Contents

1 Abbreviazioni

1.1 C

CV Combustion Vehicle (Internal Combustion Engine Vehicle)

1.2 D

DC Driverless Cup

DV Driverless

DNF Did Not Finish

DQ Disqualified

1.3 E

EV Electric Vehicle

2 Normale

2.1 La formula per

2.1.1 skidpad

cambia da manual a driverless

1. Driverless

D4.5.2 Runs with a run time without penalties >25 s will be DQ.

D4.5.3 If a team that finishes at least one DV run without a DNF or DQ, points based on the following formula are given:

$$DV\_SKIDPAD\_SCORE = P_{max}\frac{N_{all}+1-R_{DV,team}}{N_{all}}$$

with

P_{max} is the maximum points for the event according to table 3

$R_{DV,team}$ is the ranking of team's best DV time including penalties within the best driverless times including penalties of all other teams

N_{all} is the number of teams who finished at least one manual or DV run without DNF or DQ

visto che non so le domande vanno a tempi o rank le ho implementate entrambe

```
def our_rank(ours:float, others:list[float]) -> int:
    """
            DOVREBBE dare il rank dati il nostro tempo e i tempi
        degli altri team
            non ho idea di come dovrei controllare per penalità con
        st'api del cazzo
            dipende da come ci tira con le domande
            almeno credo che venga definito così il rank, non da
        altre spec nel 4.5
            e non so se voglio andare a cercare roba OLTRE il 4.5
    """
    if ours > 25.0:
        print("comunque squalificato che ci mai messo più di 25
        ↪  secondi")

    bigger_than:int = 0
    for i in others:
        if ours > others:
```

```
            bigger_than += 1

    int before_us = len(others) - bigger_than
    return before_us + 1

def fs_dv_skidpad_score_rank(nteams:int, rank:int) -> float:
    pmax = fs_max_scores['skidpad']
    return pmax * ((nteams + 1 - rank) / nteams)

def fs_dv_skidpad_score_times(ours:float, others:list[float]) ->
↪   float:
    if ours > 25.0:
        print("squalificato perchè più di 25 secondi")
        return 0

    return fs_div_skidpad_score_rank(1 + len(others), our_rank(ours,
    ↪   others))
```

2. Driverless Cup

D4.6.3 5 % of the maximum points for the event according to table 3 are awarded to every team that finishes at least one run without DNF or DQ.

D4.6.4 If a team's run time including penalties is below T_{max}, additional points based on the following formula are given:

$$DC\_SKIDPAD\_SCORE = 0.95\, P_{max} \left(\frac{\left(\frac{T_{max}}{T_{team}}\right)^2 - 1}{1.25} \right)$$

P_{max} is the maximum points for the event according to table 3.
T_{team} is the team's best DV run time including penalties.
T_{max} is 1.5 times the time of the fastest DV run of all teams including penalties.

in python

```
def fs_dc_skidpada_score(fastest:float, team:float, dq:bool=False)
↪   -> float:
    pmax = 75 # dalla tabella 3
    tmax = 1.5 * fastest
    score = 0.95 * pmax * (((tmax / team)**2)-1)/1.25
    if not dq:
        score += (5/100) * pmax

    return score
```

2.1.2 accelleration

1. Driverless

D5.5 Driverless Acceleration Scoring

D5.5.1 Runs with a time without penalties >25 s will be DQ.

D5.5.2 If a team that finishes at least one DV run without a DNF or DQ, points based on the following formula are given:

$$DV\_ACCELERATION\_SCORE = P_{\max} \frac{N_{\text{all}} + 1 - R_{\text{DV,team}}}{N_{\text{all}}}$$

with

$P_{\max}$ is the maximum points for the event according to table 3

$R_{\text{DV,team}}$ is the ranking of team's best DV time including penalties within the best DV times including penalties of all other teams

N_{all} is the number of teams who finished at least one manual or DV run without DNF or DQ

culo che è lo stesso di skidpad

```
def fs_dv_accelleration_score_rank(nteams:int, rank:int) -> float:
    return fs_dv_skidpad_score_rank(nteams, rank)

def fs_dv_accelleration_score_times(ours:float, others:list[float])
↪   -> float:
    return fs_dv_skidpad_score_times(ours:, others)
```

2.1.3 autocross

autocross driverless non fa parte della cup normale, quindi qui basta una funzione. Purtroppo è un po'...

D6.5 [DC ONLY] Driverless Autocross Scoring

D6.5.1 10 % of the maximum points for the event according to table 3 are awarded to every team that finishes at least one run without DNF or DQ.

D6.5.2 If a team's $T_{team,total}$ is below $T_{\max}$, points based on the following formula are given:

$$AUTOCROSS\_SCORE = 0.9\, P_{\max} \left(\frac{T_{max} - T_{team,total}}{T_{max} - T_{min}} \right)$$

$$T_{team,total} = \min(T_{team,1}, avg\,(T_{team,1}, T_{team,2}))$$

P_{max} is the maximum points for the event according to table 3.
$T_{team,1}$ is the team's time including penalties of run 1.
$T_{team,2}$ is the team's time including penalties of run 2.
T_{max} is the time for driving the lap with 6 m/s.
T_{min} is the fastest $T_{team,total}$ of all teams.
For runs that are DNF or DQ and for runs with a $T_{team,i}$ above T_{max}, $T_{team,i}$ is set to T_{max}.

io non ho idea di come cazzo fare questo e non ho trovato una domanda che lo riguardasse per vedere come farlo

```python
def team_total(one:float, two:float) -> float:
    return min(one, (one+two)/2)

def fs_autocross_score(fastest:flaot, team_one:float, team_two:float,
↪  fast:float, dq:bool = False) -> float:
    pmax = max_scores['autocross']
    score = 0.9 * pmax ((fast - team_total(one, two)) / (fast - fastest))
    if not dq:
        score += 0.1 * pmax
    return score

def fs_autocross_score_times(team_one:float, team_two:float, super6:float,
↪  others:list[tuple[float, float]]) -> float:
    fastest:float = min(team_total(x[0], x[1]) for x in others)
    return fs_autocross_score(fastest, team_one, team_two, super6)
```

2.1.4 track drive

normale

D8.4 Trackdrive Scoring

D8.4.1 Each lap of the trackdrive event is individually timed. The corrected elapsed time is determined by adding any penalty times.

D8.4.2 If a team's corrected elapsed time is below T_{max} and the run was not DNF or DQ, points based on the following formula are given:

$$TRACKDRIVE\_SCORE = 0.75\, P_{max} \left(\frac{T_{max}}{T_{team}} - 1 \right)$$

P_{max} is the maximum points for the event according to table 3.
T_{team} is the team's corrected elapsed time.
T_{max} is 2 times of the corrected elapsed time of the fastest vehicle.

D8.4.3 An additional 2.5 % of the maximum points for the event according to table 3 are awarded for every completed lap, independent of the corrected elapsed time. This is also applied for teams that do not finish the trackdrive i.e. get a DNF.

```python
def fs_trackdrive_scoring(fastest:float, team:float, dq:bool=False) ->
↪  float:
    pmax:float = max_scores['trackdrive']
    tmax:float = 2*fastest
    score = 0.75 * pmax * ((tmax/team) - 1)
    if not dq:
        score += (2.5/100) * pmax
    return score
```

2.2 E quindi serve

2.2.1 D 4.4.1

2.2.2 Tabella 3

	CV & EV	DC
Static Events:		
Business Plan Presentation	75 points	-
Cost and Manufacturing	100 points	-
Engineering Design	150 points	150 points
Dynamic Events:		
Skidpad	50 points	-
Driverless (DV) Skidpad	75 points	75 points
Acceleration	50 points	-
Driverless (DV) Acceleration	75 points	75 points
Autocross	100 points	-
Driverless (DV) Autocross	-	100 points
Endurance	250 points	-
Efficiency	75 points	-
Trackdrive	-	200 points
Overall	1000 points	600 points

Table 3: Maximum points awarded

```
# se proprio vogliamo esagerare
# hanno ci sono mai max_score diversi a seconda della cup dc o meno
fs_max_scores = {
    'skidpad'       : 75,
    'accelleration' : 75,
    'autocross'     : 100,
    'trackdrive'    : 200,
}
```

3 Germania

uguale

4 East

5 Portogallo

uguale

6 Repubblica Ceca

uguale