Task 11 - Spike Summary Report

> 💡 **Spike:** Task_11
> **Title:** Game Graphs from Data
> **Author:** Thomas Horsley, 103071494

Goals & Deliverables

Aim: Implement a game world graph with locations and connections to each other. The data from these worlds should be loaded from a text file and the user can enter various '*go*' commands to move through the locations.

Deliverables:

- Spike Report
- Git Commit History
- Functioning solution

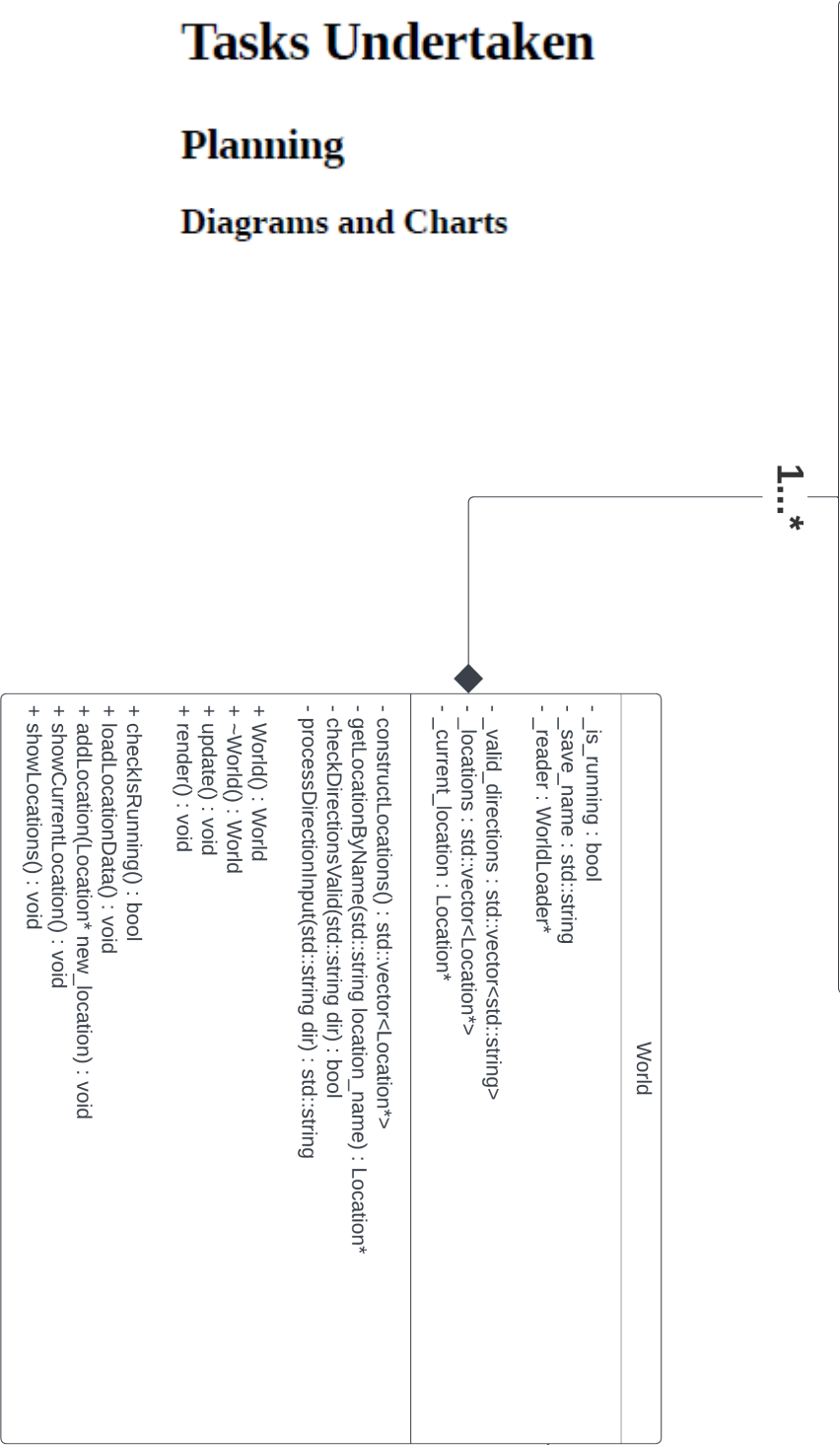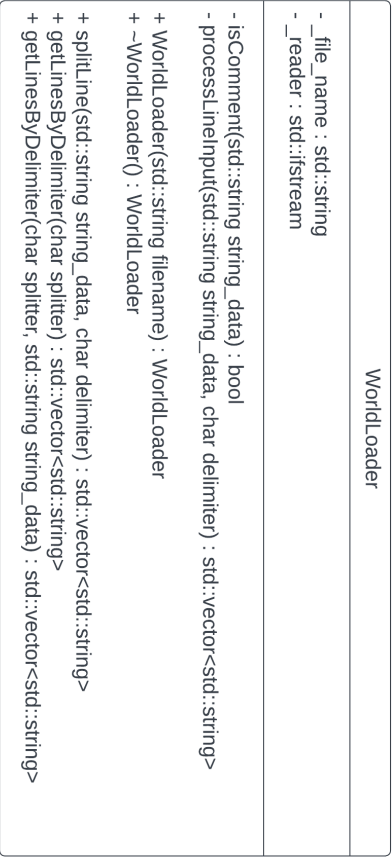## Technology, Tools and Resources
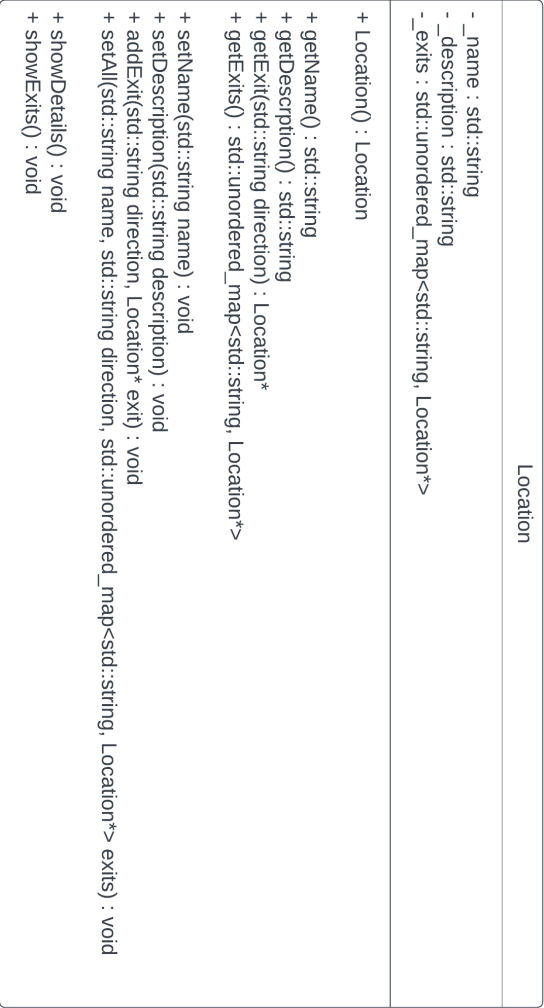
Tech and Tools

> 💡 The project was scripted in C++ 17 using Visual Studio Community 2022.
>
> UML's and charts are made with *www.Lucidchart.com*
>
> Source control is handled using Git.

Resources

- Echo360 Lectures *"Topic 4.1 - Graph Worlds"*
- Maps in C++ — The Cherno https://www.youtube.com/watch?v=KiB0vRi2wlc
- Entirety of stack overflow

Tasks Undertaken

Planning

Diagrams and Charts

Location

- _name : std::string
- _description : std::string
- _exits : std::unordered_map<std::string, Location*>

+ Location() : Location
+ getName() : std::string
+ getDescription() : std::string
+ getExit(std::string direction) : Location*
+ getExits() : std::unordered_map<std::string, Location*>
+ setName(std::string name) : void
+ setDescription(std::string description) : void
+ addExit(std::string direction, Location* exit) : void
+ setAll(std::string name, std::string direction, std::unordered_map<std::string, Location*> exits) : void
+ showDetails() : void
+ showExits() : void

World

- _is_running : bool
- _save_name : std::string
- _reader : WorldLoader*
- _valid_directions : std::vector<std::string>
- _locations : std::vector<Location*>
- _current_location : Location*

+ World() : World
+ ~World() : World
+ update() : void
+ render() : void
- constructLocations() : std::vector<Location*>
- getLocationByName(std::string location_name) : Location*
- checkDirectionsValid(std::string dir) : bool
- processDirectionInput(std::string dir) : std::string
+ checkIsRunning() : bool
+ loadLocationData() : void
+ addLocation(Location* new_location) : void
+ showCurrentLocation() : void
+ showLocations() : void

WorldLoader

- _file_name : std::string
- _reader : std::ifstream

- isComment(std::string string_data) : bool
- processLineInput(std::string string_data, char delimiter) : std::vector<std::string>
+ WorldLoader(std::string filename) : WorldLoader
+ ~WorldLoader() : WorldLoader
+ splitLine(std::string string_data, char delimiter) : std::vector<std::string>
+ getLinesByDelimiter(char splitter) : std::vector<std::string>
+ getLinesByDelimiter(char splitter, std::string string_data) : std::vector<std::string>

1...*

1

Task 11 - Spike Summary Report

Class Descriptions and Notes

The World acts as a interface between the WorldLoader and Location objects in addition to housing functionality allowing for location traversal and user input (though this will be changed when implemented to zorkish) . The World handles post processing and assignment of data provided by the WorldLoader to the Locations, with the WorldLoader being responsible for reading and splitting the text file data.

Each Location houses an unordered map containing a set of directions (the key) and Location pointers (the value). The World will take input from the player, translate this input (if valid) to the requested key and replace it's _current_location with the associated exit location.

Implementation

Git Commit History

```
Commits

  main

  Commits on Sep 2, 2023
      Completed Task 11 - Spike 'Game Data from Graphs'  ...          7412ab1  <>
      unknown committed 1 minute ago

      Refactored the Location _exits vector to an unordered map  ...   3168fc0  <>
      unknown committed 3 hours ago

  Commits on Sep 1, 2023
      Refactored and Split code amongst class files.  ...             dddcf9b  <>
      unknown committed yesterday

  Commits on Aug 31, 2023
      World can now interact with FileReader to instantiate Location objects  ...   2dc5555  <>
      unknown committed 2 days ago

  Commits on Aug 30, 2023
      Implemented file reader class from Task 10 - Lab 'File Input Output'  ...   4f65645  <>
      unknown committed 3 days ago

      Finished basic Location node traversal  ...                     1557169  <>
      unknown committed 3 days ago

      Started Task 11 - Spike 'Game Graphs From Data'  ...            fa253f7  <>
      unknown committed 3 days ago
```

The code for this project has been uploaded to canvas a seperate pdf file.

Task 11 - Spike Summary Report

What was Learned?

💡 This spike heavily emphasized string manipulation and data processing with nearly the entirety of the functionality provided by the World and WorldLoader classes designed to process a formatted text file.

Additionally, the use of maps in C++ was unfamiliar to begin however with some research the pro's and con's were understood and the project transitioned over to the use of maps.