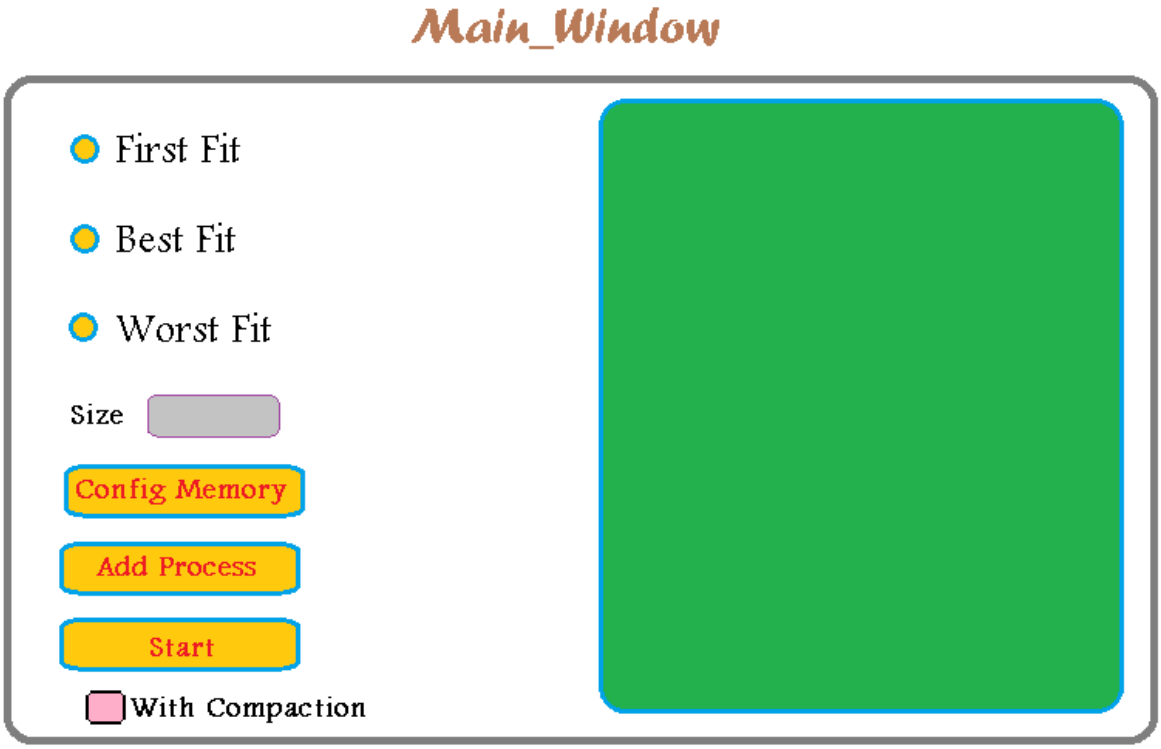
OS Project 2 prototype

When the user first opens the program he see this window:



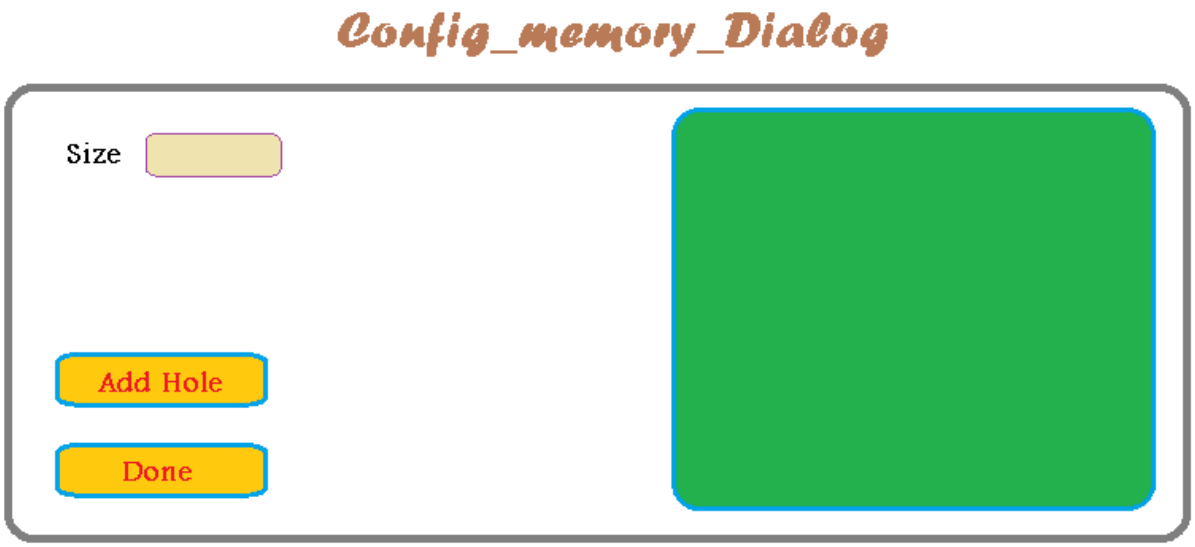
-The user should choose the allocation algorithm from the three radio buttons

-The green box on the left will show the list of segments.

-The size label shows the size of the memory

-The user can if memory compaction is needed by checking “With Compaction”

When the user clicks Config Memory he will see this:



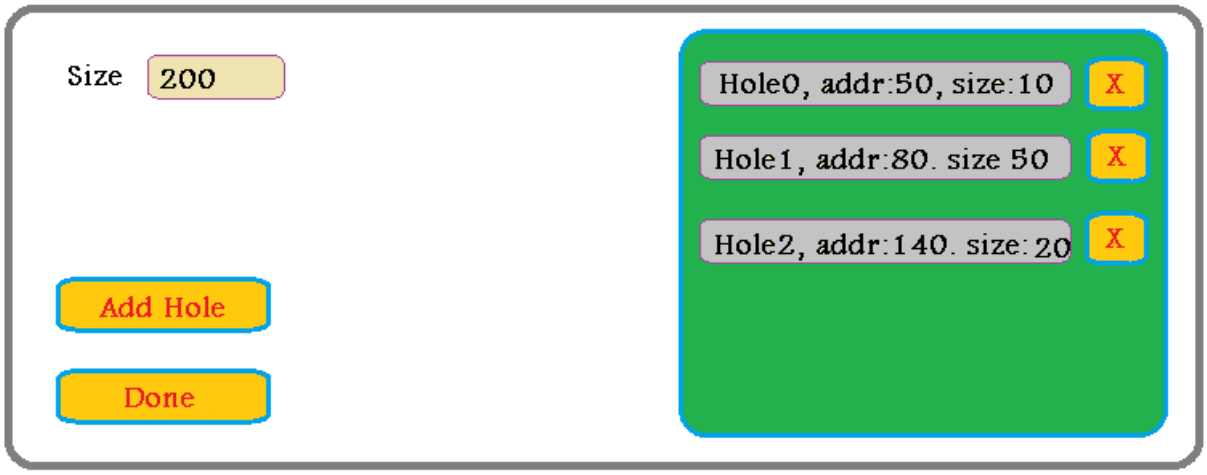
-The user can determine the size of memory.

When the user clicks Add Hole he will see this:

-The user can specify the start address and size of the hole.

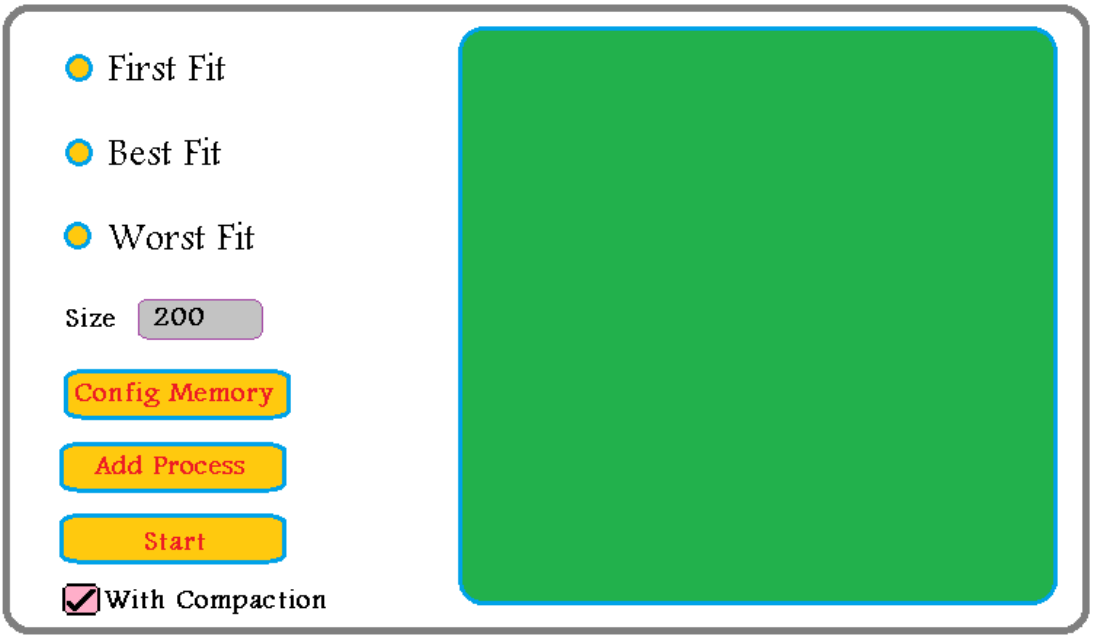
The user will return to Config Memory Dialog after he clicks “Add Hole”.

Config_memory_Dialog Example

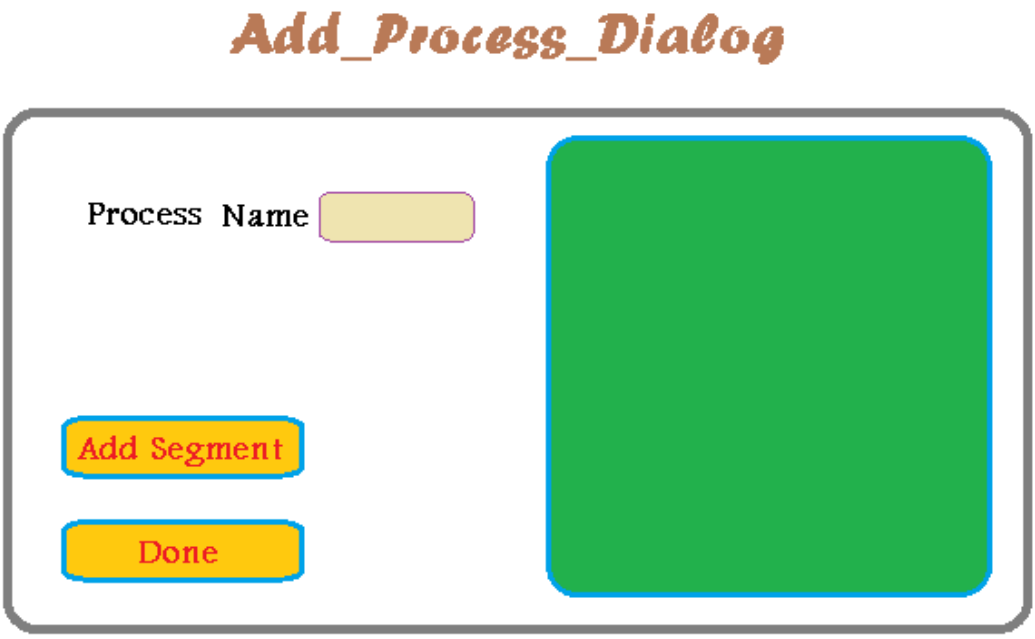


-After the user has configured the memory ,he can return to the Main Window by clicking “Done”.

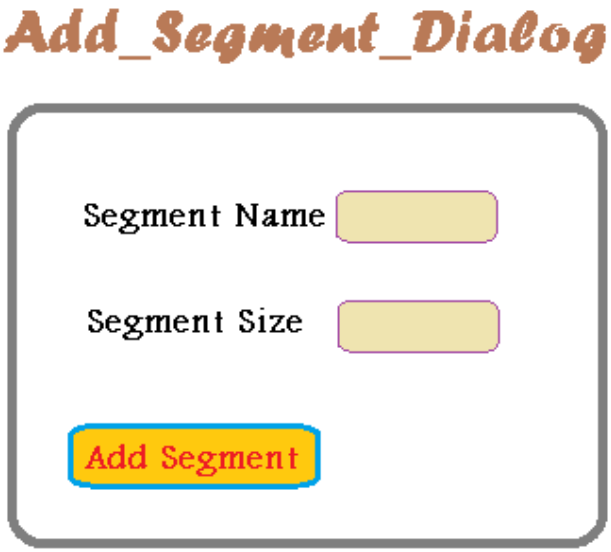
Main_Window Example



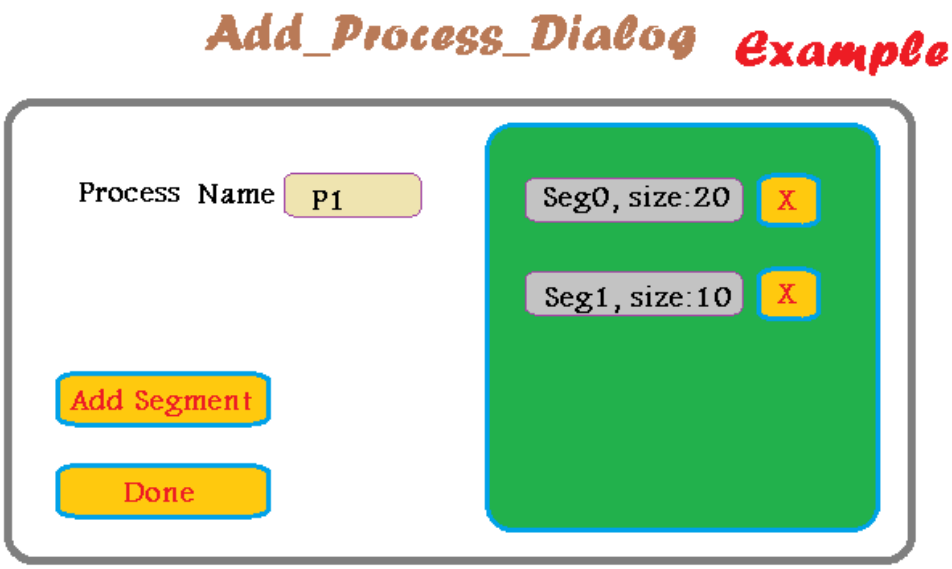
- The user can see that the Size label has changed to the value he had set in the Config Memory Dialog.
- The user can Add Processes by clicking " Add Process" button and he will see this.



- The user can define the process name.
- The user can add segments to the process by clicking "Add Segment".

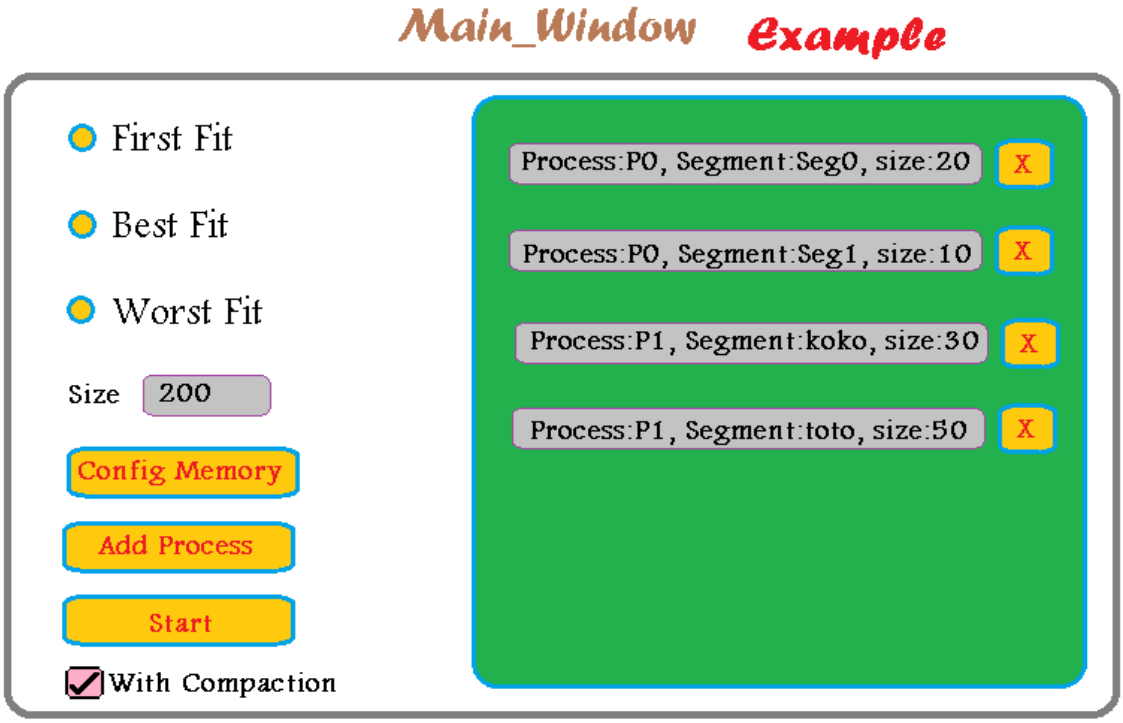


After the user supplies the segment name and size and clicks “Add Segment” he will return to the process dialog.



- The user can delete segments by press the “X” button beside them.

After the user clicks “done” he will be returned to the Main Window.



-The user can see the segments of every process in the green box, he can delete segments by clicking the “X” button beside it.

If any logical error happened(e.g , the user inserted a negative number as the size of memory), the user should be greeted with this error Dialog explaining the error he had done.

OS Project 2 specifications

----------Classes responsible for back end----------

Hole
Segment
Process
Memory

Allocator
- First_Fit
- Best_Fit
- Worst_Fit

----------Classes responsible for front end----------
Gtk.***** <-- choose whatever Gtk class you may like, replace ***** with window / grid / dialog

- Main_Window
- Add_Process_Dialog
- Add_Segment_Dialog
- Add_Hole_Dialog
- New_Memory_Dialog
- Output_Grid
- Error_Dialog

Gtk.Box <-- Gtk container class to hold multiple items (e.g text + button)

- list_element
  - Hole_list_element
  - Segment_list_element
  - Process_list_element

Hole:

- vars:
  - int start_address
  - int size

Segment:

- vars:
  - int segment_index
  - int process_index
  - string name
  - int start_address
  - int size
- functions:
  - __init__(self, String name,int segment_index,int process_index, int start_address, int size)
  - remove_segment()

```
process:
        vars:
                int process_index
                string name
                array Segment
        functions:
                __init__(self, String name)
                __del__(self)
                add_segment(Segment segment)
                remove_segment(int segment_index)

Memory:
        vars:
                int size
                array Process
        functions:
                __init__(self, int size, array hole)
                add_Process(Segment segment)
                remove_Process(int process_index)

Allocator:
        vars:
                array Process
        functions:
                __init__(self, array Process)
                allocate() <---- the children of this class should implement this

First_Fit:
        functions:
                __init__(self, array Process) <--call parent's constructor
                allocate()
Best_Fit:
        functions:
                __init__(self, array Process) <--call parent's constructor
                allocate()

Worst_Fit:
```

functions:

```
__init__(self, array Process) <--call parent's constructor
allocate()
```