```
──────────────────── MODULE blablamovie ────────────────────
Users vote for a 1 . . 3 Movies during one week.  After the week, the most voted Movie is displayed.

EXTENDS Integers, FiniteSets

PT ≜ INSTANCE PT

CONSTANTS MOVIES, VOTE_DURATION, USERS
ASSUME MOVIES ⊆ Int
ASSUME ∀ m ∈ MOVIES : m ≥ 0
ASSUME USERS ⊆ Int
ASSUME VOTE_DURATION ∈ Int
ASSUME VOTE_DURATION > 0

MoviesVotes ≜
    [MOVIES → SUBSET USERS]

   --algorithm ElectionOfMovie

variables
    days_left ∈ 1 . . VOTE_DURATION,
    movies_votes = [m ∈ MOVIES ↦ {}],
    winner = −1;

define
    TotalVoted ≜
        PT ! ReduceSet(
             LAMBDA m, acc : acc + Cardinality(movies_votes[m]),
             DOMAIN movies_votes, 0)

    TotalVotedOf(movie) ≜
        Cardinality(movies_votes[movie])

    HasMovieToVote(user) ≜
        ∃ m ∈ MOVIES : user ∉ movies_votes[m]

    ChooseNotVotedMovies(n, user) ≜
        PT ! ReduceSet(
             LAMBDA i, acc : IF HasMovieToVote(user)
                  THEN acc ∪ {
                     CHOOSE m ∈ MOVIES :
                         m ∉ acc ∧ user ∉ movies_votes[m]}
                  ELSE acc,
             1 . . n, {})

    NumberOfMoviesToVote(user) ≜
        Cardinality(PT ! ReduceSet(
           LAMBDA m, acc : IF user ∉ movies_votes[m]
```

```
                  THEN  $acc \cup \{m\}$
                  ELSE  $acc$,
             $MOVIES$, $\{\}$))

   $Winner \triangleq$ CHOOSE $m \in MOVIES : \forall m2 \in (MOVIES \setminus \{m\}) :$
        $TotalVotedOf(m) \geq TotalVotedOf(m2)$
end define ;

fair process $User \in USERS$
variable $votesLeft = 3$ ;
begin $Loop$:
   while $days\_left > 0 \land votesLeft > 0$
    do
      with $numberOfVotes \in 1 \mathrel{..} PT!Min(NumberOfMoviesToVote(self), votesLeft)$,
           $moviesToVote = ChooseNotVotedMovies(numberOfVotes, self)$
       do
         $movies\_votes := PT!ReduceSet($
            LAMBDA $m, acc : [acc$ EXCEPT
               $![m] = acc[m] \cup \{self\}],$
            $moviesToVote, movies\_votes)$ ;
         $votesLeft := votesLeft - numberOfVotes$ ;
      end with ;

      $days\_left := days\_left - 1$ ;
   end while ;

   $DisplayWinner$:
      $winner := Winner$ ;
end process ;
end algorithm   ;
 BEGIN TRANSLATION ($chksum(pcal) =$ "$7294aa2a$" $\land$ $chksum(tla) =$ "$dbee08ab$")
VARIABLES $days\_left, movies\_votes, winner, pc$

 define statement
$TotalVoted \triangleq$
   $PT!ReduceSet($
        LAMBDA $m, acc : acc + Cardinality(movies\_votes[m]),$
        DOMAIN $movies\_votes, 0)$

$TotalVotedOf(movie) \triangleq$
   $Cardinality(movies\_votes[movie])$

$HasMovieToVote(user) \triangleq$
   $\exists m \in MOVIES : user \notin movies\_votes[m]$

$ChooseNotVotedMovies(n, user) \triangleq$
   $PT!ReduceSet($
        LAMBDA $i, acc :$ IF $HasMovieToVote(user)$
```

```
                THEN  acc ∪ {
                    CHOOSE  m ∈ MOVIES :
                         m ∉ acc ∧ user ∉ movies_votes[m]}
                ELSE  acc,
          1 . . n, {})

NumberOfMoviesToVote(user) ≜
     Cardinality(PT ! ReduceSet(
          LAMBDA  m, acc : IF  user ∉ movies_votes[m]
               THEN  acc ∪ {m}
               ELSE  acc,
          MOVIES, {}))

Winner ≜ CHOOSE  m ∈ MOVIES : ∀ m2 ∈ (MOVIES \ {m}) :
     TotalVotedOf(m) ≥ TotalVotedOf(m2)

VARIABLE  votesLeft

vars ≜ ⟨days_left, movies_votes, winner, pc, votesLeft⟩

ProcSet ≜ (USERS)

Init ≜  Global variables
         ∧ days_left ∈ 1 . . VOTE_DURATION
         ∧ movies_votes = [m ∈ MOVIES ↦ {}]
         ∧ winner = −1
          Process User
         ∧ votesLeft = [self ∈ USERS ↦ 3]
         ∧ pc = [self ∈ ProcSet ↦ "Loop"]

Loop(self) ≜  ∧ pc[self] = "Loop"
              ∧ IF  days_left > 0 ∧ votesLeft[self] > 0
                    THEN  ∧ ∃ numberOfVotes ∈ 1 . . PT ! Min(NumberOfMoviesToVote(self), votesLeft[self])
                               LET  moviesToVote ≜ ChooseNotVotedMovies(numberOfVotes, self)IN
                                 ∧ movies_votes′ =              PT ! ReduceSet(
                                                        LAMBDA  m, acc : [acc EXCEPT
                                                             ! [m] = acc[m] ∪ {self}],
                                                        moviesToVote, movies_votes)
                                 ∧ votesLeft′ = [votesLeft EXCEPT ! [self] = votesLeft[self] − numberOfVotes]
                          ∧ days_left′ = days_left − 1
                          ∧ pc′ = [pc EXCEPT ! [self] = "Loop"]
                    ELSE  ∧ pc′ = [pc EXCEPT ! [self] = "DisplayWinner"]
                          ∧ UNCHANGED ⟨days_left, movies_votes, votesLeft⟩
              ∧ UNCHANGED  winner

DisplayWinner(self) ≜  ∧ pc[self] = "DisplayWinner"
                       ∧ winner′ = Winner
```

$$
\begin{aligned}
&\land\ pc' = [pc \ \text{EXCEPT}\ ![self] = \text{“Done”}]\\
&\land\ \text{UNCHANGED}\ \langle days\_left,\ movies\_votes,\ votesLeft \rangle
\end{aligned}
$$

$$User(self) \triangleq Loop(self) \lor DisplayWinner(self)$$

Allow infinite stuttering to prevent deadlock on termination.

$$
\begin{aligned}
Terminating \triangleq\ &\land \forall\, self \in ProcSet : pc[self] = \text{“Done”}\\
&\land\ \text{UNCHANGED}\ vars
\end{aligned}
$$

$$
\begin{aligned}
Next \triangleq\ &(\exists\, self \in USERS : User(self))\\
&\quad \lor Terminating
\end{aligned}
$$

$$
\begin{aligned}
Spec \triangleq\ &\land\ Init \land \Box[Next]_{vars}\\
&\land \forall\, self \in USERS : \mathrm{WF}_{vars}(User(self))
\end{aligned}
$$

$$Termination \triangleq \Diamond(\forall\, self \in ProcSet : pc[self] = \text{“Done”})$$

END TRANSLATION

$$
\begin{aligned}
&SumVotesLeft \triangleq\\
&\quad PT!ReduceSeq(\\
&\qquad \text{LAMBDA}\ i,\ acc : acc + i,\\
&\qquad votesLeft,\ 0)
\end{aligned}
$$

$$
\begin{aligned}
&TypeOK \triangleq\\
&\quad \land \quad movies\_votes \in MoviesVotes\\
&\quad \land \quad days\_left \in 0 \,..\, VOTE\_DURATION\\
&\quad \land \quad \forall\, u \in USERS : votesLeft[u] \in 0 \,..\, 3\\
&\quad \land \quad winner = -1 \lor winner \in MOVIES
\end{aligned}
$$

$$
\begin{aligned}
&Invariants \triangleq\\
&\quad \land\ TotalVoted = (Cardinality(USERS) * 3) - SumVotesLeft
\end{aligned}
$$

$$
\begin{aligned}
&WinnerIsFound \triangleq\\
&\quad \Diamond(winner \in MOVIES)
\end{aligned}
$$

\ * Modification History

\ * Last modified *Thu Mar* 25 10:45:43 *CET* 2021 by *davd*

\ * Created *Wed Mar* 24 12:26:25 *CET* 2021 by *davd*