

Project Description

Team Members:

- Wasif Ibrar
Student ID: 202149944
- Ripudaman Singh
Student ID: 202054565

How to Load and Run the Project:

1. Download List.zip
2. Extract List.zip
3. Navigate into the folder
4. Double click on List.uproject
5. Click on Play button

Note: The app was made in UE 5.3

Contribution

Ripudaman Singh:

- Implementation of the input validation
- Implementation of delete button and add button.
- Implementation of “Task Complete” checkbox to move items across lists.
- Implementation of the due date functionality

Wasif Ibrar:

- Creation of the widgets (the task widget as well as the Main App Widget that houses all tasks)
- Made the error display functionality.
- Implemented the functionality to add Child Widget to create dynamic list of tasks.

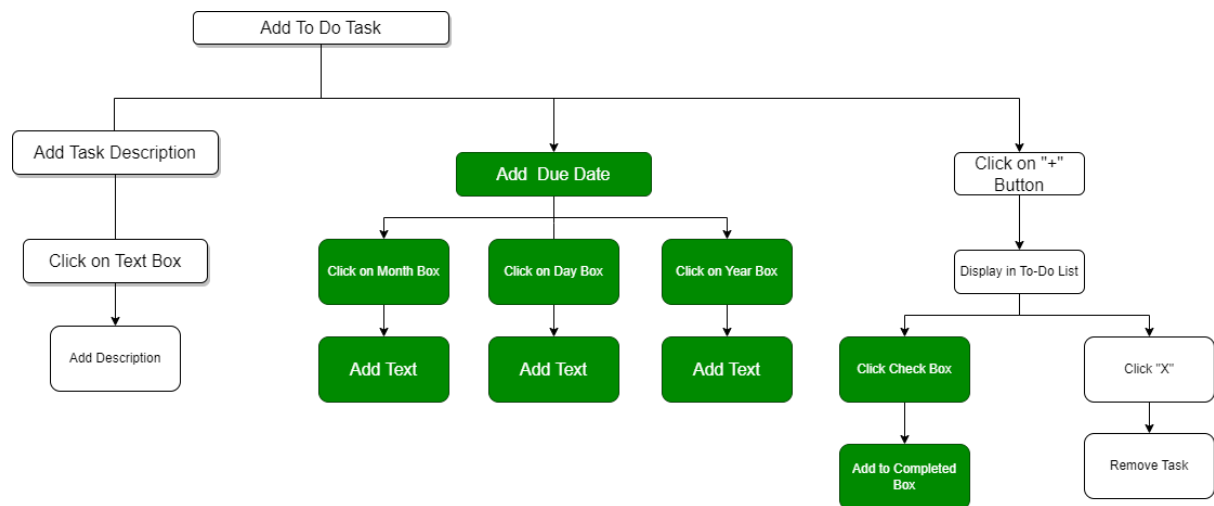
Sources and Attribution

We have developed original code for this project from scratch. However, we referred to the Unreal Engine documentation and community forums for guidance on Unreal UMG programming techniques and the following YouTube video:

<https://www.youtube.com/watch?v=rBTQr6eLWFQ&t=1893s>

Hierarchical Task Analysis

Before implementing the prototype, we performed a hierarchical task analysis (HTA) to identify potential UX improvements. The analysis focused on simplifying the user's task of managing tasks efficiently. As a result, we designed a user interface that streamlines task prioritization, addition, deletion, and deadline tracking. We have outlined the process of adding a task to the list in the following diagram. The green parts are the changes we have proposed to streamline the UX.



UX Principles

Feedback:

We have chosen to improve the principle of feedback by adding error messages. We added input validation for the task description and due date, ensuring that task description is added, date is numeric and date is in the required format.

Affordances:

We have displayed two clearly labelled buttons following conventions such as 'X' used for removing a task and '+' for adding a task. We have also set up a checkbox that moves the task to the completed To-do list and vice-versa.

Signifiers:

We have added hint text in the Add Description Text Box, and for the date we have a signifying the required format. We also added labels for Due Date, "Task Completed" checkbox and for marking the list whether it is in Completed List or To-do list. We added error text for invalid inputs in the text boxes for task description and date.

Mapping:

We structured a layout for each button to ensure user-friendly access. Additionally, we positioned the '+' button adjacent to the task description box, facilitating intuitive usage for users.

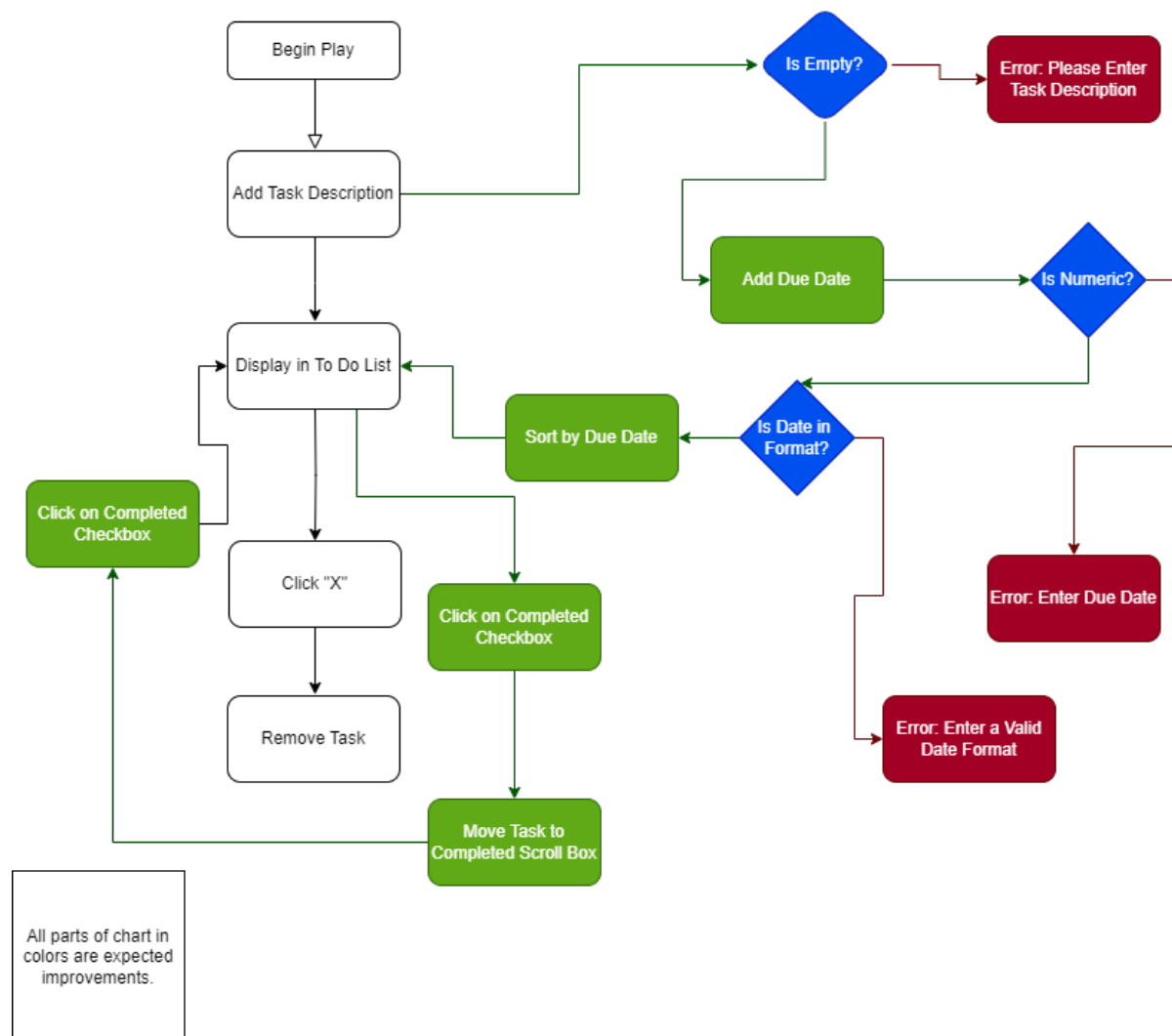
Constraints:

We set up a MM/DD/YY for the date format as a cultural constraint specifically for North American users. Additionally, used red colour for error messages as semantic constraint for it to be clearly visible for the users if they are doing a mistake while adding a task.

UX Improvement Choice

The chosen UX design aspect for focus in the cognitive walkthrough and testing is the implementation of separate completed list. We used it to keep historical record of completed tasks instead of removing them outright.

We have also added a diagram to indicate our choices to improve UX overall.



Cognitive Walkthrough

Date: 26th March 2024

Inspection Conducted By: Ripudaman Singh and Wasif Ibrar

User Task (Primary Goal): Add task to to-do list.

User Profile: Computer science student familiar with UI/UX applications who wants to sort his tasks in a streamlined manner.

Materials: Unreal Engine Installation (5.3), copy of project

Notes: The user was told to just add several tasks to the list along with the due dates. No prior instructions were given.

Walkthrough Steps: User needs to add multiple tasks to the list, each with their individual due dates, descriptions, and completion status. The user can mark a task as complete to move it to the “Completed Tasks” List or remove it altogether from the application.

Question	Observations	UX/UI concerns/suggestions
1. Representation [Yes]	The two lists were clearly labelled, and the buttons used conventions such as “+” for add and “X” for remove	Maybe change the “X” button to a trash can or change its colour to red to signify a potentially dangerous/information losing step.
2. Attention: [Yes]	The user was able to accurately interpret the usage of each UI component with the help of hint texts and labels.	Suggestion was to add a line between each tasks to separate them more clearly.
3. Evaluation [Yes]	The tasks are clearly labelled and sorted into their correct categories, and have their own due dates and remove buttons visible next to them	No suggestions in this category.
4. Goal [Yes]	The user wants to add tasks to the list	Maybe add an option to add a task directly to the “Completed” list for bookkeeping.
5. Intention [Yes]	There are error messages and cursor hover interactions, adding interactivity to the application. The hint texts in the text fields also provide good useability.	Maybe provide more feedback in the form of confirmation messages for tasks such as deletion.
6. Specification [Yes]	There are clearly labelled buttons and text areas with hint texts, making the app very intuitive to use	Maybe enclose the Task Complete and the associated checkbox in their own section to better show that they are linked controls.
7. Execution [Yes]	The task along with its due date and completion status will be added to the list.	Maybe add some sort of tests to ensure functionality is consistent.

Testing

A user test was conducted with an individual subject not associated with the course. We asked the user to enter a task into the to-do list. WE provided no prior instruction other than clicking on the play button to start the application. The tester was able to quickly enter the task in the list due to the provided hint texts in the text fields. The user also mentioned it was pretty clear that the “+” button was for adding. He quickly figured out how to move the task to the “Completed” section as well through the use of the checkbox. The “X” button also seemed to be a good enough indicator for removal of a task from the list. The error messages were also clear enough according to the tester. One possible source of bias is that the tester is a Computer Science student familiar with developing UI/UX applications. The user was able to add a task in roughly 30 seconds without any prior instructions. The addition of the “Completed” section was also appreciated, as the user said they liked keeping a track of past work as motivation. According to the testing, the UX improvements have significantly improved the useability and efficiency of the to-do application.

Overall Conclusions

Based on the HTA, walkthrough, and user test results, the addition of a “Completed” section the application proved to be a successful addition to the TODO App prototype. Further improvements could involve refining the user interface and adding additional features to enhance task organization and prioritization such as implementing sorting based on due date.

Code Effort

We contributed original code to the project, including the implementation of the completed scroll box and due date functionality. We have outlined our specific code contributions in the Project Description section of this document (top of the document). This project was made from scratch, so all code is written by the team. There are only 2 widget files in the project, and one struct to hold all the data for the task.

Screenshots

3300 Assignment

+

Due Date: 11 11 11

To-Do

Completed

Add Task Description

+

Due Date: MM DD YY

To-Do

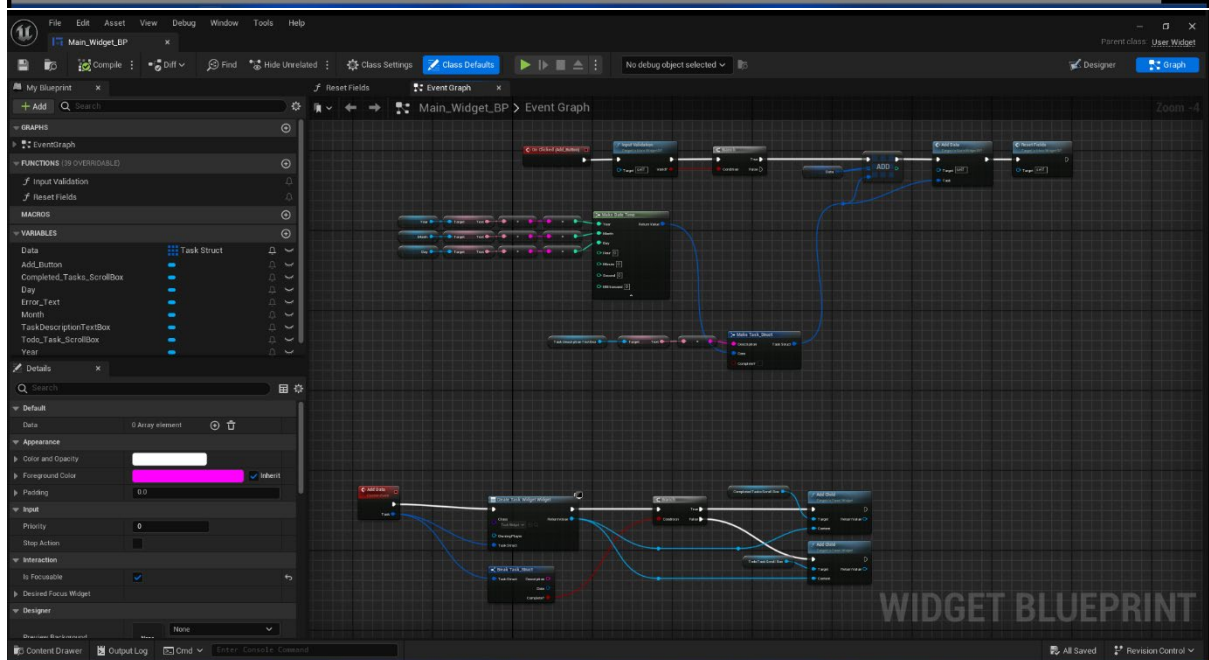
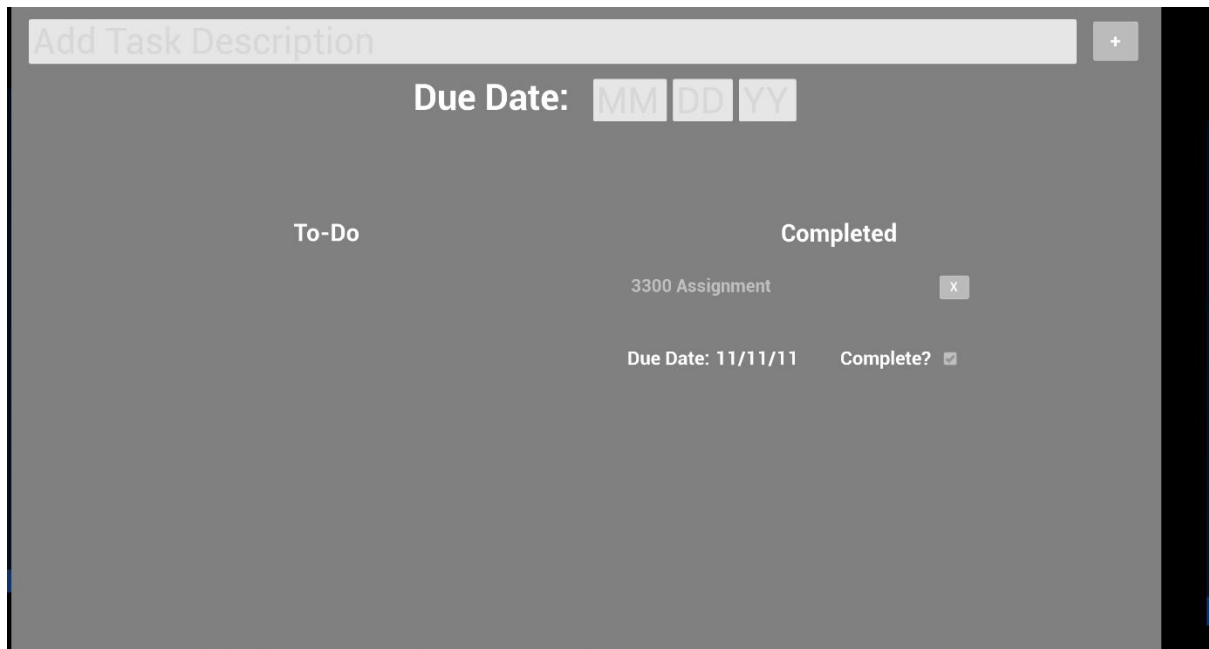
Completed

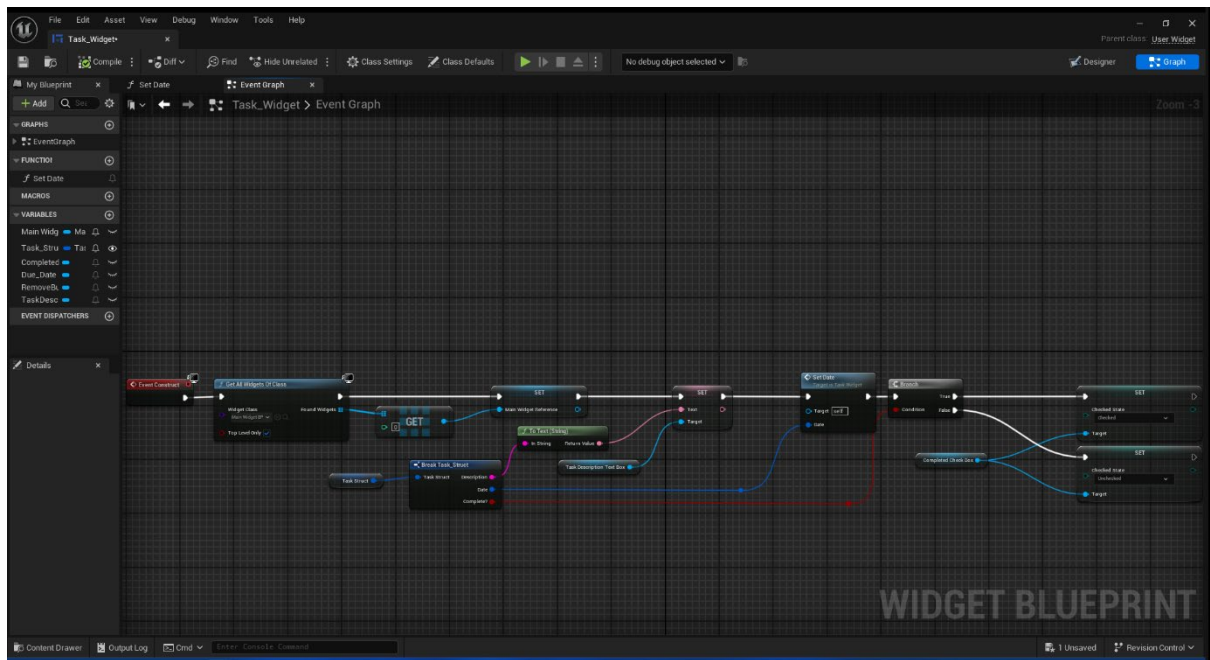
3300 Assignment

x

Due Date: 11/11/11

Complete? ☐





Add Task Description

Due Date: MM DD YY

To-DoCompleted