Software Requirements Specification (SRS)
MesoMerchant

Team: Group 4
Authors: Benjamin Alexander, Tynan Hollyer, Nicky Nath, Brandon Swanberg, Raj Vekeria
Customer: Elementary Schools
Instructor: Dr James Daly

1 Introduction

The first section of this document provides an overview of MesoMerchant. It explores both the purpose and scope of the game while also setting up information for the rest of the document. The following three subsections detail the design, limitations, and capabilities of the software created for MesoMerchant; examining the logic and how specific functions interact with one another to run the game. In the remaining sections, the document provides instructions on how to run a prototype of MesoMerchant and a list of references.

The following are subsections of the Software Requirements Specification (SRS) document:

1. Introduction
2. Overall Description
3. Specific Requirements
4. Modeling Requirements
5. Prototype
6. References
7. Point of Contact

1.1 Purpose

The purpose of this document is to detail the software components of MesoMerchant such that the reader will have a better understanding of its capabilities and characteristics. This is achieved through several of the provided Unified Modeling Language (UML) diagrams. This document is intended for those who desire a deeper understanding of both the capabilities and constraints of MesoMerchant.

Additionally, this document explains the purpose of the game and how it serves as an educational tool for elementary students, ages between 10 and 11, in the state of Massachusetts. Students who are currently enrolled in the sixth grade begin learning about the Mesopotamia era and how it shaped much of human history. This game helps illustrate the time period through interactions with NPCs and the environment. Students may also indirectly strengthen their mathematical skills as a result of the trading mechanisms in the game.

1.2 Scope

The developed software, referred to as MesoMerchant, is meant to help educate children about history and the importance of remembering what has come before. It's a telling of history and how much you can learn from the people before you. This application is supposed to help young kids gain knowledge of the history behind Mesopotamia.

This is a Video Game for the purpose of education. This is also known as an edutainment game. The application runs on the Solarus Engine and loads quests the user can play.

The goal of the game is for the player to understand more about Mesopotamia Era civilizations, interact with those civilizations, and answer questions about those civilizations. Through natural story progression with combat, dialogue, and items the player learns about these ancient cultures. The benefit of this is that a student learns while at the same time enjoying the experience, which hopes to produce better learning outcomes.

The player will wake up in a straw-built hut overlooking farmland filled with irrigation lines and crops. In order to complete the game, the player will need to traverse the world of MesoMerchant and obtain treasure, Egyptian Gold, from the king of Babylon. To achieve this, the player will take the role of a merchant and interact with NPCs through combat, trading, and or communicating to acquire specific items that will allow them to progress through the story. The player will have the opportunity to acquire unique items based upon their location on the map. Some of the items they collect will reflect the culture of the area or represent certain story elements from well-known literature of the period such as the *Epic of Gilgamesh*.

Teaching is accomplished by using trading, exploration, npc interaction, and setting to make the player feel like they're in the Mesopotamia era as a merchant. The challenges they play through in the story will help them and test them on learning about the era.

1.3 Definitions, acronyms, and abbreviations

MesoMerchant - Game Title

Treasure - Other word for Item in game engine

Quest - Main data file for Game, stores all lua code and assets

Solarus - Game engine used to develop MesoMerchant

UML - Unified Modeling Language, modelling language for various designs

SRS - Software requirements specifications

NPC - Non-player character, a computer controlled character that appears human

Edutainment Game - Educational and entertainment video game

Player - The user interacting with the game

1.4 Organization

The next section will give an overview of the product we want to create as well as the details we need to take into account in order to make everything. In Section 3, we go over the specific requirements of what we are going to make. In Section 4 we go over detailed

illustrative design documents to further extrapolate that functionality. In Section 5, we go over how to play the product demo and the features/UI we've already implemented. In Section 6, we conclude with references used in designing this document and developing our game.

2 Overall Description

The information in this section will cover the characteristics of the software and the end user. It will also address both the hardware and software constraints and or requirements of the edutainment game.

2.1 Product Perspective

When choosing a topic to teach kids we collectively agreed upon history, With that being said we pick the history era if Mesopotamia is a historical topic most kids start learning around the 6th grade. Around this grade students are learning Mesopotamia which was known for literature, mathematics and astronomy. Which all ties into trading goods among the public at the time.

The way we implemented a way to teach kids on this topic was by trading, the trading aspect leads kids to learn how they traded for goods and started the first biggest of trading goods at the time. The player must travel with each area learning historical events and be able to trade their items well in order to gain better material. One of the bigger elements of MesoMerchant is the trading system and being able to gain items that lead you to the next area.

With the Solarus Engine it's very easy to implement aspects of learning and interactive gameplay for a 6th grader to grain interest within the game, we expect to primarily focus on the user interfaces which include: title menus, load and save menus, action/interaction menus, and dialogue menus.

2.2 Product Functions

The major functionality of the MesoMerchant is the trading system for the game with each town and each person you talk to is a different encounter that causes different challenges when trying to haggle your way through the town. Within each area, there are town folk who have lived in the area who you can talk to, then answer historical questions that could grant you items that help guide the player within the area and explain the importance of the item.

2.3 User Characteristics

The target demographic of this edutainment game are ten- to eleven-year old children who are enrolled in the sixth grade. Their level of expertise and skill level can be ascertained using DoE of Mass. Educational Frameworks.

For History, students are expected to understand the facts about a specific area (like where they were located or what goods they produced). Furthermore, they should be able do some extrapolation from that data. For instance, they should be able to answer what the effects were of the Nile flooding on an area.

This information is hard to quantify into specific learning outcomes, but we can expect most 6th graders to be able to memorize information and use that information for minor inferences.

Also, due to the general curriculum taught in other subjects listed in this document outside history we also expect a proficient level of English and meaning from sentences, at least addition/subtraction skills, and a level of technical fluency that encompasses our 6 button layout.

We also expect the user will have playtimes of 30 minutes a session due to attention and limitations on time. A cohesive level should be playable within this time frame.

2.4 Constraints

The other main constraints was Solarus itself; the game engine is very nice and simple to work with along with coming with premade sprites and code. As well as having to learn Lua, a different programming language along with having to program in features we wanted that were not initially given with the engine. The other properties of Solarus made it more complex to do the trading system, when attempting to give the townspeople interactive dialogue figures with more than one for the user to respond with, you could only make yes or no questions.

This means that further development is needed to create a significant amount of custom Lua functions to better fit our needs.

2.5 Assumptions and Dependencies

We are assuming the player has internet access and a hardware device capable of running either a Windows, Mac, Linux, Android, and Raspberry Pi, any of these one OS systems can run the game. The game does require internet access in order to download Solarus and the separate Quest Rom files needed in order to play the game. Rudimentary technical knowledge in downloading, searching, and selecting files is assumed. From an accessibility standpoint we expect users to have input control in the form of a keyboard or accessible joystick that can move the character with some level or precision and timing.

2.6 Apportioning of Requirements

Currently we are only scheduled to do two levels at a total playtime of 30 minutes. More levels in the game would create a longer play experience and allow for better repetition of knowledge. This would likely result in better learning outcomes for the student.

Future releases could add in more levels to test more cities in history and add more challenges. Essentially, the game is limited by our development time. More dialogue, maps, and items could be added to the existing mechanics to extend the game.

We believe that our combat, trading, NPC dialogue, and item mechanics will be sufficient to make the game educational and engaging. But adding more detailed or other mechanics could also be addressed in future releases.

Furthermore, we could add in user logins and achievement tracking that teachers could use to track progress. This could help if the game was released in conjunction with a course on the material covered in the Mesopotamian era.

3 Specific Requirements

1. Menu Requirements
    - 1.1. Main Menu
        - 1.1.1. Start a new game
        - 1.1.2. Load a previous game
        - 1.1.3. Exit
    - 1.2. Pause Menu
        - 1.2.1. Save current game
        - 1.2.2. Quit game
    - 1.3. Inventory Menu
        - 1.3.1. Item Menu for Items that player is trading
        - 1.3.2. Item Menu acts as an objective Menu
        - 1.3.3. Items have a known, owned, and used state that is visually discernible
2. World Requirements
    - 2.1. Areas/Cities
        - 2.1.1. Each area is a historical location in the ancient world
        - 2.1.2. Subsequent areas are connected by historical trading/merchant activity
        - 2.1.3. Each area will have Non-Player characters (NPCs) within them
        - 2.1.4. Areas will have elements that are historically accurate to teach history
    - 2.2. Motivation

2.2.1. The protagonist is a merchant trader

2.2.2. First area is protagonist's home city

2.2.3. Protagonist must work and trade in Mesopotamian era to advance and learn about the world of Ancient Mesopotamia

2.2.4. End goal is to make it to the final city, so the trade route can be established

2.3. Gameplay

2.3.1. Areas may be moved between by interacting with the protagonist's carriage

2.3.2. In order to move to an area, a number of items must be possessed (Eg. If there is a desert between the current area and the next , the player must have water, cloth, dust masks, etc.)

2.3.3. The player discovers what items are required to move on by conversing with NPCs

2.3.4. After learning the items, the player must find a merchant NPC who sells each item

2.3.5. The player must buy each item using money (or maybe other items if it is an area which did not use currency at the current historical time)

2.4. Items

2.4.1. Items that player trades are historically accurate and relevant to setting

2.4.2. Item List

2.4.2.1. Leather Garment

2.4.2.2. Cedar Wood

2.4.2.3. Cooking Oil

2.4.2.4. Copper

2.4.2.5. Bronze

2.4.2.6. Indian Ivory

2.4.2.7. Anatolian Silver

2.4.2.8. Stone Tablets

2.4.2.9. Egyptian Gold

2.4.2.10. Reeds

2.4.2.11. Pottery

2.4.2.12. Campion Flower

3. NPC Requirements

3.1. Interaction

3.1.1. NPCs turn to face the player if facing away

3.1.2. Player should be able to interact with the NPC by walking next to them, facing them, and the pressing a W,A,S,D

3.2. Dialogue

3.2.1. Interacting with NPCs brings up a dialogue window

3.2.2. Dialogue window shows text of what the NPC is saying to the player

3.2.3. Dialogue window has options the player can select in order to respond

3.2.3.1. Players should be able to respond to Yes/No dialogue

3.2.4. Dialogue window has a continue option so larger segments of text can be spoken by NPCs

3.2.4.1. Solarus Engine has built in options for 3-line text

3.2.4.2. Text longer than this should be able to be displayed within the Dialogue window by switching to the next set of 3-lines

3.3. Purchasing

3.3.1. Interacting with merchant NPCs gives the option to view the merchant's wares

3.3.2. Viewing wares brings up a UI element listing the wares and their current prices, along with their text description

3.3.3. Items are priced based on historical factors

3.3.4. After selecting an item, the player must confirm that they want the item

3.3.5. Purchasing removes the cost of the item from the player and adds the item to the player's inventory

4. Educational Requirements

4.1. Educational requirements are in line with subjects and testing outcomes described by https://www.doe.mass.edu/frameworks/current.html

4.2. Requirements are specifically for 6th Graders learning about Ancient Mesopotamia

4.3. Learning Taught through Outcomes Through

4.3.1. NPC Dialogue

4.3.2. Object Interaction

4.3.3. Area Setting + Items

4.4. Specific Learning Outcomes

4.4.1. Euphrates River

4.4.2. Farming, Irrigation, and Livestock

4.4.2.1. Barley Staple Crop

4.4.2.2. Irrigation from Euphrates River

4.4.2.3. Surplus of Goods used for Trading and Livestock

4.4.2.4. Player understands through communication or gameplay relevance and effects of irrigation on Mesopotamian society

4.4.3. Code of Hammurabi

4.4.3.1. Law that governed Ancient Mesopotamia Cities

4.4.3.2. Player understands through communication or gameplay elements of Code of Hammurabi Legal Code

4.4.4. Epic of Gilgamesh

4.4.4.1. Elements of Epic of Gilgamesh incorporated into story

4.4.4.2. Oral tradition converted into hand written accounts

4.4.4.3. Player understands through communication or gameplay elements of Gilgamesh's story and what he did

4.4.5. Babylon

4.4.5.1. City in Ancient Mesopotamia

4.4.5.2. River Runs through City

4.4.5.3. Player understands through communication or gameplay the relevance of trading and large cities on Ancient Mesopotamian Society

4.5. Learning outcomes are at a 6th Grade Level

4.5.1. Topics used + questions posed are taken directly from MA Academic outlines for 6th grade level

4.5.2. Vocabulary used is Comparable to the language used in Suggested Resources in MA Academic outlines for 6th grade level

4 Modeling Requirements

4.1 Use Case Diagrams

The use case diagram details the interaction between the player and MesoMerchant. The following diagram documents each case a player will encounter and the actions and or events that will occur. For example, when a player starts MesoMerchant, the player will first view the main menu where they will have the option to load a previous game, exit the game, or start a new game.

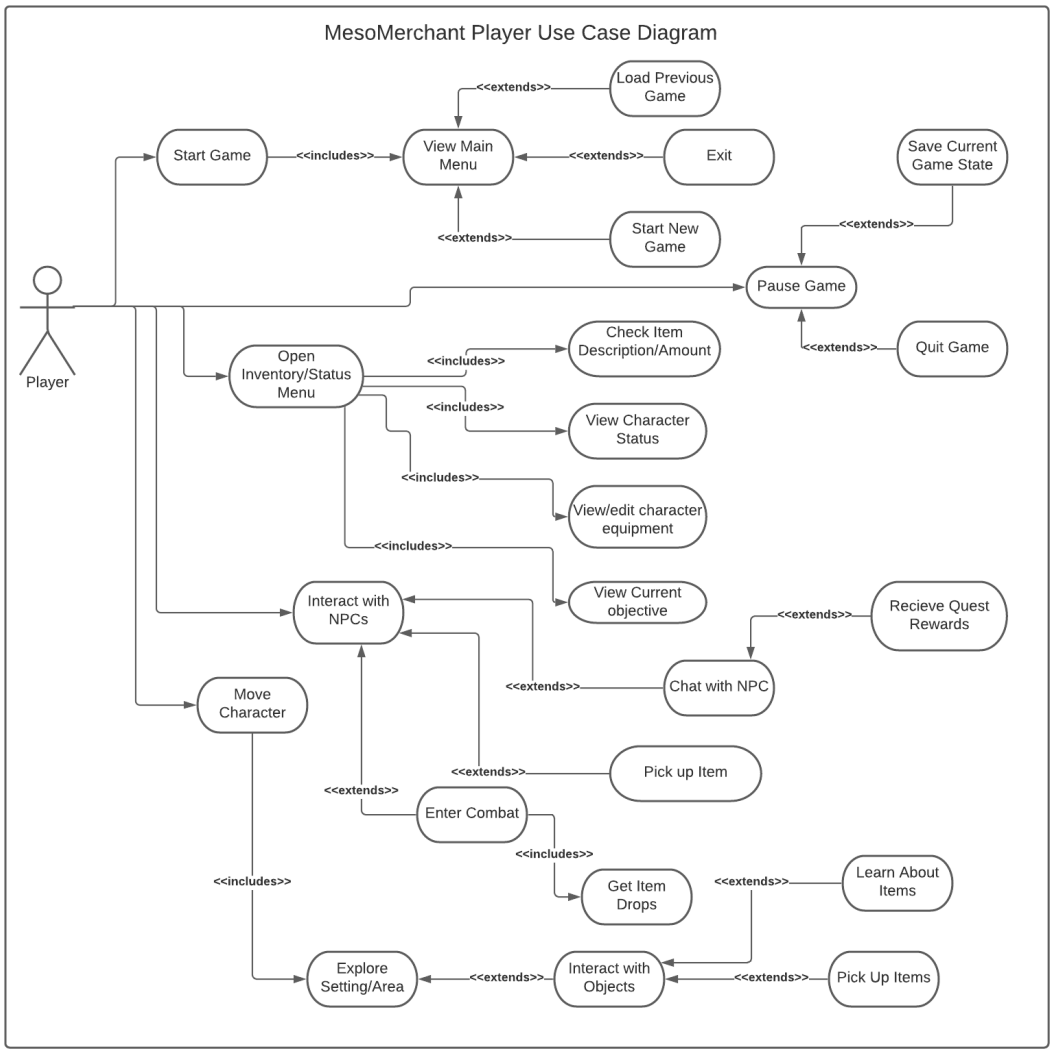

Figure 4.1.1

Use Case Name:	Start Game
Actors:	Player
Description:	Start a new game.
Type:	Input
Includes:	View Main Menu
Extends:	None
Cross-refs:	None
Uses cases:	View Main Menu, Load Previous Game, Exit, Start New Game

Use Case Name:	View Main Menu
Actors:	Player
Description:	The Main Menu appears to give the player a selection of choices to start their game, with the option of Loading, Creating a New Game, or Exiting Completely.
Type:	Input
Includes:	None
Extends:	Load Previous Game, Start New Game, or Exit
Cross-refs:	None
Uses cases:	Load Previous Game, Exit, Start New Game

Use Case Name:	Pause Game
Actors:	Player
Description:	While playing the game, allows the user to pause the game and save that game state for replaying levels later on.
Type:	Input
Includes:	None
Extends:	Save Current Game State, Quit Game
Cross-refs:	None
Uses cases:	Save Current Game State, Quit Game

Use Case Name:	Open Inventory/Status Menu
Actors:	Player
Description:	Shows user inventory to display what the user has collected within each area as they progress.
Type:	UI Element
Includes:	Check Item/Desc, View Char Status, View/ Edit Character, View current objective
Extends:	None
Cross-refs:	None
Uses cases:	User inventory

Use Case Name:	Interact with NPCs
Actors:	Player
Description:	Interact with NPCs in order to gain items and learn about the world. Interaction can take the form of dialogue or combat. Interactions give the player an opportunity to gain items in some way.
Type:	Input
Includes:	None
Extends:	Chat with NPC, Pick Up Item, Enter Combat
Cross-refs:	None
Uses cases:	Chat with NPC, Receive Quest Rewards, Pick Up Item, Enter Combat, Get Item Drops

Use Case Name:	Chat with NPC
Actors:	Player
Description:	An NPC who trades/talks to the user, progresses the story and teaches the user about that specific history and trades user items.
Type:	Input
Includes:	None
Extends:	Receive Quest Rewards
Cross-refs:	None
Uses cases:	Chat with NPC, Receive Quest Rewards, Pick Up Item, Enter Combat, Get Item Drops

Use Case Name:	Enter Combat
Actors:	Player
Description:	The player can enter combat with an enemy that drops items. Items are needed for level progression.
Type:	Input
Includes:	Get Item Drops
Extends:	None
Cross-refs:	None
Uses cases:	Chat with NPC, Receive Quest Rewards, Pick Up Item, Enter Combat, Get Item Drops

Use Case Name:	Explore Setting/Area
Actors:	Player
Description:	Lets the User move the character within the game within each area. The exploration of the character helps detail historical settings through repetition learning.
Type:	Input
Includes:	None
Extends:	Interact with Objects
Cross-refs:	None
Uses cases:	Interact with Objects, Learn About Items, Pick Up Items

Use Case Name:	Interact with Objects
Actors:	Player
Description:	The player can interact with certain objects in the settings. This teaches about the objects and further illustrates historical settings. In some cases, the player can receive an object from interaction.
Type:	Input
Includes:	None
Extends:	Learn About Items, Pick Up Items
Cross-refs:	None
Uses cases:	Interact with Objects, Learn About Items, Pick Up Items

4.2 Class Diagram

The class diagram describes the structure of the Solarus engine. The diagram illustrates engine specific names and engine specific organization. This diagram is critical in designing implementation for Solarus. Development needs to be planned around what the Solarus Engine can do and how the Solarus Engine is organized. This helps outline the need for entity design, resources, and scripting. Solarus specifically revolves around map-centric scripting.

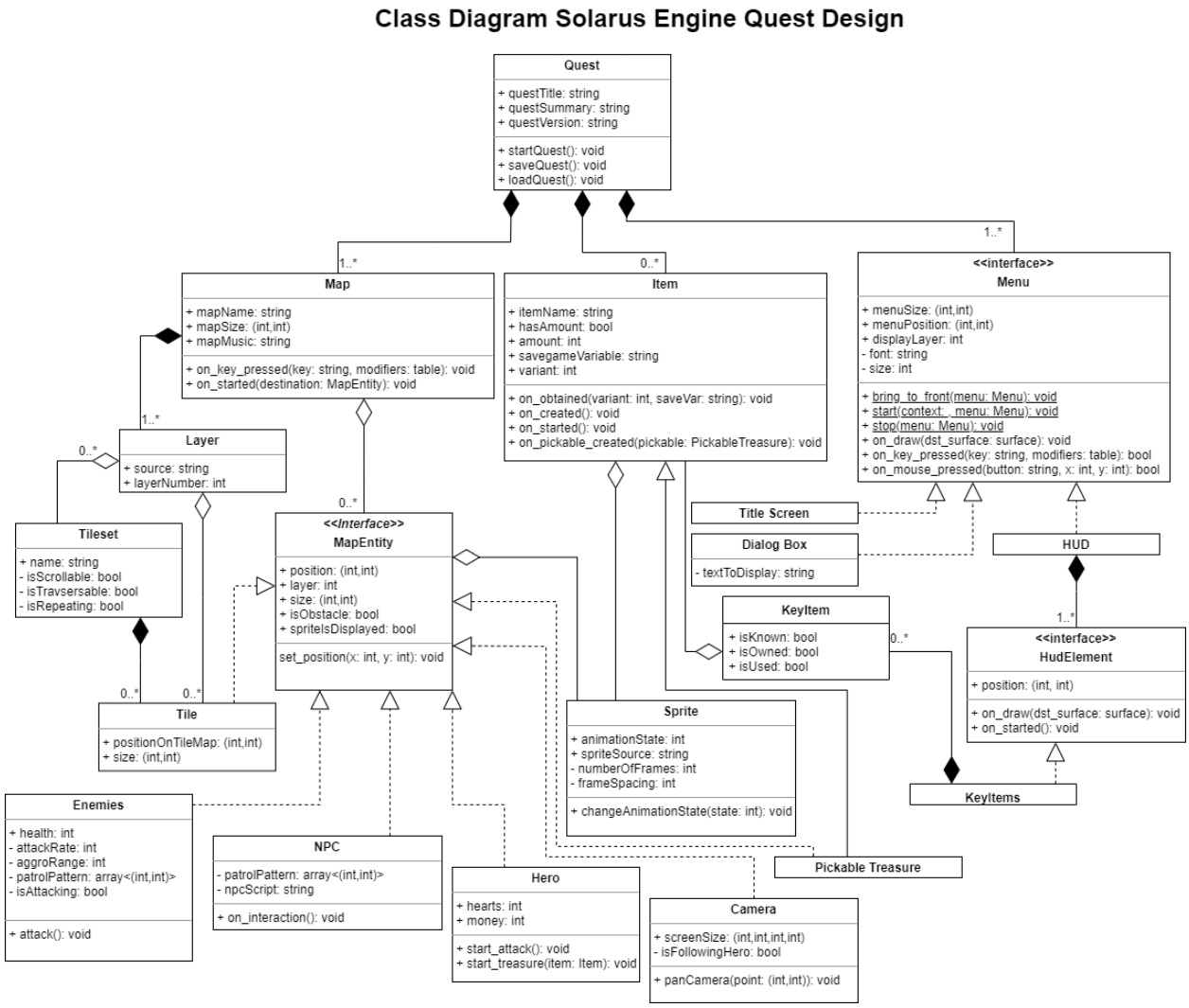

Figure 4.2.1

Data Dictionary

Element Name	**Description**
Quest	The main
Attributes	
questTitle:string	The game's name
questSummary:string	A short explanation of the quest
questVersion:string	The current version of the game
Operations	
startQuest():void	Begins the game

saveQuest():void	Saves the current state of the game
loadQuest():void	Continues the previous save file from the user

Element Name	**Description**
Map	An area of the game
Attributes	
mapName:string	The map's name
mapSize:(int,int)	How big the map is, in pixels
mapMusic:string	Name of the music file
Operations	
on_key_pressed(key:string, modifiers: table): void	Called when the player presses a key
on_start(destination:MapEntitiy):void	Called when the player moves into this map, taking in the location where the player spawned into

Element Name	**Description**
Item	An item which the player can collect
Attributes	
itemName:string	The name of the item
hasAmount:bool	If the player can hold multiple of this item
amount: int	if hasAmount is true, the number of items the player owns
savegameVariable:string	What this is saved as when saveQuest is called
variant:int	Which version of the item the player currently owns
Operations	
on_obtained(variant:int, saveVar:string):void	Called when the player obtains the item

on_created():void	Shows what the user created
on_started():void	Called when the game is started, if this item needs anything initialized
on_pickable_created(pickable: PickableTreasure): void	Called when the game creates a PickableTreasure of this item

Element Name	**Description**
Menu	A GUI element
Attributes	
menuSize:(int,int)	The size of the menu
menuPosition(int,int)	The position of the menu
displayLayer:int	What layer this menu should be drawn on, with higher layers being drawn on top of lower layers
Operations	
bring_to_front(menu:Menu):void	Sets menu to be at front layer
start(context:,menu:Menu):void	Begins showing this menu
stop(menu: Menu):void	Stops showing this menu
on_draw(dst_surface:surface): void	Called when the game is drawing the menu, taking in the surface to draw the menu on
on_key_press(key:string, modifier:table):bool	Called when the user presses a key, returning whether or not the keypress was handled
on_mouse_pressed(button:string, x: int, y int): bool	Called when the user presses a mouse button, returning whether or not the button press was handled

Element Name	**Description**
DialogBox	Handles showing dialog to the user
Attributes	

textToDisplay:string	The text to be shown

Element Name	**Description**
Title Screen	Main Menu of the game

Element Name	**Description**
Tilemap	A set of tiles
Attributes	
source:string	The name of the file defining this tilemap
tileMapName:string	The name with which to reference this tilemap

Element Name	**Description**
MapEntity	An object on the map
Attributes	
position(int,int)	The position on the map
size:(int,int)	Size of the Map objects
Operations	
set_position(x:int, y:int):void	Sets the entity relative to the map

4.3 Sequence Diagrams

In order for the player to access or view specific items, they will need a method of opening up and traversing their inventory. The following diagram documents how the game will handle a player interacting with their inventory. For a player to access their inventory menu, they will first need to first press the enter key. From here, they will be able to traverse the inventory menu using the arrow keys. If a player wishes to select a specific item they must hover over it and press the enter key once again; this will both assign the item to the player and close the inventory menu.

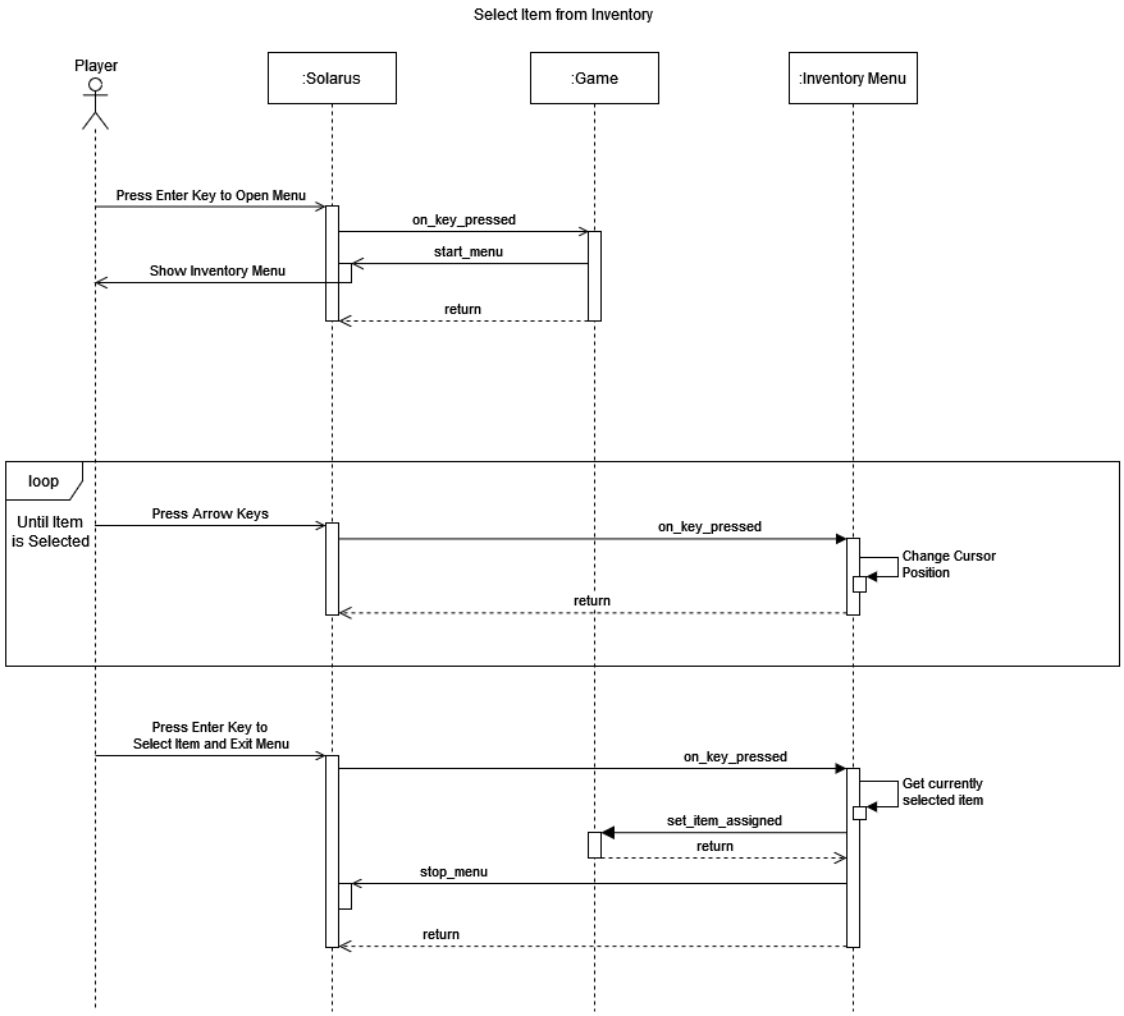

Figure 4.3.1

The following sequence diagram details how the game will handle a player purchasing an item from an NPC. In order to begin a transaction between a player and an NPC, the player must position themselves until they are facing directly in front of the NPC. They will then press an interaction key which will initiate the NPC's dialogue. The player will then have the opportunity to view the NPC's wares menu. From here, the player will traverse the wares menu until they hover over the item they desire. If the player hits enter, they will both receive the item and their payment removed from their inventory.

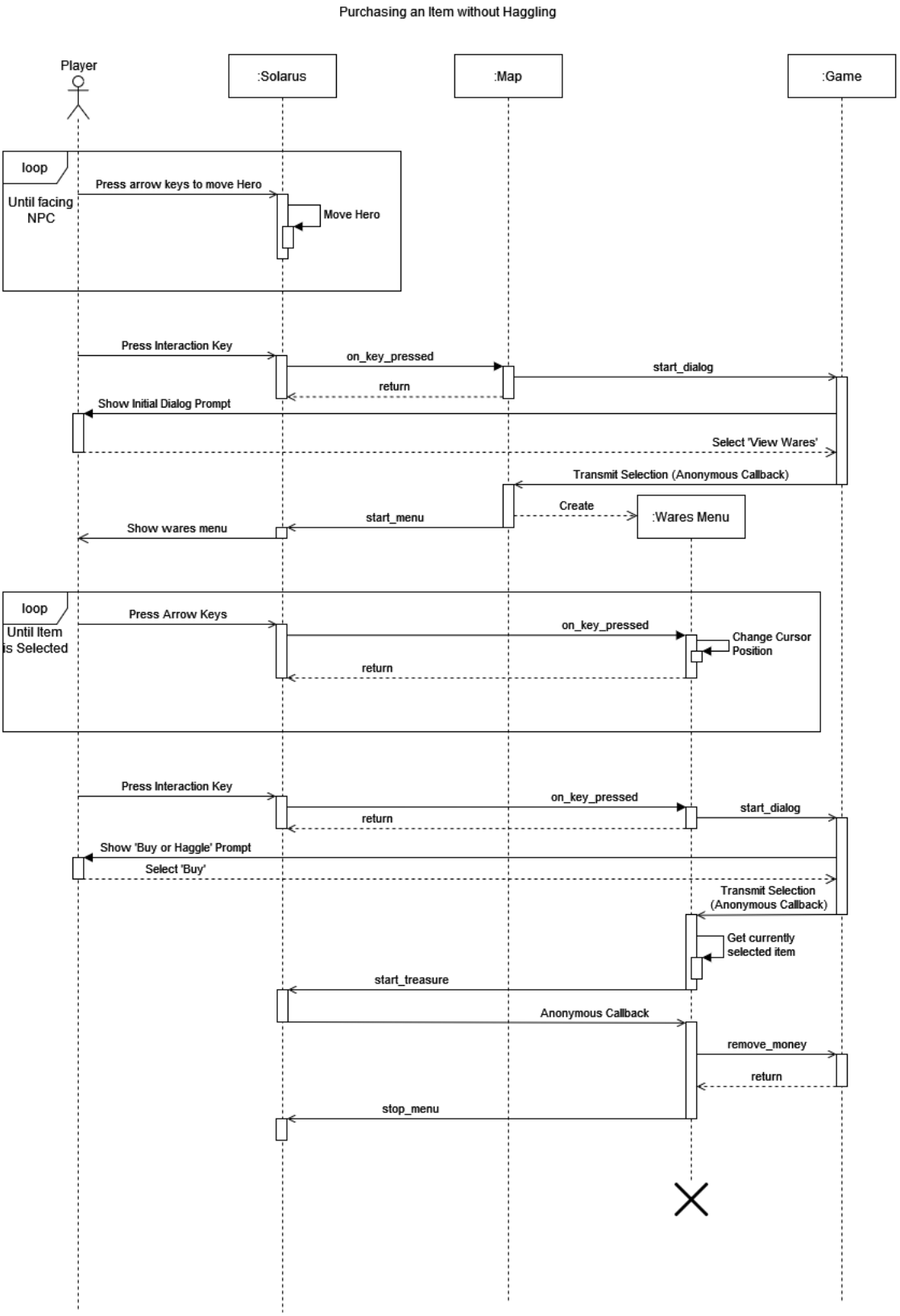
Purchasing an Item without Haggling
Player
:Solarus
:Map
:Game
loop
Until facing NPC
Press arrow keys to move Hero
Move Hero
Press Interaction Key
on_key_pressed
start_dialog
return
Show Initial Dialog Prompt
Select 'View Wares'
Transmit Selection (Anonymous Callback)
Create
:Wares Menu
start_menu
Show wares menu
loop
Until Item is Selected
Press Arrow Keys
on_key_pressed
Change Cursor Position
return
Press Interaction Key
on_key_pressed
start_dialog
return
Show 'Buy or Haggle' Prompt
Select 'Buy'
Transmit Selection (Anonymous Callback)
Get currently selected item
start_treasure
Anonymous Callback
remove_money
return
stop_menu

Figure 4.3.2

4.4 State Diagram

The Item_Known State Diagram (Figure 4.4.1) is used to describe UI item states for drawing. The items are added to the UI element in the top-right hand corner. Items begin as greyed out sprites of their shape when the player learns of their existence. When an item is picked up, it is shown in full color on the item UI. When an Item is Used (traded for), the item disappears from the UI.

Item_Known State Diagram

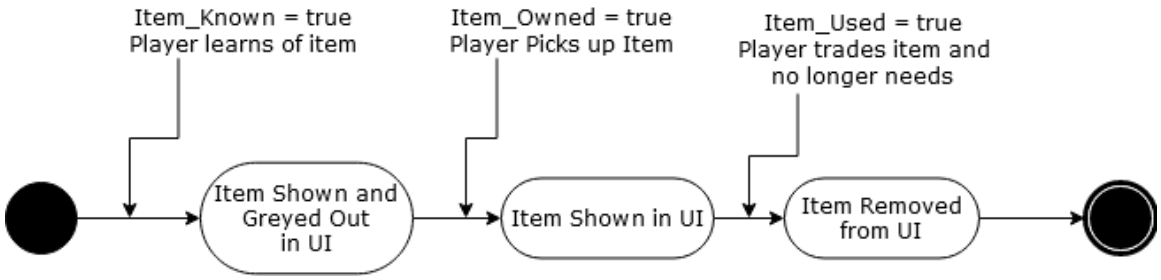

Figure 4.4.1

The Irrigation Animation State Diagram (Figure 4.4.2) describes the logic behind the Irrigation Animation scene. Although this scene is simple, it is used as a core tutorial for the player to help describe change in the game. Through one interaction the player should see a change or output in what they are doing. If the irrigation is not fixed, then the animation of the broken irrigation is drawn. Once the irrigation is fixed, the animation of the fixed irrigation replaces it.

Irrigation Animation State Diagram

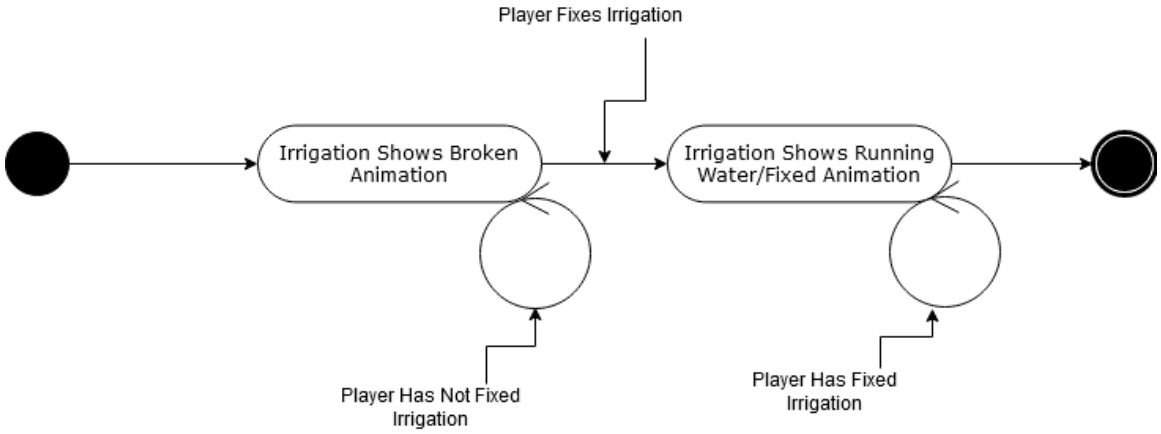

Figure 4.4.2

The NPC Show Dialogue State Diagram (Figure 4.4.3) details the core mechanics of NPC dialogue. Every NPC asks for an item they wish to trade. This gives game objectives for the player of what they should seek out. When a player gets an item an NPC is looking for, a trade is completed and historically relevant dialogue pops up. When the player goes to talk to the NPC again, a thank you is repeated.

NPC Show Dialogue State Diagram

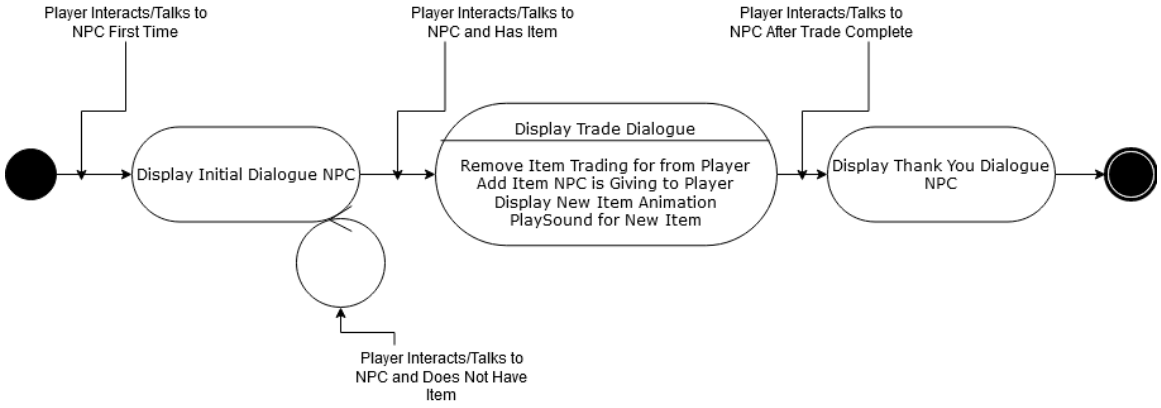

Figure 4.4.3

5 Prototype

Our prototype demonstrates core UI elements and limited functionality for the game. This functionality includes a map, combat, item inventory, NPCs, and dialogue. These features will be used to help create a game that is fun to explore and is also educational. Through the use of the environment, items, and NPC interactions, the players learn and are tested about Mesopotamia era history.

5.1 How to Run Prototype

The MesoMerchant prototype is able to run on the following operating systems: Windows, Mac, Linux, Android, and Raspberry Pi. Solarus acts similar to running an emulator such that quests are loaded like ROM files.

MesoMerchant currently runs on version 1.6.5 of Solarus which can be downloaded from the following website:

(https://www.solarus-games.org/en/solarus/download)

The player version of the launcher should be downloaded. Next, the quest download can be obtained by downloading the following repository. To the right of "Clone" in top right, click the 3 dots and select "Download as Zip":

(https://dev.azure.com/team4sedev/Team4/_git/phoenician)

Unzip both files. Open the Solarus folder and run "solarus-launcher.exe". Click "File"->"Add Quest". Then navigate to the file titled "Phoenician/Data/quest.dat", from the repository. Then press play.

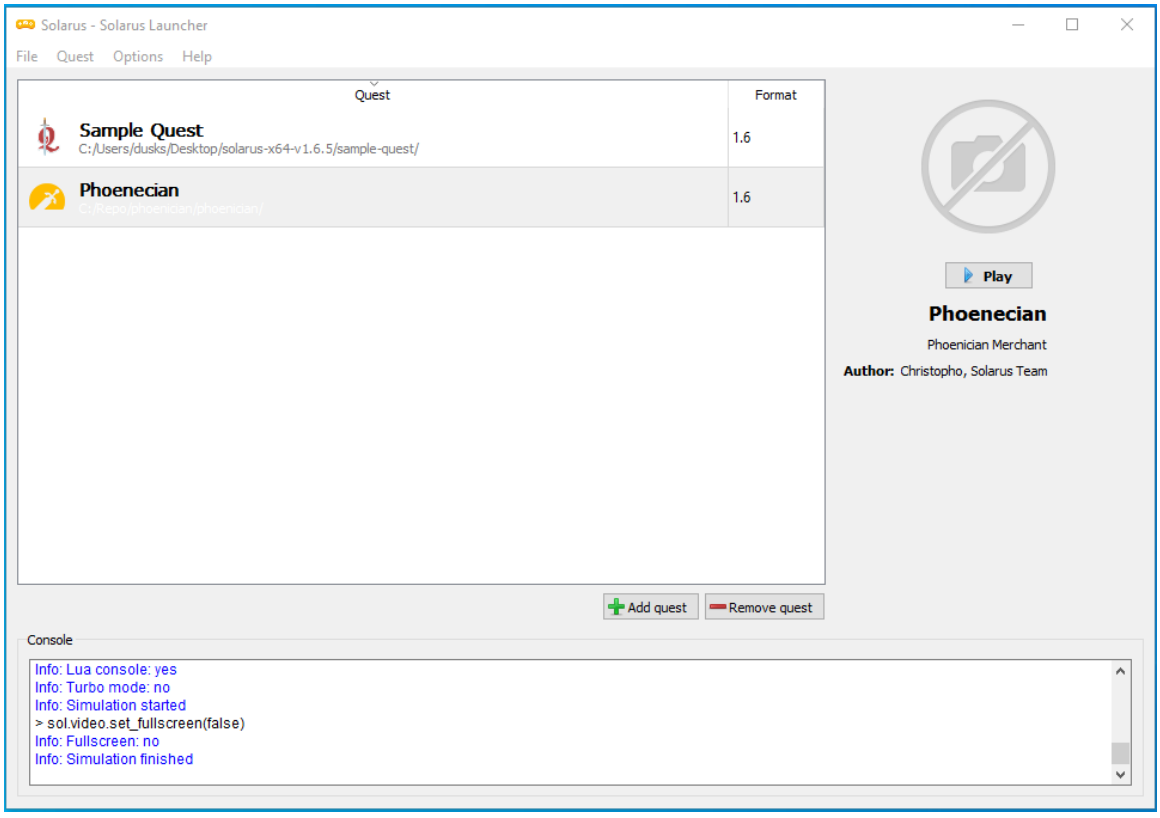

Figure 5.1.1: Solarus Launcher with MesoMerchant Ready to Play

5.2 Sample Scenarios

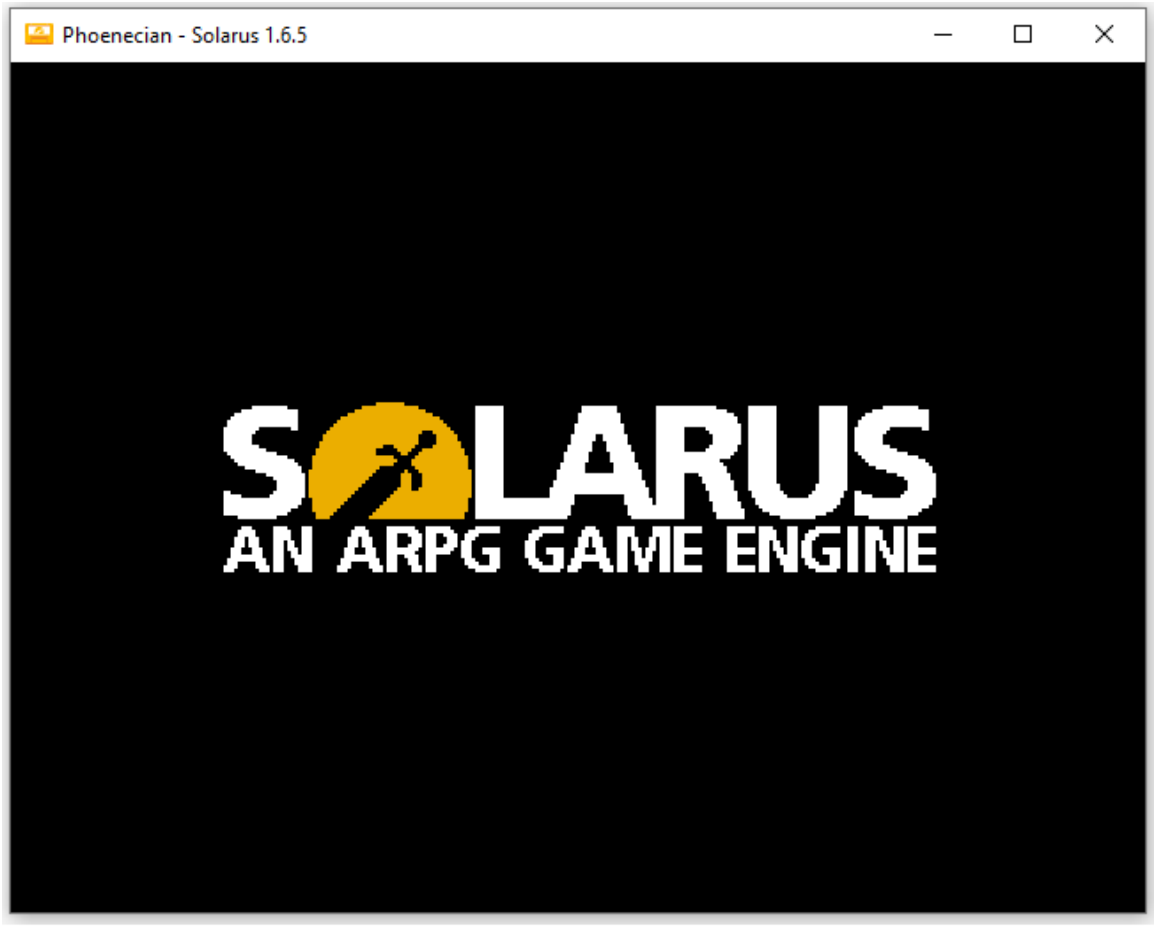

Figure 5.2.1: Solarus Title Screen on Game Load.

Figure 5.2.2: MesoMerchant Title Screen with New Game, Load Game (if available), and Quit Options.

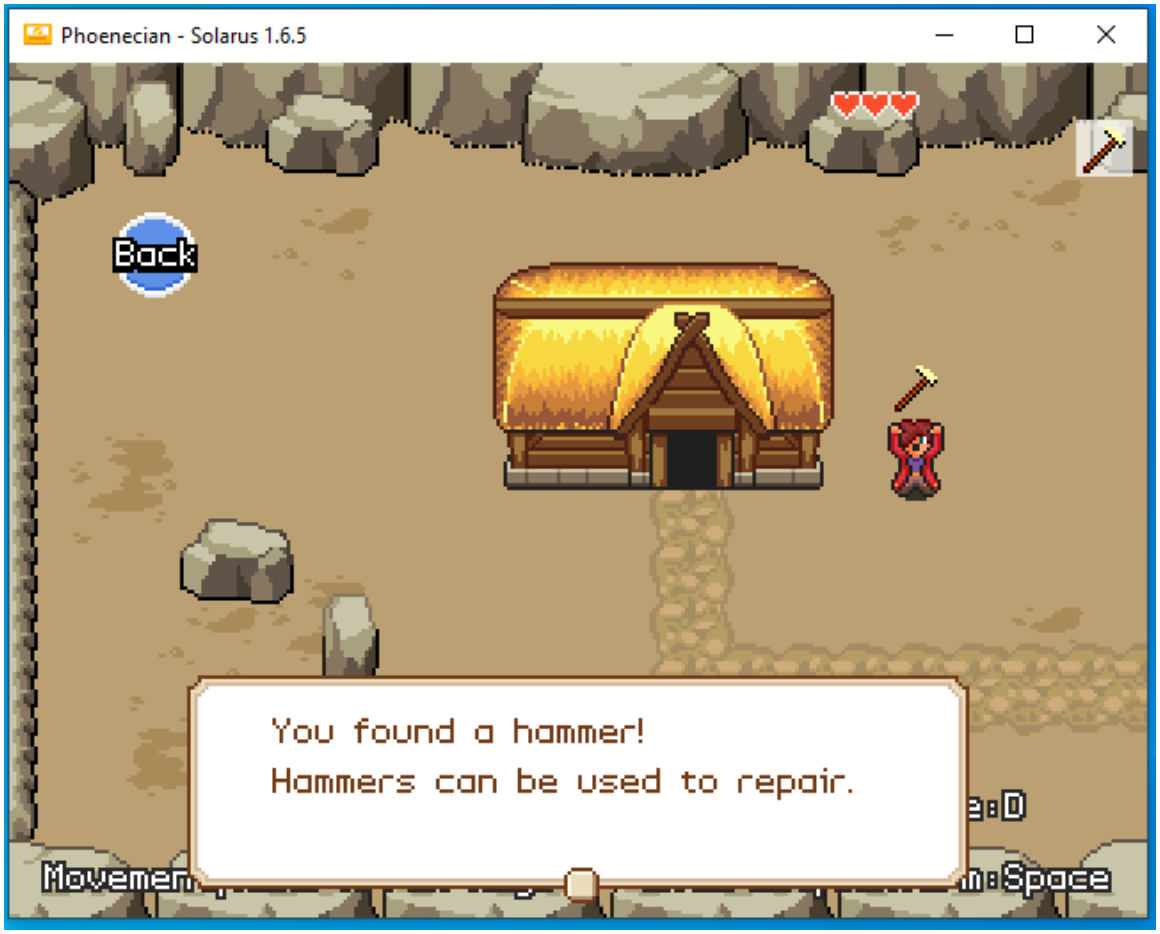

Figure 5.2.3: Receive Item Animation, and add to Inventory UI.

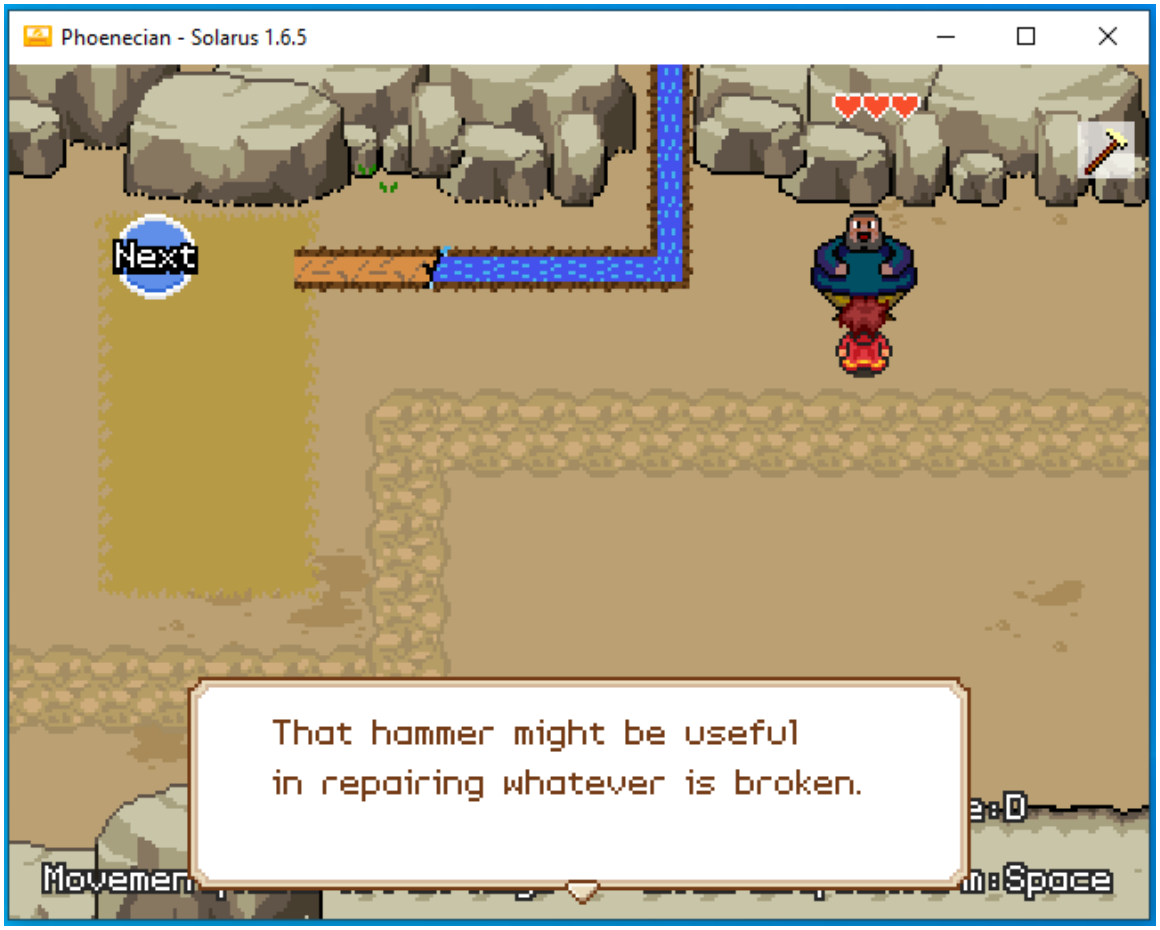

Figure 5.2.4: NPC Interaction and Dialogue guides gameplay. Setting is historically accurate and teaches about irrigation.

Figure 5.2.5: Player fixes irrigation and water flows. Object interaction guides Historical context. In this case, the usage of irrigation in Ancient Mesopotamia.

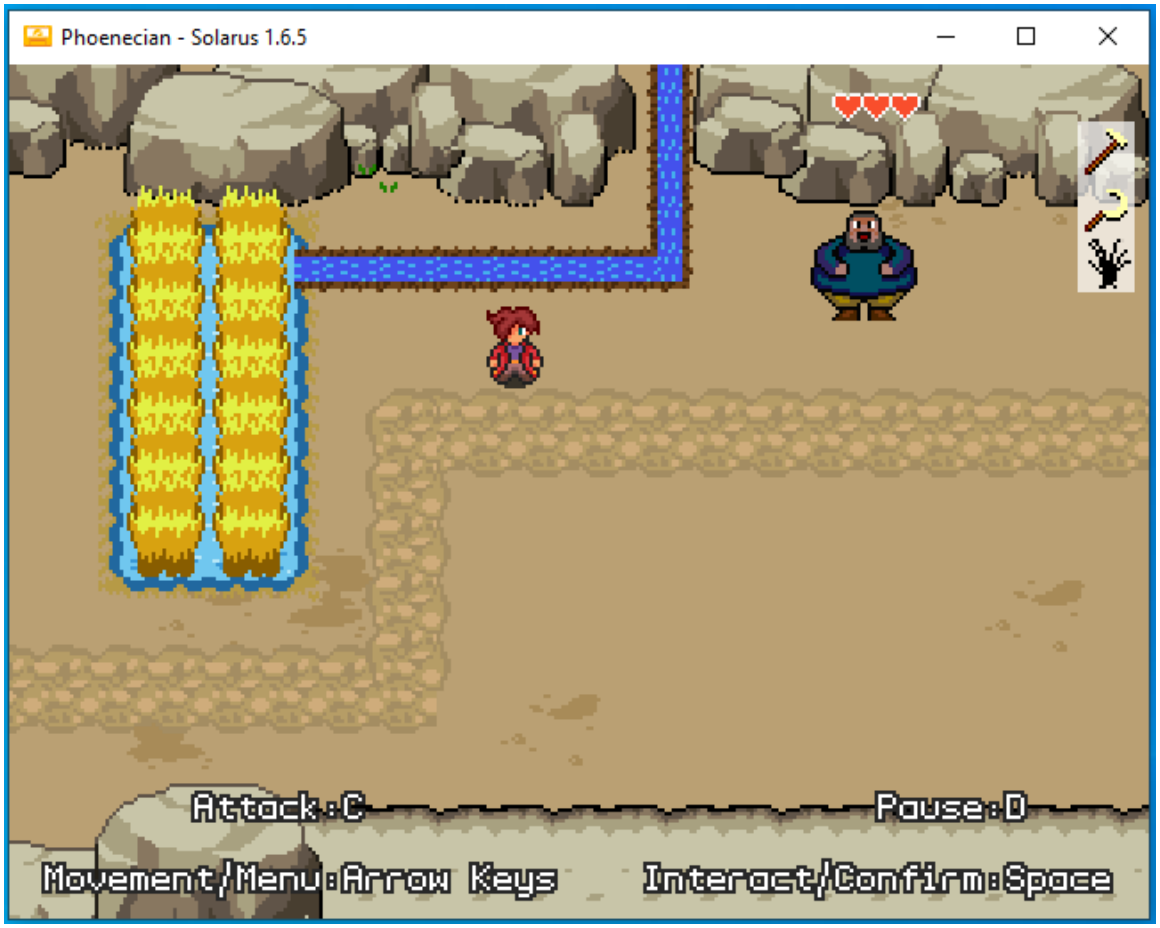

Figure 5.2.6: Learning an item exists adds it to Item UI, which acts as an objective list.

Figure 5.2.7: Reaping Barley adds it to player inventory. Items can then be traded.

Figure 5.2.8: Trading items with NPCs is a core mechanic. It uses historically relevant items and drives player interaction by requesting to trade for items.

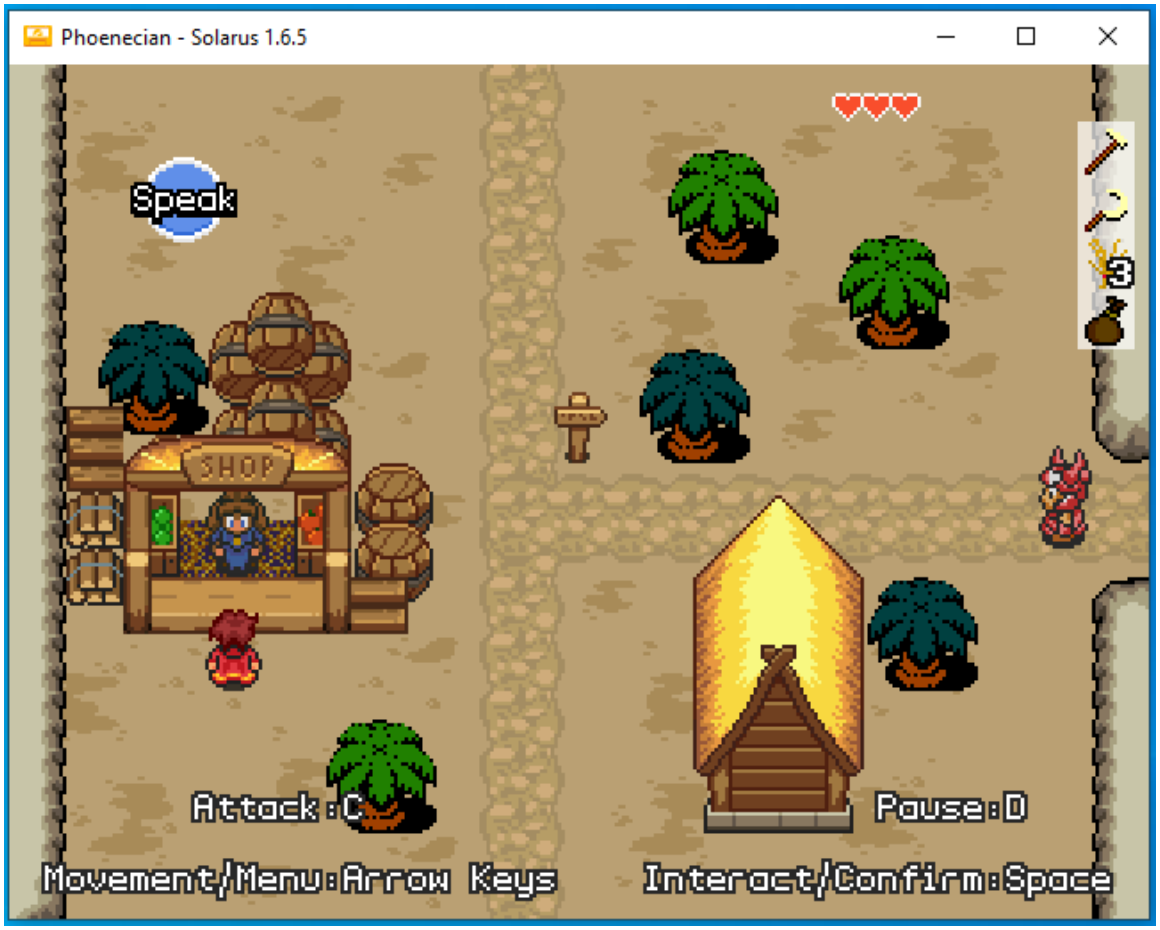

Figure 5.2.9: Trading for items successfully adds new items to your inventory.

Figure 5.2.10: Progressive trading unlocks new areas.

6 References

[1] Crabben, Jan van der. "Agriculture in the Fertile Crescent & Mesopotamia." World History Encyclopedia. World History Encyclopedia, 15 Nov 2021. https://www.worldhistory.org/article/9/agriculture-in-the-fertile-crescent--mesopotamia/. 04 Dec 2021.

[2] D. Thakore and S. Biswas, “Routing with Persistent Link Modeling in Intermittently Connected Wireless Networks,” Proceedings of IEEE Military Communication, Atlantic City, October 2005.

[3] Mark, Joshua J.. "Gilgamesh." World History Encyclopedia. World History Encyclopedia, 29 Mar 2018. https://www.worldhistory.org/gilgamesh/ Web. 04 Dec 2021.

[4] Massachusetts Board for the Department of Elementary and Secondary Education. Massachusetts History and Social Science Curriculum ... https://www.doe.mass.edu/frameworks/hss/2003-08.docx. 2018.

[5] Massachusetts Department of Elementary and Secondary Education, “History and Social Science Framework”, 2018, https://www.doe.mass.edu/frameworks/hss/2018-12.pdf. 2021

[6] "Mesopotamia Trade: Merchants and Traders" History on the Net © 2000-2021, Salem Media. December 3, 2021 <https://www.historyonthenet.com/mesopotamian-merchants-and-traders>

[7] Puhvel, Jaan. "cuneiform". Encyclopedia Britannica, 13 Mar. 2019, https://www.britannica.com/topic/cuneiform. Accessed 4 December 2021.

[8] Saggs, Henry W.F.. "Babylon". *Encyclopedia Britannica*, 7 May. 2021, https://www.britannica.com/place/Babylon-ancient-city-Mesopotamia-Asia. Accessed 4 December 2021.

[9] Solarus Project. Christolpho. https://www.solarus-games.org/. 2021.

[10] Team4.“ Phoenician.” Team4 Project Website, https://rvekeria678.github.io/SoftwareEngineering_Team4_WEB/.

[11] 7Soul1. “420 -Pixel Art- Icons for RPG”. 17 July. 2021, https://www.deviantart.com/7soul1/art/420-Pixel-Art-Icons-for-RPG-129892453

[12] Gif, and Noiracide. “Super Retro World Interior Pack”, https://gif-superretroworld.itch.io/interior-pack

[13] Bennyboi_hack. “16x16-weapon-sprites-free”. https://opengameart.org/content/16x16-weapon-sprites-free

7 Point of Contact

For further information regarding this document and project, please contact **Prof. Daly** at University of Massachusetts Lowell (james_daly at uml.edu). All materials in this document have been sanitized for proprietary data. The students and the instructor gratefully acknowledge the participation of our industrial collaborators.