Notes from first meeting

Notebook:	First Notebook		
Created:	28-Feb-16 7:48 PM	**Updated:**	28-Feb-16 7:48 PM
Author:	Mark Craig		

4 robots minimum: Glasgow, Commerce Street, Dumfries, Wick High School
since discussed with WHS, no way to get internet access for the robots, neither through school network (sounds like it just wouldn't happen) nor through a mobile dongle (poor phone service at school). Still, what about a virtual robot in a panoramic photo of the MAKLab WHS room(s)?

Remote control spider from Mars - even just as an obstacle or jumping out as a monster.
hacked furby/teddy ruxpin/etc (don't want to scare the little nippers though do we? don't we?)

Preplan instructions for robot, send them, then see how good/accurate your instructions were. Instructions can take a few seconds/a minute to transmit. Explain this as similar to how NASA control the Mars rover, except they have a 20+min delay. Queued up instructions show up on a screen like arduino code?

Game types

- Arcade "claw", robot arm to grab something and transport it to target
- Distance to target
- Time to target
- Leaderboard to show who has done the best on the day. If there was some sort of inter-location game then it could be a scoreboard/league board.
- two robots in one location? king of the hill, racing, sumo, capture the flag etc
- shoot out targets with ping-pong gun or ring/dart launcher

Controllers

- Arcade controller/cabinet, branded with "Moons of MAKLab" livery. 2 player or 2 cabinets so players can face off against each other (two robots in one location? king of the hill, racing, sumo etc)
- dance mat controllers (get enough together to make a big dance mat big trak controller!!!)
- Reconstruct controller [you have to fix the controller when it breaks down (on purpose) to complete the task, something akin to the 'hacking' in GTAV]
    - Patch cables to re-wire your controller (i.e. joystick horiz/x-axis is red wire, plug it into terminal with red LED), RGB LEDs to allow for computer to randomise the patch wiring.
    - Instead of rewiring, you have to use pipes to redirect airflow to drive ping-pong balls around a "circuit"!
- Electric shock controllers (i.e. as penalties for hitting certain obstacles). - User Hostile Interface
- whack-a-mole buttons using capsense to detect proximity to a lit button move illumination to another button as player goes to press it. This could have zero impact on the gameplay, just a visual distraction.
- Piano keyboard as part of the interface, clues point you towards playing a classic arcade/film/etc theme, i.e. close encounters, to complete the challenge. Clues could be hidden behind obstacles, so you need to drive around the area and spot all the clues (i.e. 2nd note is middle C). [Change up the clues/tune between players to avoid someone just repeating the same tune to win in record time. what about spotting a QR code with the camera triggers the clue to register on screen?]

Interaction between MAKLab locations?

Twitter controlled obstacles [allow audience to participate by making the obstacles interrupt players route], which would also allow people at MAKLab locations some control/input.

Video Feed

- Blimp camera (helium filled air ship with RC fans) or camera hanging from 3-point gantry/pulley

arrangement. could rig it up to look like a quadcopter or a lunar lander.
- Play around with camera feed. Mirror the image. pixelate it for 8-bit style. black and white, add interference, low resolution for Apollo lander (grab actual audio for soundbytes to play during mission)
- Cameras don't have to be on the robots, there could be static/fly-on-the-wall ones. These could even be remotely operated from EMMF (i.e. security camera).
- Gamer sees something different from everyone else [using Euan's headset, have audience making suggestions/correcting the player].
- We're going to want lots of monitors and maybe a projector!

Rube Goldberg machine, marble run (what if we could have the camera in the marble!!! i.e. a hamster ball with a self-righting camera inside) or "TOMY screwball scramble game" [though timing critical games might not work too well with internet lag]

Hardware
Raspberry Pi (WiFi/Internet, limitations are only one hardware PWM on GPIO)
Arduino (to handle PWM and servo signals, taking instructions from Raspberry Pi)
ESP8266 as 1 dimensional robot (monster/obstacle which jumps out or moves along a line/track)

Software/Language/Coding
Which languages do we all use? Is it determined primarily by the hardware we choose (cart before horse?)? We selected Python on Raspberry Pi as it has an abundance of libraries and support/tutorials/examples available.

MQTT Broker
or If This Then That (IFTTT) - would it be fast enough?

Goals for next week
- [x] Set up communication channels (Facebook group and GitHub repositories were decided upon but we're still open to suggestions)
- [ ] Build a resource list
- [ ] Blair to get a raspberry pi onto his rover