1 Tank Drive

1.1 Cost

- Needs only 2 gearboxes. (4 if designed for mecanum) (upgraded if 6 motors used)
- Motors used is up to the year, with options for 2 or 4. (or 6 with upgraded gearboxes)
- Wheels used is 6. (or 4 if designed for meccanum)
- Can use any size of wheel if chains are used. Can use 6"+ if belts are used.
- Can use any type of wheel. (2 normal and 2 omni used for rotational axis change)
- Can use chain or belts with sprockets or pullies respectively.

1.2 Programming

- Can be done by freshmen with some help.
- Extremely straightforeward.
- Short development time.
- Easiest to program for autonomous. Dead reckoning or encoder reading possible.
- No need for a relative heading from a gyro.

1.3 Capabilities

- Tied for best design for pushing in the direction of the wheels.
- Resists perpendicular pushing perfectly.
- Resists $Sin(|\Theta|)$ in general where Θ is the angle of the pushing force from perpendicular.
- Can rotate about anywhere on the axis of the center wheels.
- Single best option for traction reliability. (track optional to help traction if needed)
- One of the best options for max speed/acceleration as all motors are alligned.
- More compact and leaves more space for interior components.

1.4 Hinderances

- Must rotate to move in another direction.

1.5 Driver Usage

- Easiest to learn.
- No orientation required, no gyro/accelerometer drift.
- Easy to build and most likely part of a practive bot so easy to practice on.
- Easiest control scheme, only needs 2 axis to control on seperate hands leading to less human error.
- More likely to have experienced drivers.

1.6 Design and Build

- Easiest to design.
- Easiest to build.

1.7 Assembly and Repair

- Easiest to assemble if designed well.
- Easy but annoying to repair with chain and sprockets.
- Either easy or very hard to repair with belts. (spares required)

2 Mecanum Drive

2.1 Cost

- Needs 4 gearboxes.
- Motors used is 4.
- Wheels used is 4.
- Can use any size of mecanum wheel if chains are used.

- Needs a full matching set to function, with 2 of each chirality.
- Can use chain or belts with sprockets or pullies respectively.

2.2 Programming

- Can be done by freshmen with a some help to a lot of help.
- Programming requires knowlege of how wheels are oriented.
- Medium development time.
- Hardest to program for autonomous. Dead reckoning isn't even a guarantee.
- Can do without a gyro, but one is often used and is the source of problems.

2.3 Capabilities

- Not great for pushing in the direction of the wheels, but it can do it.
- Resists roughly half of all pushing in any direction.
- Rotations possible, but not reliable due to rollers.
- Some conditional traction reliability.
- Ideal for very small perpendicular to wheel motions.
- More compact and leaves more space for interior components.

2.4 Hinderances

- Easy for most robots to push around.
- Hard to operate on uneven or slick terrain, with potential to get stuck.
- Wheels are prone to heavy wear and tear and flat spots causing sliding or skipping.

2.5 Driver Usage

- Easy to learn. Very hard to master.
- Possible orientation corrections based on presence of gyro/accelerometer and its drift.
- Relatively easy to build but there may or may not be a practice bot ready.
- Harder control scheme, often uses at least 3 axis on the same joystick where human error is likely.
- Moderately likely to have somewhat experienced drivers, but very unlikely to have mastery.

2.6 Design and Build

- Relatively simple to design.
- Needs space for extra gearboxes and power transmission.
- Easy to build.

2.7 Assembly and Repair

- Easy to assemble but requires thought of wheel orientations.
- Somewhat harder to repair.
- Full set of spares required at all times.

3 Omniwheel Configurations

3.1 Cost

- Needs 2+ gearboxes depending on style, often 3 or 4.
- Motors used is often the same as the number of gearboxes, except drift drive or H-drive where the robot is mainly going to be driving foreward.
- Wheels used is 3+, usually the same as the number of gearboxes.
- Can use any size of omni wheel if chains are used.
- Can use chain or belts with sprockets or pullies respectively.

3.2 Programming

- Drift drive, H-drive, and orthogonal holonomic drive can be done by freshmen with a some help.
- Diagonal holonomic, and kiwi drive can be done by freshmen with moderate help.
- Programming drift drive, H-drive, and orthogonal holonomic drive requires no knowlege of how wheels are oriented.
- Programming diagonal holonomic, and kiwi drive requires knowlege of how wheels are oriented and some basic trigonometry or vector math.
- Low to medium development time.
- H-drive and orthogonal holonomic drive are the easiest to program for autonomous of all setups.
- Drift drive, diagonal holonomic and kiwi drive are dead reckoning viable if only driving straight or else they need sensors in autonomous.
- Can do without a gyro, but one is often used and is the source of problems.

3.3 Capabilities

- Drift drive and H-drive great for pushing in the direction of the wheels, holonomic and kiwi aren't good for pushing.
- Resists no pushing in any direction.
- Rotations are flexible and reliable with little turning scrub.
- Some conditional traction reliability.
- Drift drive is perfect for pure speed and evasion, and is usually used for pushing/herding spherical objects as the robots momentum carries the robot along with the spheres momentum.
- H-drive is the faster cousing of mecanum and is used for regular robots who need fast sideways movement at the expense of some pushing resistance, but gains pushing power in all directions.

- Holonomic is used if complete mobility is needed, and pushing power is not as critical, or weight needs to be reduced from H-drive.
- Drift and H-drive can be swapped out for tank as needed.

3.4 Hinderances

- Trivial for most robots to push around, unless actively pushing back.
- Holonomic and kiwi drive are Hard to operate on uneven or slick terrain, with potential to get stuck.
- Wheels are prone to moderate wear and tear.

3.5 Driver Usage

- Drift is easy to learn and to be good. Mastery would take a lot of time and practice, as capabilities are potentially endless.
- H-drive is easy to learn and be great. Mastery would take competition experience.
- Orthogonal holonomic is easy to learn and master.
- Diagonal holonomic and kiwi are easy to learn, moderate to be good, hard to be great, and mastery is a pipe dream.
- Possible orientation corrections based on presence of gyro/accelerometer and its drift for diagonal holonomic and kiwi.
- Any tank and drift bot are fully interchangable so there will always be a bot potentially ready in 1hr.
- H-drive could take a modified tank and would require design.
- Holonomic and kiwi will not have a bot available most likely.
- Drift and H-drive can use tank style drive, H-drive would need some modification for sideways movement.
- H-drive, holonomic and kiwi can use one 3 axis joystick, with or without gyro.
- Likely no experienced drivers.

3.6 Design and Build

- Drift is the same design/build as tank, just different wheels.
- H-drive has extra space and weight requirements, but could use current robot frame styles.
- Holonomic and kiwi would require new robot frame styles to be made.
-
- Needs space for extra gearboxes and power transmission.
- Varying difficullty to build. Ranked from easy to medium: drift, kiwi, holonomic, then H-drive.

3.7 Assembly and Repair

- Assembyly may or may not be tight on space depending on the design.
- Somewhat harder to repair.
- One spare required at all times, possibly two depending on level of pushing going on.

4 Crab and Swerve Drive

4.1 Cost

- Needs 5 or 8 gearboxes respectively usually.
- Motors used is 5 or 8 respectively, where 0 or 4 can be intermitent respectively usually.
- Wheels used is 4 usually.
- Wheels need to be relatively small due to weight and space concerns.
- Each wheel needs lots of extra hardware to function properly.
- Must use gears for wheel power,but can use chain or pullies for crab steering, where swerve requires gears for steering.

4.2 Programming

- Can be done by seniors with a lot of help for crab, most likely swerve would be written by mentors or professional libraries.
- Programming requires knowlege of trigonometry, and matrix math for swerve.
- Extremely long development time, often longer than a full season for swerve.
- Surprisingly easy to program for autonomous. Dead reckoning is almost guaranteed.
- Needs tons of encoders and sensors, and is often used with a gyro.
- Very large risk of programming bugs, and possible physical damage from bad code.

4.3 Capabilities

- The most efficient way to push in any direction.
- Resists all pushing in any direction with driver skill with crab, resists all pushing with no driver skill in swerve.
- Rotation hard to program but very doable with swerve.
- Complete traction reliability.
- Ideal for very movement heavy games where pushing, speed, and evasion are central.

4.4 Hinderances

- For crab, rotation not possible except by running into things which could cause problems if orientation is important.
- Gyro can cause major problems if it is wrong.

4.5 Driver Usage

- Easy to learn and be great at but hard to master with crab, easy to learn and master with swerve.
- Possible orientation corrections based on presence of gyro/accelerometer and its drift.
- Most likely there will not be a practice bot ready.

- Harder control scheme, often uses at least 3 axis on the same joystick where human error is likely, but is intuitive.
- Unlikely to have experienced drivers, but it isn't a problem as it is intuitive.

4.6 Design and Build

- Incredibly hard to design, but expensive premade modules are available for purchase.
- Needs a lot of extra space and weight allowance for extra gearboxes and power transmission.
- Hard to build, especially swerve.

4.7 Assembly and Repair

- A nightmare to assemble and requires special tools most likely.
- Near impossible to repair in a reasonable amount of time.
- Spares required at all times for basically the entire drive base.