

This is the sprint log that we, as a team, came up with as a realistic and feasible amount of work to do within a 2 week sprint. This decision was made with consideration of our previous project score and the confidence in our own C++ programming abilities. For approximation purposes, we decided that a small PBI was 1-2 hrs of work, a medium was 3-4 and a large was 5-8 hrs. A sprint log with all the PBI's that the product owner wishes to achieve during this sprint would be approximately equal to 96 hours of work. With our 4 team members, we expect to be able to do 64 hrs of work during the sprint. Therefore, below are all the PBI's that we believe we are able to complete within this 2 week sprint.

1. Run IR

As an election official, I want to find the majority vote So that I can run an Instant Runoff Election.
Acceptance Criteria: Election official is able to run a majority vote election with multiple candidates and one clear and definitive winner.
Definitions of Done: Majority vote election algorithm is functional and produces one fair winner. Election is properly logged on an audit file. Unit and system testing is successful.
Effort: Medium

As an election official, I want to eliminate the candidate with the least votes in an IRV if there is no majority winner So that I can still find a winner if there is not a clear majority.
Acceptance Criteria: Election officials are able to run a IRV election with multiple candidates, where the lowest candidate is removed if there is no majority until one clear and definitive winner is determined.
Definitions of Done: IRV election algorithm is functional and produces one fair winner. Election is properly logged on an audit file. Unit and system testing is successful.
Effort: Medium

As an election official, I want to be able to keep track of the information from each ballot So that I can easily keep track of each vote.
Acceptance Criteria: All information from each ballot is stored within the program, separately from the input file.
Definitions of Done: Ballot information is stored. Unit and system testing is successful.

Effort: Medium

As an election official, I want to be able to keep track of the information for each IRV ballot So that I can easily keep track of the candidate rankings and current votes.
--

Acceptance Criteria: All information from each IRV ballot is stored within the program, separately from the input file.

Definitions of Done: Ballot information is stored. Unit and system testing is successful.

Effort: Medium

As an election official, I want to be able to keep track of the information for each candidate or party being voted for So that I can know each party or candidate voted for and the votes for each.
--

Acceptance Criteria: All information from each votable item is stored within the program, separately from the input file.

Definitions of Done: Votable item information is stored. Unit and system testing is successful.

Effort: Medium

As an election official, I want to be able to keep track of the information for each candidate So that I can easily get all the candidate information.
--

Acceptance Criteria: All information from each candidate is stored within the program, separately from the input file.
--

Definitions of Done: Candidate information is stored. Unit and system testing is successful.
--

Effort: Medium

2. Run CPL

As an election official, I want to be able to determine which party in Cpl has the most votes
--

So that a party winner can be determined in a Cpl election
Acceptance Criteria: The party with the most votes is determined to be the winning party.
Definitions of Done: runCpl function works up until determineSeatAllocation.
Effort: Medium

As an election official, I want to be able to determine how many seats are allocated to each party in a Cpl election, So that I can determine which party members won a seat.
Acceptance Criteria: Seats are distributed by a quota described by the cpl protocol.
Definitions of Done: runCpl function works up until distributeRemainderSeats
Effort: Medium

As an election official, I want to be able to use the largest remainder algorithm to distribute any of the winning party's extra seats. So that leftover seats are distributed fairly.
Acceptance Criteria: All seats are distributed to the appropriate parties.
Definitions of Done: runCpl function works properly and is tested.
Effort: Medium

As an election official, I want to be able to keep track of the information for each CPL ballot So that I can easily keep track of each ballot's rankings and current vote
Acceptance Criteria: All information from each CPL ballot is stored within the program, separately from the input file.
Definitions of Done: Ballot information is stored. Unit and system testing is successful.
Effort: Medium

As an election official,
I want to be able to keep track of the information for each party
So that I can easily get all the party information.

Acceptance Criteria: All information from each party is stored within the program, separately from the input file.

Definitions of Done: Party information is stored. Unit and system testing is successful.

Effort: Medium

As an election official,
I want to be able to determine which party in Cpl has the most votes
So that a party winner can be determined in a Cpl election

Acceptance Criteria: The party with the most votes is determined to be the winning party.

Definitions of Done: runCpl function works up until determineSeatAllocation.

Effort: Medium

As an election official,
I want to be able to determine how many seats are allocated to each party in a Cpl election,
So that I can determine which party members won a seat.

Acceptance Criteria: Seats are distributed by a quota described by the cpl protocol.

Definitions of Done: runCpl function works up until distributeRemainderSeats

Effort: Medium

As an election official,
I want to be able to use the largest remainder algorithm to distribute any of the winning party's extra seats.
So that leftover seats are distributed fairly.

Acceptance Criteria: All seats are distributed to the appropriate parties.

Definitions of Done: runCpl function works properly and is tested.

Effort: Medium

3. Input File

4. Tie

As an election official, I want to be able to fairly determine a tie between two candidates by a coin toss So that a tie can be determined fairly and randomly.
Acceptance Criteria: Tie breaking can handle two candidates. One candidate is determined as the winner.
Definitions of Done: Tie breaker is functional and produces one fair winner. Unit and system testing is successful.
Effort: Medium

5. Audit

As an election official, I want to have a file tracking the process of the election So that I can review how the election was determined, and the result.
Acceptance Criteria: Audit file is created and can hold the information on how ballots were assigned throughout the election.
Definitions of Done: Audit file is functional and follows the process of the election. Election is properly logged on the file. Unit and system testing is successful.
Effort: Medium