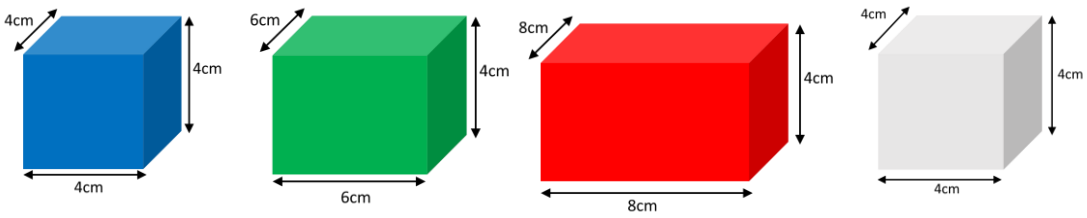
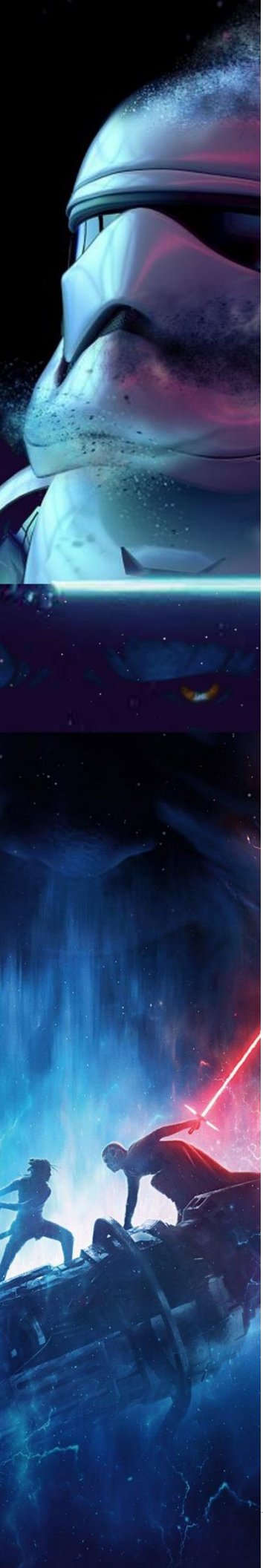
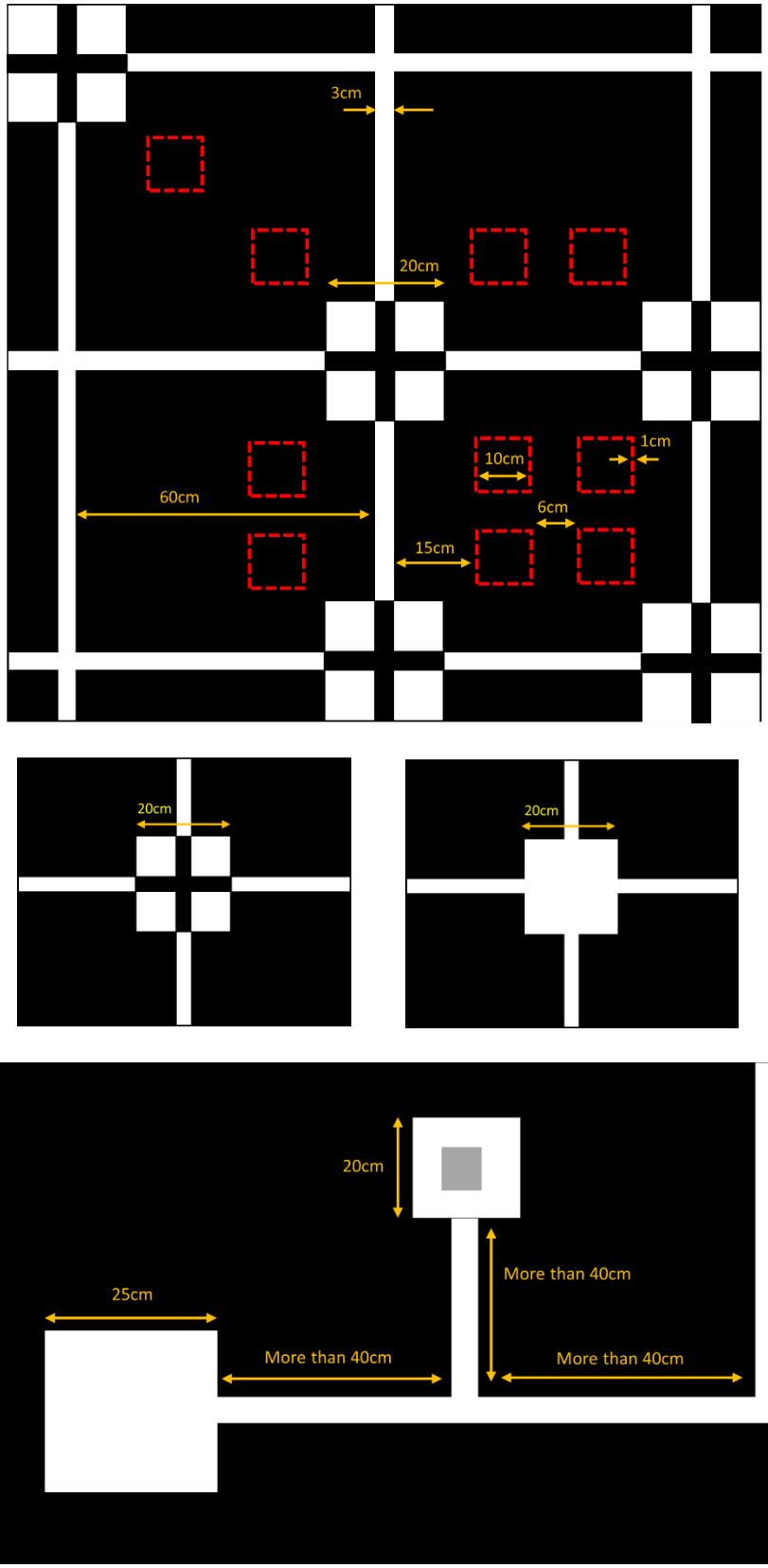


Fig 8. Dimensions of boxes

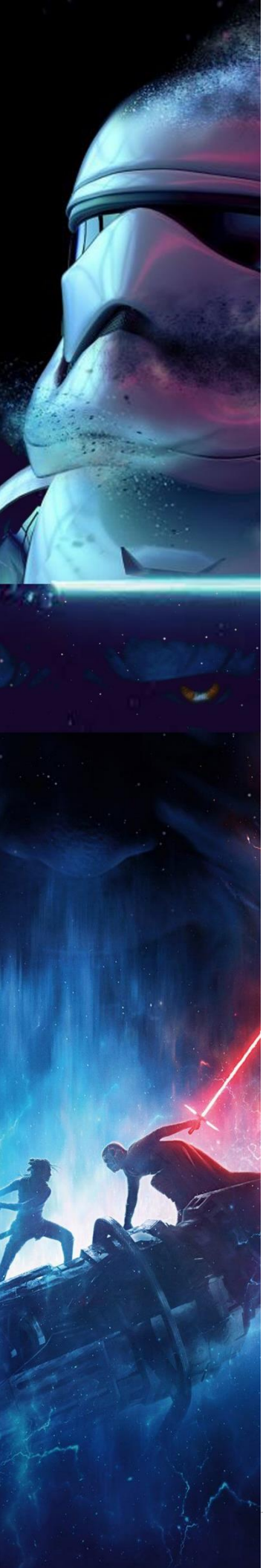
- All the contact properties of the arena and the boxes will be default values in Webots.





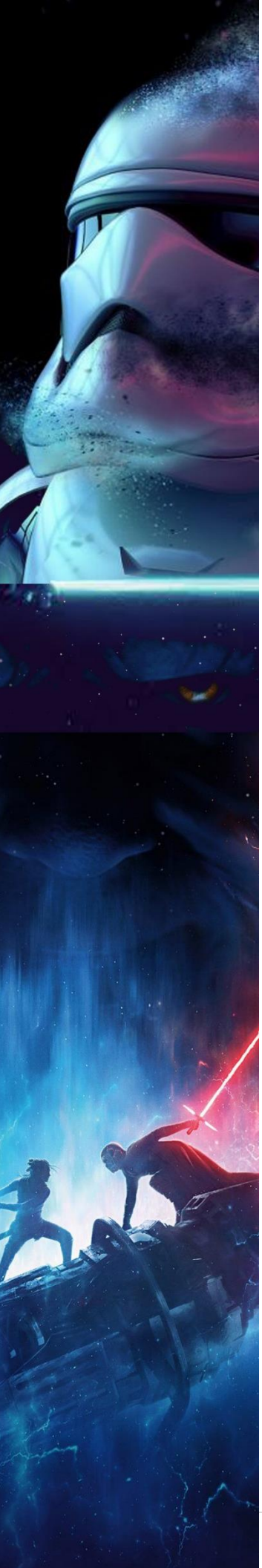
Starting area

Fig 9. Dimensions of the sample arena



Rules and Restrictions

- Both robots have the same restrictions.
- The robots must be completely autonomous, with the only expectation of communicating with each other.
- Maximum speed of the robot: 15 cm/s
- Maximum range for IR sensors (line follow): 5cm
- Maximum range for other distance sensors: 150cm
- Maximum rotational speed for servo motors: 7 rad/s
- Maximum torque for servo motors: 1 Nm
- Maximum reachable distance of the arm: 25cm
- Cameras (in Webots) can only be used for box color detection tasks as a "color sensor".
    - Maximum field of view for camera: 0.3 rad
    - Maximum range of camera ('far' in webots): 3cm
- All sensors, and actuators used in the robots should correspond to a real-world element **without violating above restrictions**. (Model number and datasheet should be available). All parameters of the simulated sensor/actuator should be configured to be very close to the real-world element.
- If you are using a distance sensor which has a range more than 150cm in real world, (e.g., sonar sensor) you must restrict the maximum range in webots to the above-mentioned value.
- Minimum dimensions allowed for a robot are 10x10 cm and the maximum dimensions are 25x25 cm. (without arm extended)
- Maze exploring should be done using **white line following**.
- Always a part of a robot **must be on a white surface**. The robots are not allowed to completely move to the black surface.
- When placing the white boxes in cells, the boxes must be **completely inside** the red dotted line boundary without touching the red boundary. Otherwise, some penalty will be given.
- Contestants are not allowed to use built-in Proto Robots. You are required to build your own robots.
- Two robots do not need to be identical.
- Teams are allowed to use any communication method between the two robots.
- C++ is the only allowed language for coding.
- Each team is given 3 attempts within 15 minutes to complete the task. Code changes will not be allowed after submission. Before each attempt, the robots and the arena will be **reset**.



Marking Criteria

- Time taken for the completion of the task. (The key factor)
- Accuracy and smoothness of robot movements
- Realistic nature of robots
- Navigation techniques and algorithms
- Cost efficiency of robots / part utilization.
  - Approximated cost will be calculated using the prices in www.pololu.com
  - Maximum total cost for sensors and actuators in both robots: $800.00

At least one robot should be able to **enter the maze** to qualify the team to gain marks. Otherwise, the team will not receive any marks.

Submissions

You are required to submit the following files:

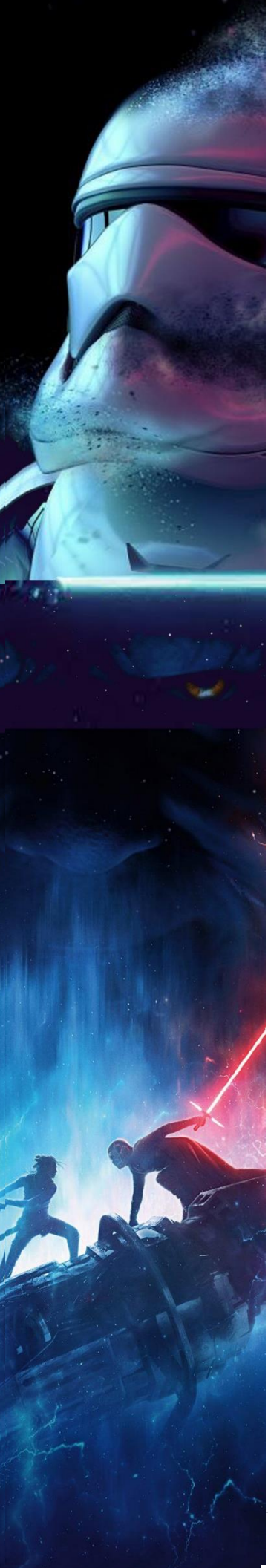
- Content of the entire project you have worked on, compressed as a ZIP file.
- ".CPP" file of your robot controller. If you are using separate header files, include them separately.
- Your robot nodes, exported as ".wbo" files.
- All the additional Computer Aided Design (CAD) files (e.g. .STL files) that are inside your "worlds" directory, compressed as a ZIP file and named as "CAD_files.zip". This step is optional if you do not use separate CAD files.
- Datasheets of all the sensors and actuators that you used, compressed as a ZIP file and named as "Datasheets.zip".
- A separate text file briefly explaining your comments for the evaluators for setting up your robot for final evaluation.

Please **double check** before submitting whether your submission is working on another computer. When submitting your files, please put them into one zip file and rename it as follows.

- <Team_Name>_<Webots>

Judging

- The judges have the full authority in giving marks in every round.
- The judges can ask for an explanation about your submissions by arranging a viva examination.
- Decision of the panel of judges will be final.
- Your team can be disqualified at any moment for not following the rules and restrictions stated in the task document.
- Late submissions will be neglected.



Additional Information

- For evaluation purposes the “basicTimeStep” field of the “WorldInfo” node will be set to 32ms. (Similar to stage 01)
- The lighting of the arena (e.g., using “DirectionalLight” nodes) will be defined in the sample testing arena that will be provided and will not be changed during evaluation.
- Evaluations will be conducted on a windows laptop with a dedicated GPU available.

Any further changes will be informed to the team leaders via emails.

Contact

Tharuka Nilupul
- 0717665259
- tharukanilupul@gmail.com

Chathumi Navodya
- 0715346486
- chathuminavodya11@gmail.com

Amashi Niwarthana
- 0779112504
- amashiniwarthanaoffi@gmail.com