Software Requirements Specification (SRS)
Project X

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 segment of this document describes the purpose of this game and why it is being created. It then explores the scope of the software and how specific functions interact with one another. This in turn will allow for a better understanding of the software and both its functionality and characteristics. In the end, the document describes how one may run a prototype of the game.

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 the game 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.

Additionally, the 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 software, called Phonecian, being developed 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 GUI Program/Video Game for the purpose of Education. This is also known as an Edutainment game. The application runs on the Solaris Engine and loads quests the user can play.

The goal of the game is for the player to understand more about Phoenician Era civilizations, interact in 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.

A player from the modern-era is sent back in time by his/her life-long enemy. In order to get back to his original time period, he/she must collect a specific item/items. Players must take the role of a merchant and trade various items until they obtain the item(s) they need to return back to their original time period. The Player might need to travel to other cities to obtain specific items and learn cultures. The Player may receive gifts if they correctly answer questions related to the time period. The Player may encounter hostile entities on their travels between cities.

The Player may encounter hostile entities on their travels between cities. They might need to save their father from pirates. In order to do so, they must raise certain funds (items) in exchange for his safe return.

Teaching is accomplished by using trading, exploration, npc interaction, and setting to make the player feel like they're in the Phoencian 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

Phoenician - Game title for Phoenicient Merchant Game the document describes

Treasure - Other word for Item in game engine

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

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

1.4 Organization

The next sections 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

The entire product is within the system of the Solaris Engine. This system sits in a larger context of Edutainment games for audiences in the 6th grade. The constraints of this don't create significant product constraints for hardware or software. The main constraints are created by the user through what is accomplishable of a 6th grader and what it is appropriate for them to learn.

This means that the major constraints are on the UI and interaction. The UI and interaction needs to be simple and intuitive enough that a 6th grader with less developed motor skills and technical fluency can still play the game.

Even more important than this is the content of the game. The content must be reflective of both the type of history 6th graders are learning and the level of questions/challenges that are reasonably expected for a 6th grader to understand.

The main context for Phonecian learning the trading and exploration skills of Mesopotamia and overview of the skills used during that Phoencian era. One of the bigger elements of Phoencian is the trading system and being able to gain items that lead you to the next area.

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 Phonecian 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 down. 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 decent graph 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 of startings was Solarus itself, the game engine is very nice an simple to work with along with coming with premade sprites and code with a caveat of having to learn a Lua a different programming language along with having to program in features we wanted that was not initially given with the engine. The other properties of the 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 the internet and a hardware device capable of running: Windows, Mac, Linux, Android, Raspberry Pi. Additionally, a browser is needed to access both the Solaris game 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 playtime of 30 mins each. 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 Phoenician 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/Status Menu
        1.3.1. Check item description/amount
        1.3.2. View character status
        1.3.3. View/edit character equipment
        1.3.4. View current objective(s)
    1.4. World Menu
        1.4.1. Display a map of the world + time period
        1.4.2. Show the historically accurate locations of areas
        1.4.3. Each area has a piece of descriptive text for the player to read
2. World Requirements
    2.1. Areas
        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 (Silk Road, etc.)
        2.1.3. Each area will have Non-Player characters (NPCs) within them
    2.2. Motivation
        2.2.1. The protagonist is a merchant trying to establish a trade route/agreement (eg. Spices from India to Europe)
        2.2.2. First area is protagonist's home city
        2.2.3. Protagonist starts with a sum of money and number of items
        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

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 button/key

3.2. Dialogue

3.2.1. Interacting with NPCs brings up a dialogue window

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

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

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

3.2.5. Dialogue window has a back button, to re-read precious parts of long segments of text

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

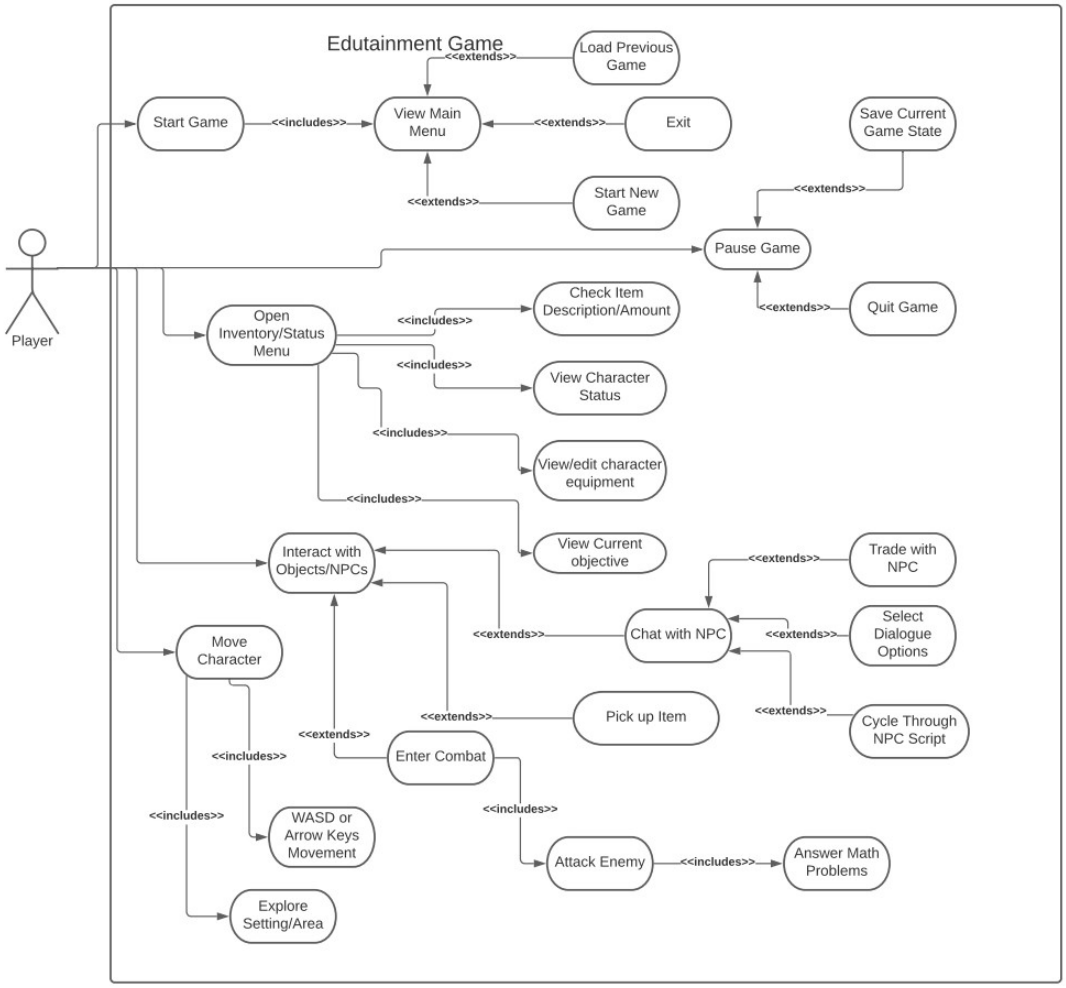
4.1. Educational requirements are in line with subjects and testing outcomes described by https://www.doe.mass.edu/frameworks/current.html

4.2. Primary Source Documents dealing with ancient cities and societies with 1:1 map + detail design

4.3. Mathematics at the appropriate skill level are tested through combat

4.4. Historical understanding and reading comprehension are tested through dialogue with NPCs

4 Modeling Requirements



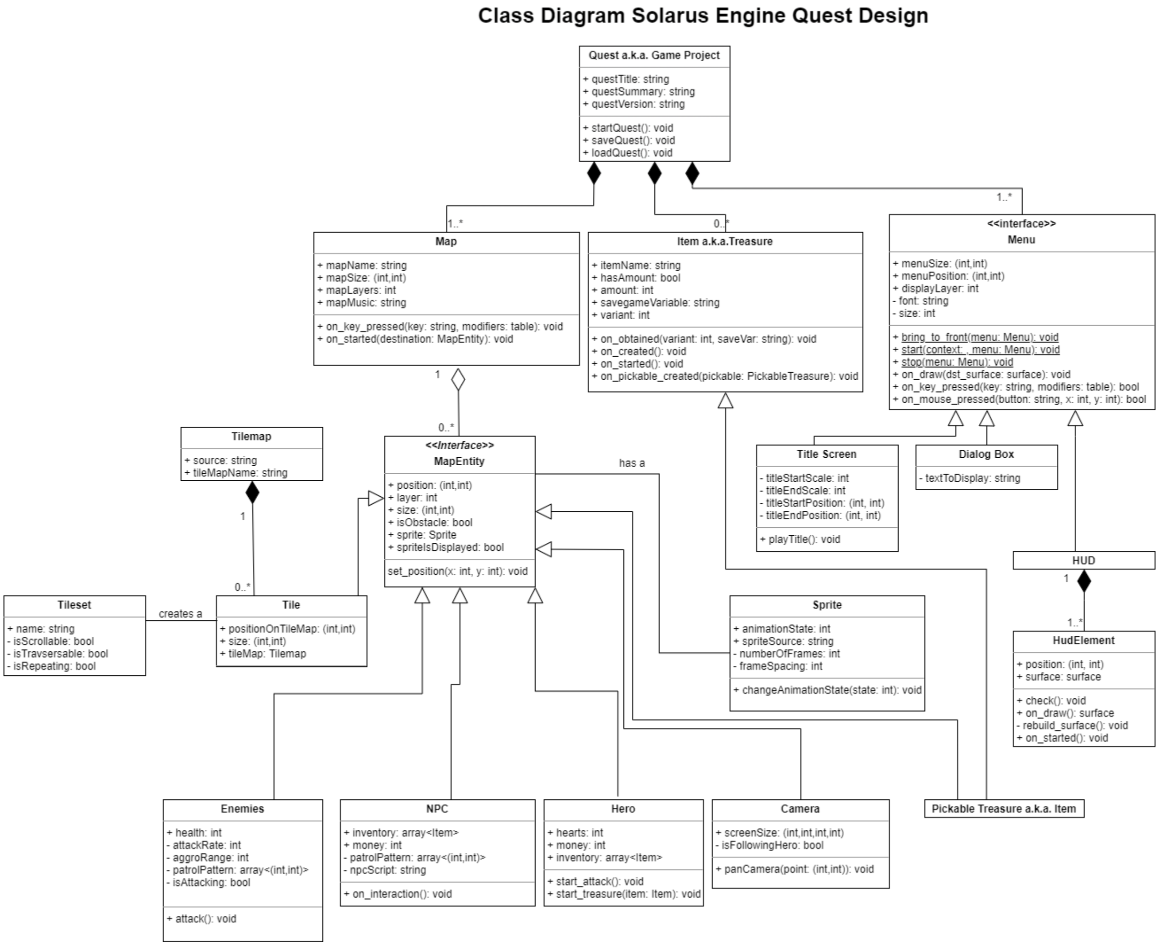
Use Case Name:	Start Game
Actors:	Player
Description:	Starts a new game.
Type:	Input
Includes:	View Main Menu
Extends:	Load Previous Game, Exit, Start New Game
Cross-refs:	None
Uses cases:	View Main Menu, Load Previous Game, Exit, Start New Game

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

Use Case Name:	Interact with Objects/NPCs
Actors:	None Player Character
Description:	A NPC who trades/talk to with the user, that progress the story an teaches the user about that specific history
Type:	Input
Includes:	Attack Enemy, Answer Math Problem
Extends:	Enter Combat, Pick up items, Chat with NPC, Trade, Select Dialogue, Cycle through NPC Script
Cross-refs:	None
Uses cases:	View Main Menu, Load Previous Game, Exit, Start New Game

Use Case Name:	Move Character
Actors:	Player
Description:	Lets the User move the character within the game within each area
Type:	Input
Includes:	The WASD keys movement, Explore Setting/Area
Extends:	Load Previous Game, Exit, Start New Game
Cross-refs:	None
Uses cases:	View Main Menu, Load Previous Game, Exit, Start New Game

Class Diagram



Data Dictionary

Element Name		Description
Quest A.k.a Gaming Project		Is the main window for the game
Attributes		
	questTitle:string	Is use to display the titles of the game on the home screen
	questSummery:string	Explains what the player is doing during an area for a

		quest
	questVersion:string	Display what version of quest the person is on
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		Displays the area that the player is in
Attributes		
	mapName:string	Shows the name of the map
	mapSize:(int,int)	Is use to determine how big the map is
	mapLayer:int	Displays the layers of the map within the town for the other buildings
	mapMusic:string	Plays the music for the appropriate area
Operations		
	on_key_pressed(key:string, modifiers: table): void	A key press from the user to interact with the map
	on_start(destination:MapEntitiy):void	When opening the map it displays the

		destination

Element Name		Description
Item a.k.a Treasure		Display the treasure and explain the use of each one
Attributes		
	itemName:string	Displays the item for the user
	hasAmount:bool	Displays the the player has
	amount: int	Displays the user amount of object
	savegameVariable:string	Saves what the user has gained throughout the game
	variant:int	Counts how many version of an object the player has
Operations		
	on_obtain(variant:int, saveVar:string):void	Shows the user obtained an how much of the object
	on_created():void	Shows what the user created
	on_started():void	Shows were the user has started off with
	on_pickable_created(pickable: PickableTreasure): void	Lets the user pick the item that be grabbed from the ground

Element Name		**Description**
Menu		A Menu when the user pauses the game
Attributes		
	menuSize:(int,int)	Sets the size of the menu
	menuPosition(int,int)	Positions of the menu
	displayLayer:int	Shows the layers of the menu
Operations		
	bring_to_font(menu:Menu):void	Sets font for menu
	start(context:,menu:Menu):void	Shows start Text
	stop(menu: Menu):void	Stops the game and quite
	on_draw(dst_surface:surface): void	Draws text
	on_key_press(key:string, modifier:table):bool	Detects if the button has been pressed to navigate the menu
	on_mouse_pressed(button:string, x: int, y int): bool	Detects if the mouse button has been pressed within the menu

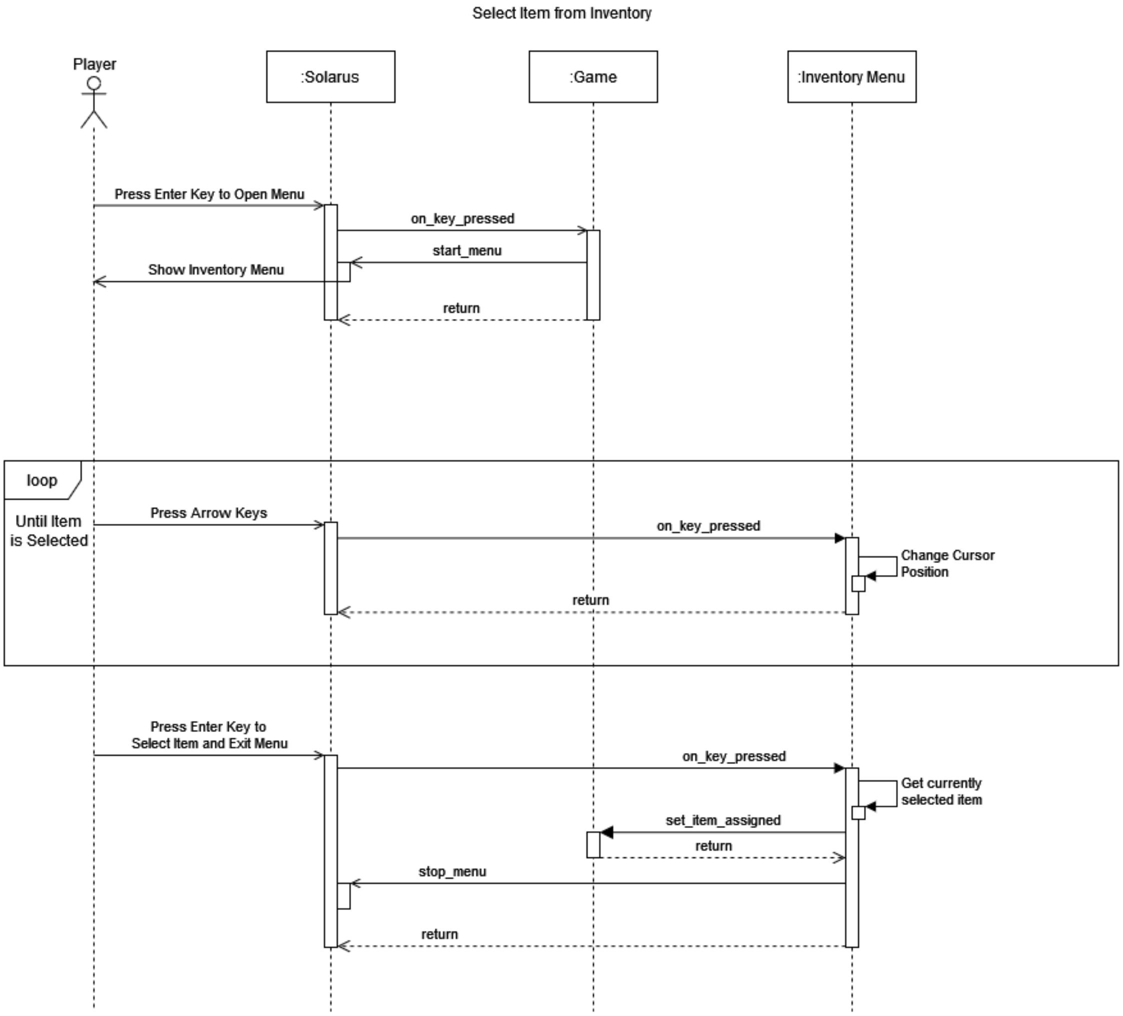
Element Name		**Description**
Tilemap		Tiles for the map
Attributes		
	source:string	The name of the string
	tileMapName:string	Displays the name of the tileMap

Element Name		**Description**
MapEntity		Show the objects you have
Attributes		
	position(int,int)	Set the position of the object
	layer:int	Layers of the object
	size:(int,int)	Size of the Map objects
Operations		
	set_position(x:int, y:int):void	Sets the entity relative to the map

Element Name		**Description**
DialogBox		Starts Dialog between user an NPC
Attributes		

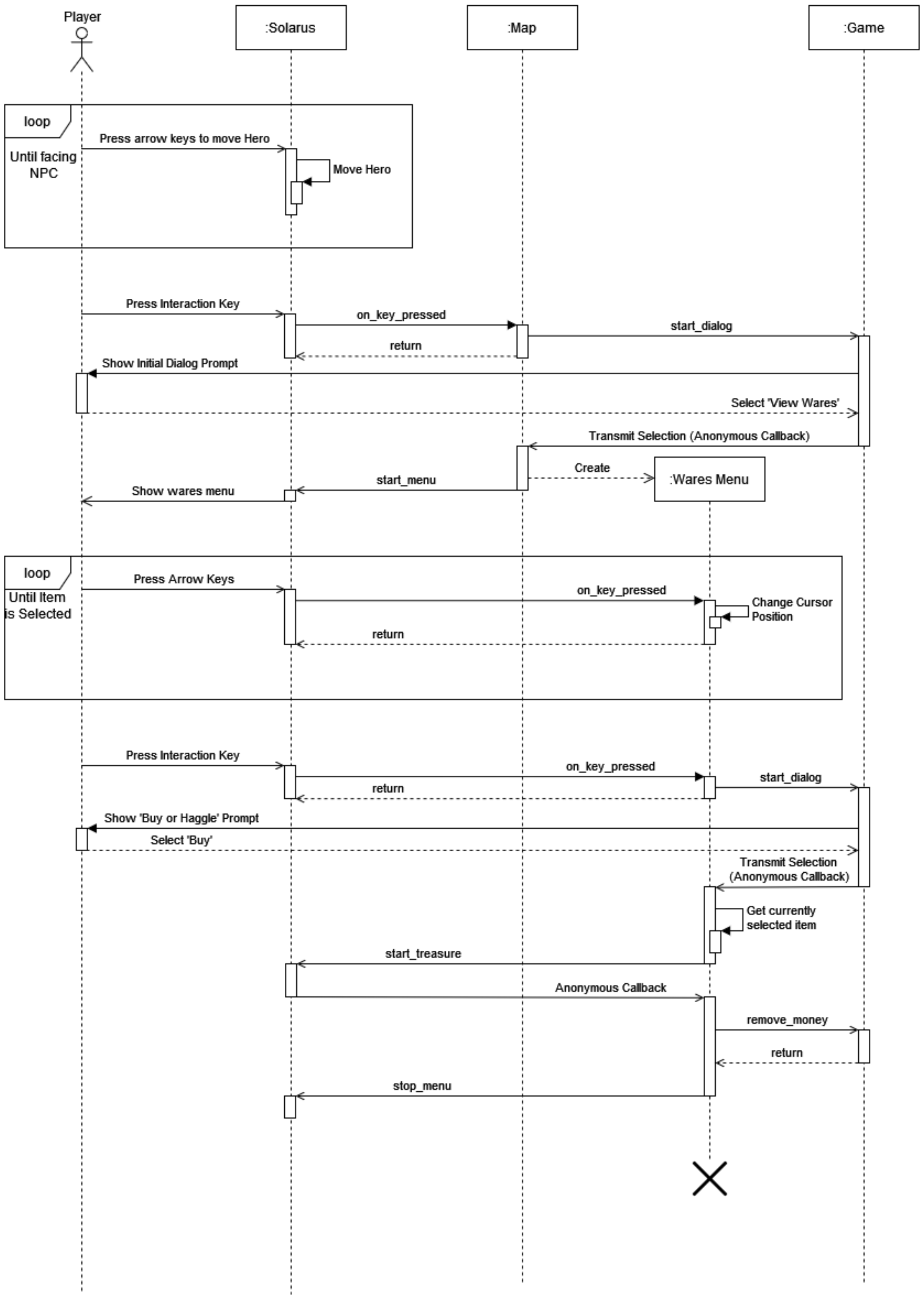
	textToDisplay:string	Shows the text

Element Name		**Description**
Title Screen		Main Menu of the game
Attributes		
	titleStartScale:int	Set size for the text
	titleEndScale:int	Set size for the end
	titleStartPosition:(int,int)	Set Position for Start
	titleEndPosition(int,int)	Set End Position
Operations		
	playTitle():void	Displays the play title



Select Item from Inventory
Player
:Solarus
:Game
:Inventory Menu
Press Enter Key to Open Menu
on_key_pressed
start_menu
Show Inventory Menu
return
loop
Until Item is Selected
Press Arrow Keys
on_key_pressed
Change Cursor Position
return
Press Enter Key to Select Item and Exit Menu
on_key_pressed
Get currently selected item
set_item_assigned
return
stop_menu
return

Purchasing an Item without Haggling



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 environment, items, and NPC interactions the players learn and are tested about Phoenician era history.

5.1 How to Run Prototype

Operating Systems Available: Windows, Mac, Linux, Android, Raspberry Pi

Solaris acts like an emulator and the quests are loaded like ROM files.

1. First download Solaris:

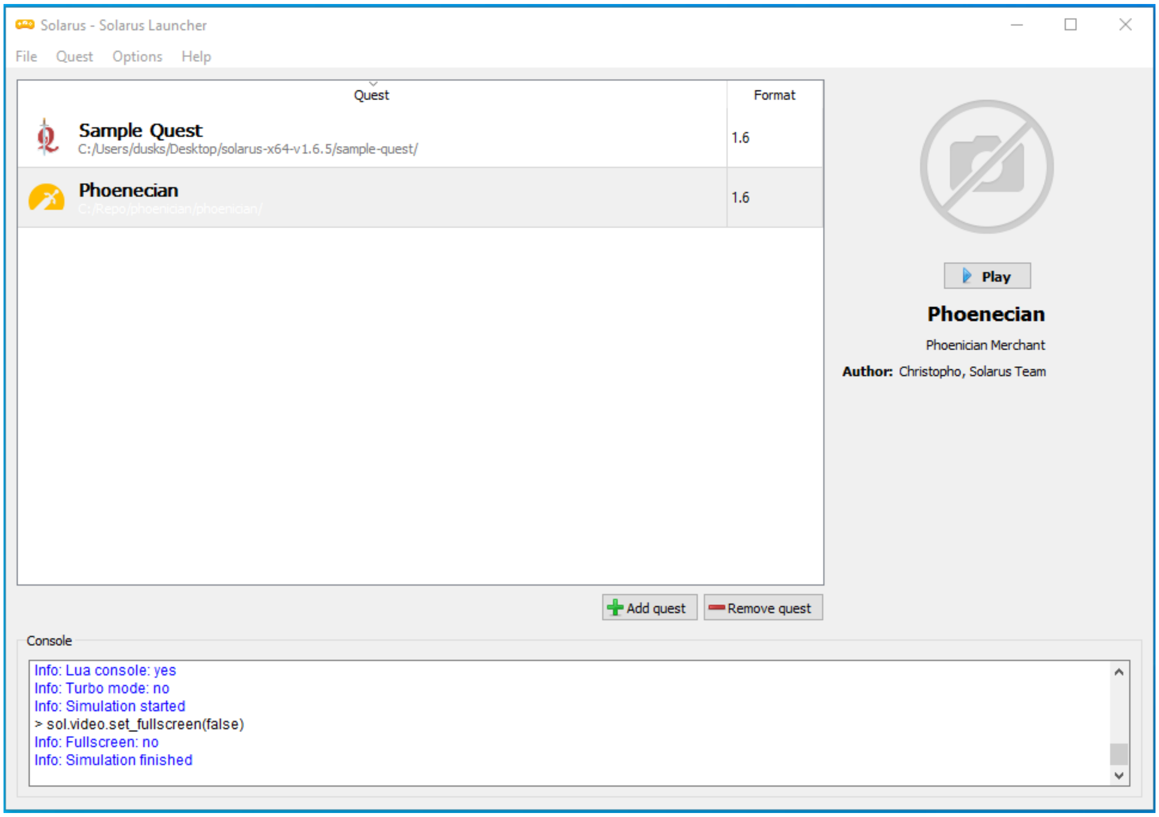
Download Solaris: https://www.solarus-games.org/en/solarus/download

-Click the Player Download

2. Download the Quest file.

Clone Repo Here: https://dev.azure.com/team4sedev/Team4/_git/phoenician

3. Open the Solarus Launcher and Load the Quest file.



Quest	Format
Sample Quest C:/Users/dusks/Desktop/solarus-x64-v1.6.5/sample-quest/	1.6
Phoenecian C:/Repo/phoenician/phoenician/	1.6

Phoenecian

Phoenician Merchant

Author: Christopho, Solarus Team

```
Info: Lua console: yes
Info: Turbo mode: no
Info: Simulation started
> sol.video.set_fullscreen(false)
Info: Fullscreen: no
Info: Simulation finished
```

Figure 5.1.1: Solarus Launcher with Phoenician Ready to Play

Click File -> Add Quest

Navigate to the phoenician/phoenician/data/.
Double click quest.dat.

4. Double click the Phoenician quest to start playing.

5.2 Sample Scenarios

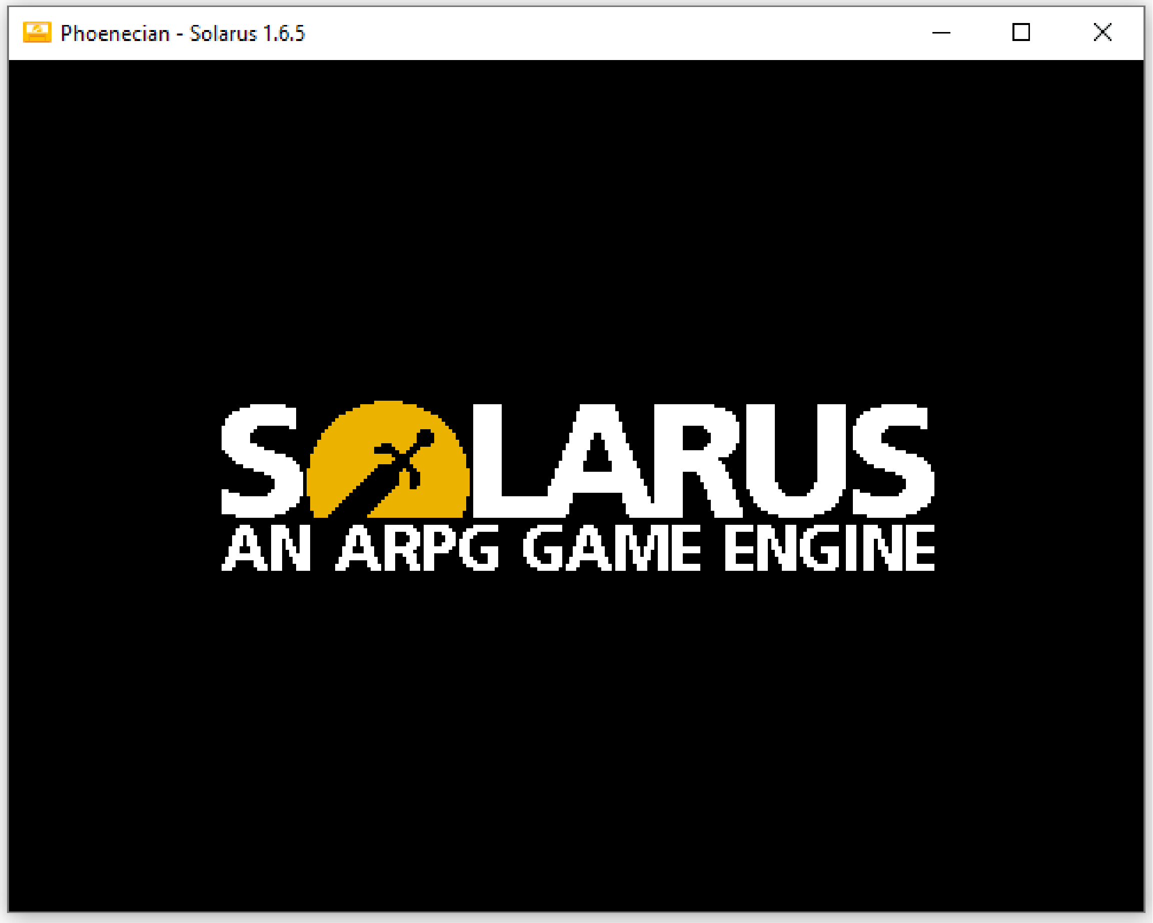


Figure 5.2.1: Solarus Title Screen on Game Load.

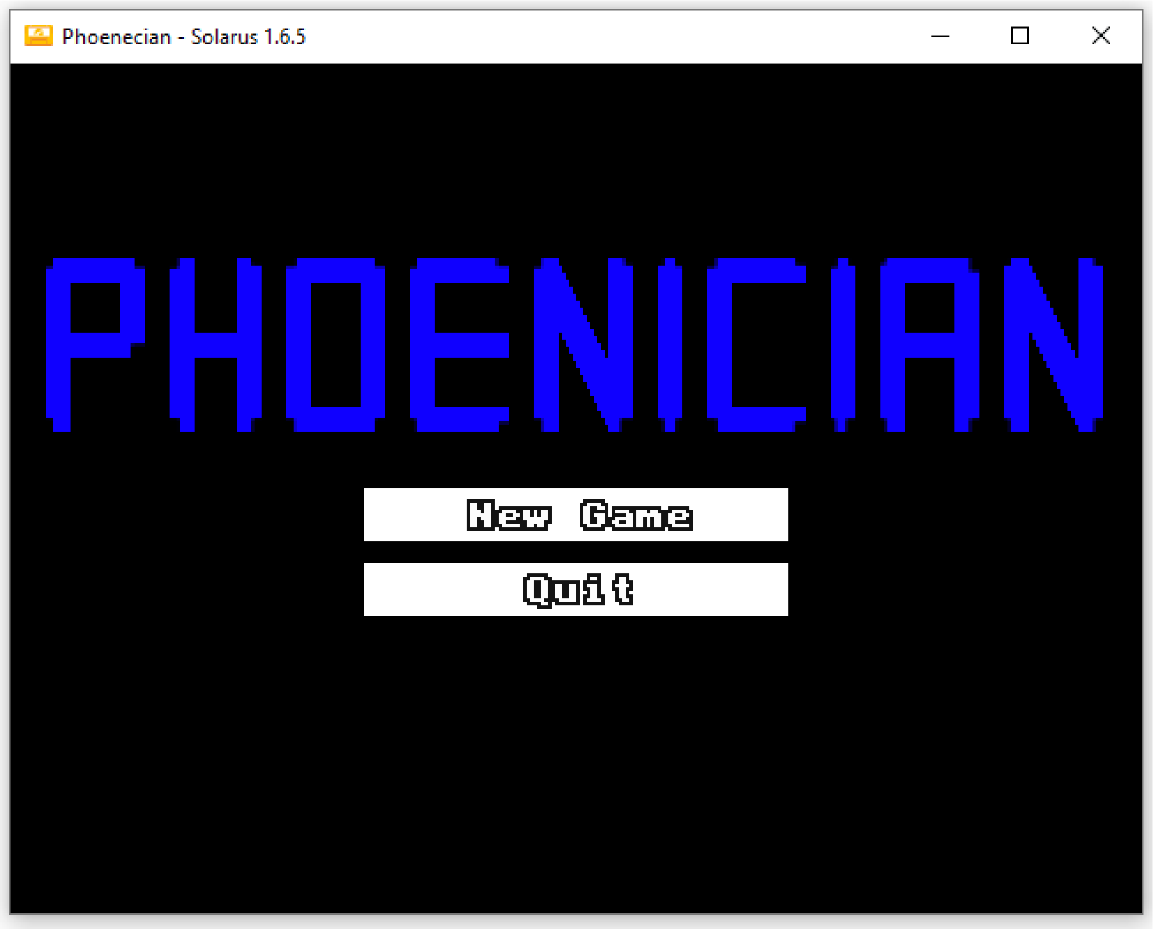


Figure 5.2.2: Phoenician Title Screen.

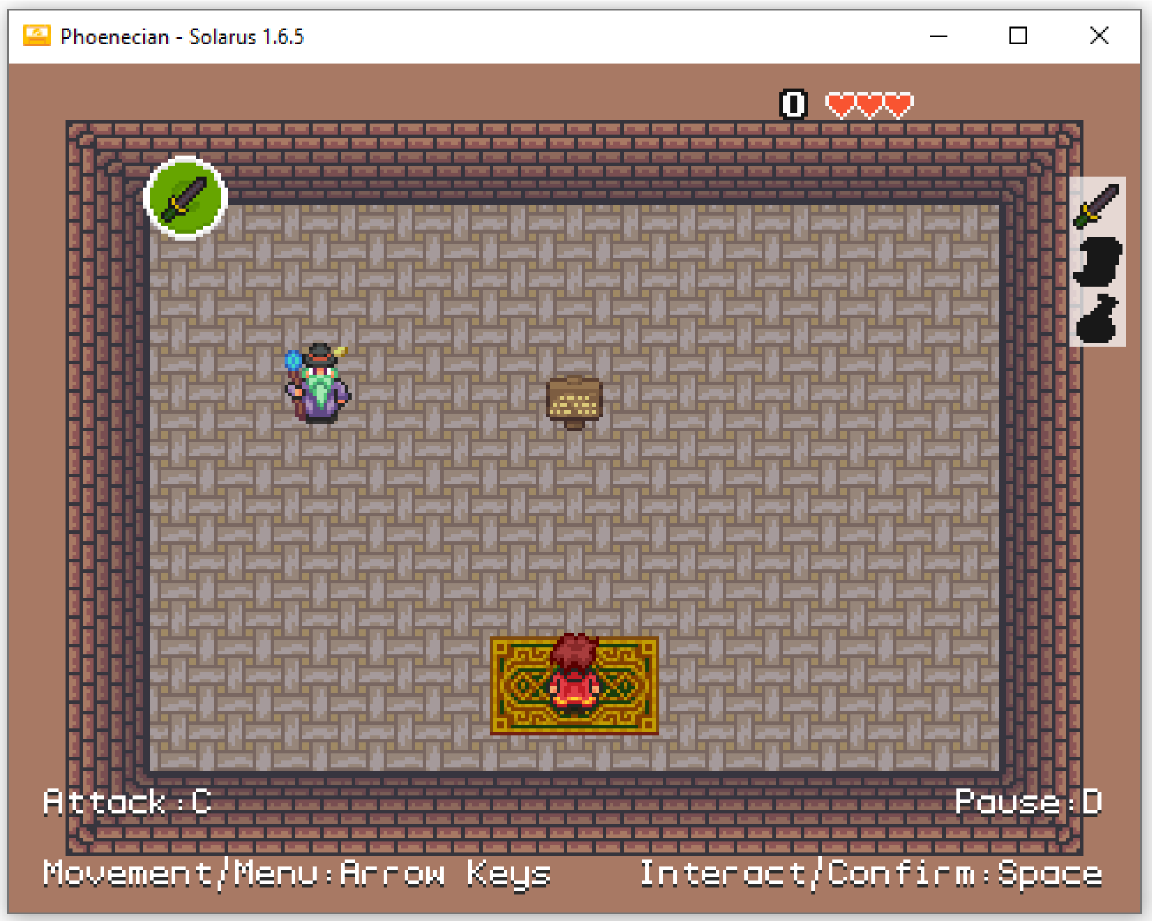


Figure 5.2.3: Main Demo Screen. Actions (Top-Left), Money/Health (Top-Right), Items (Right).

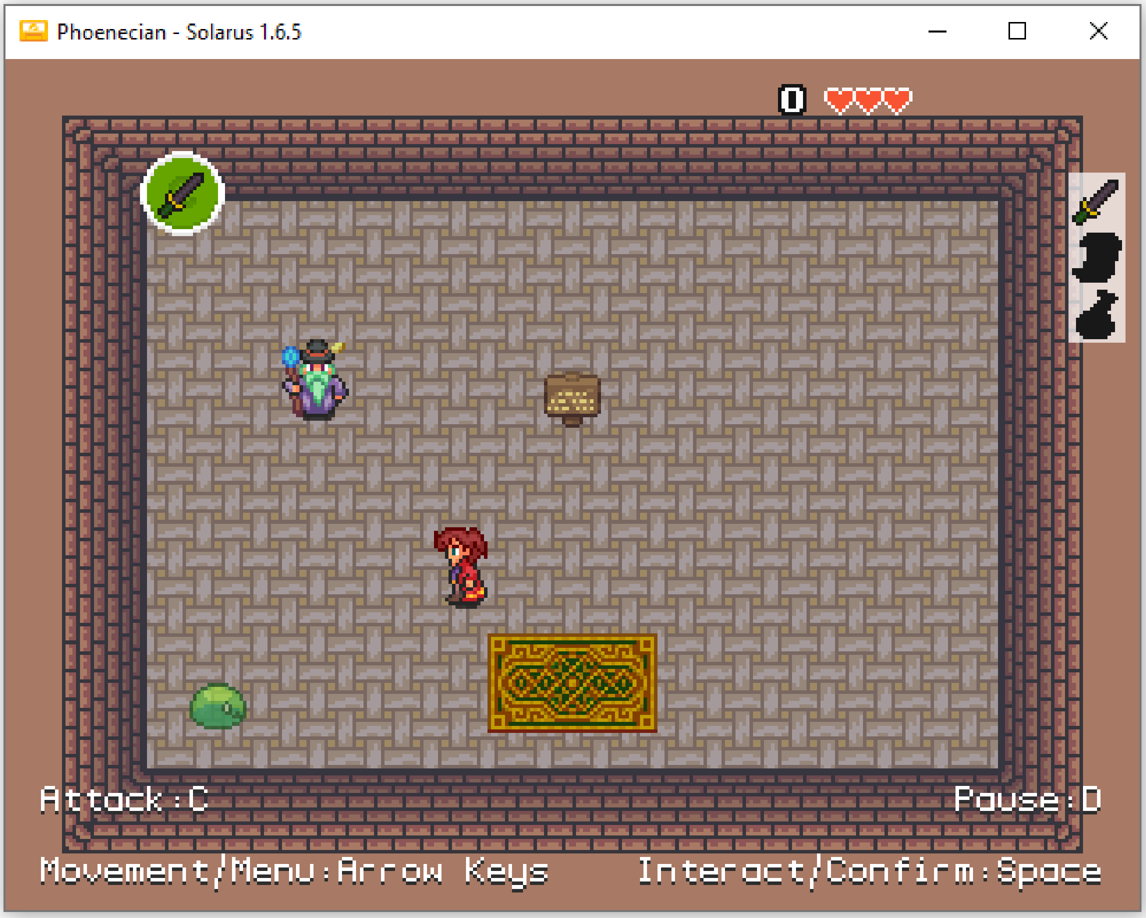


Figure 5.2.4: Main Demo Screen. Move left to attack the slime.

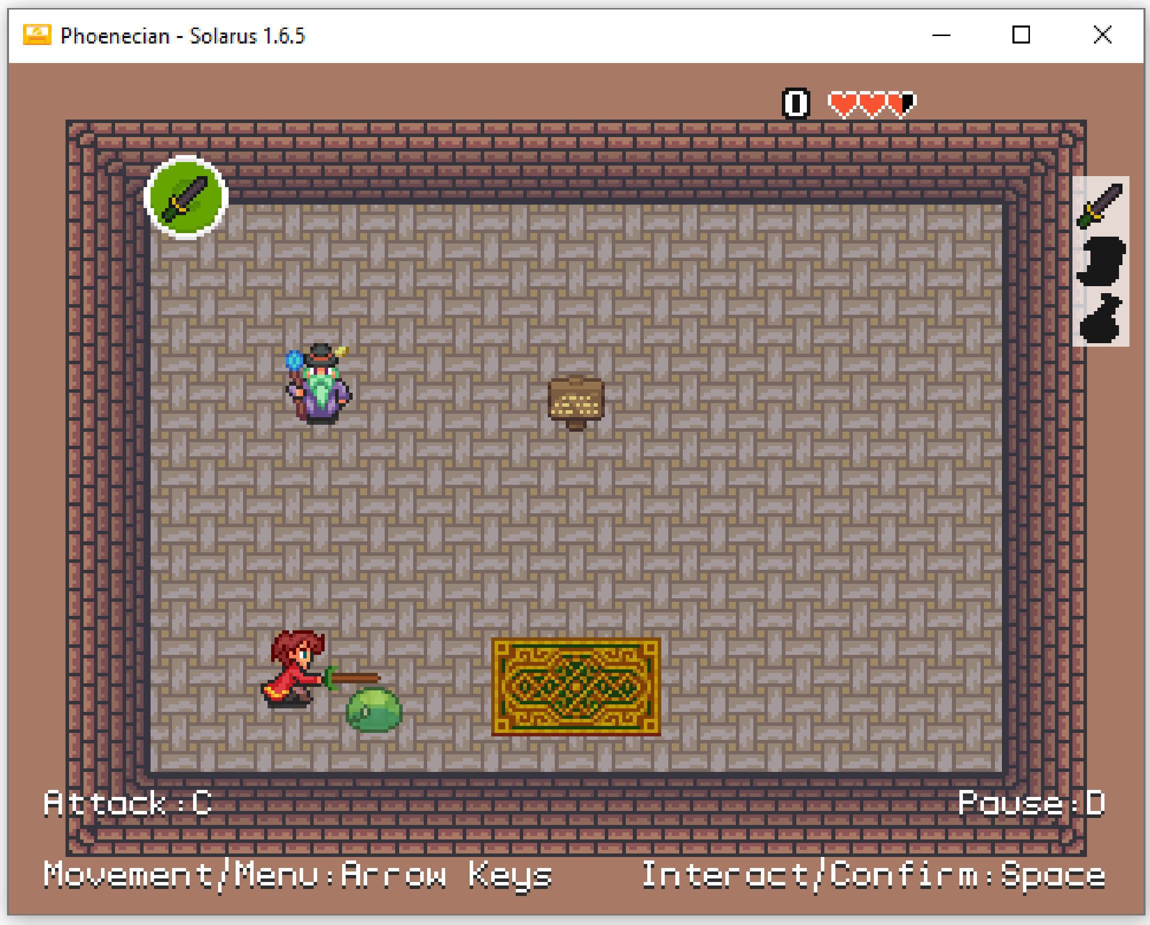


Figure 5.2.5: Main Demo Screen. Engage in combat with the slime using “C”.

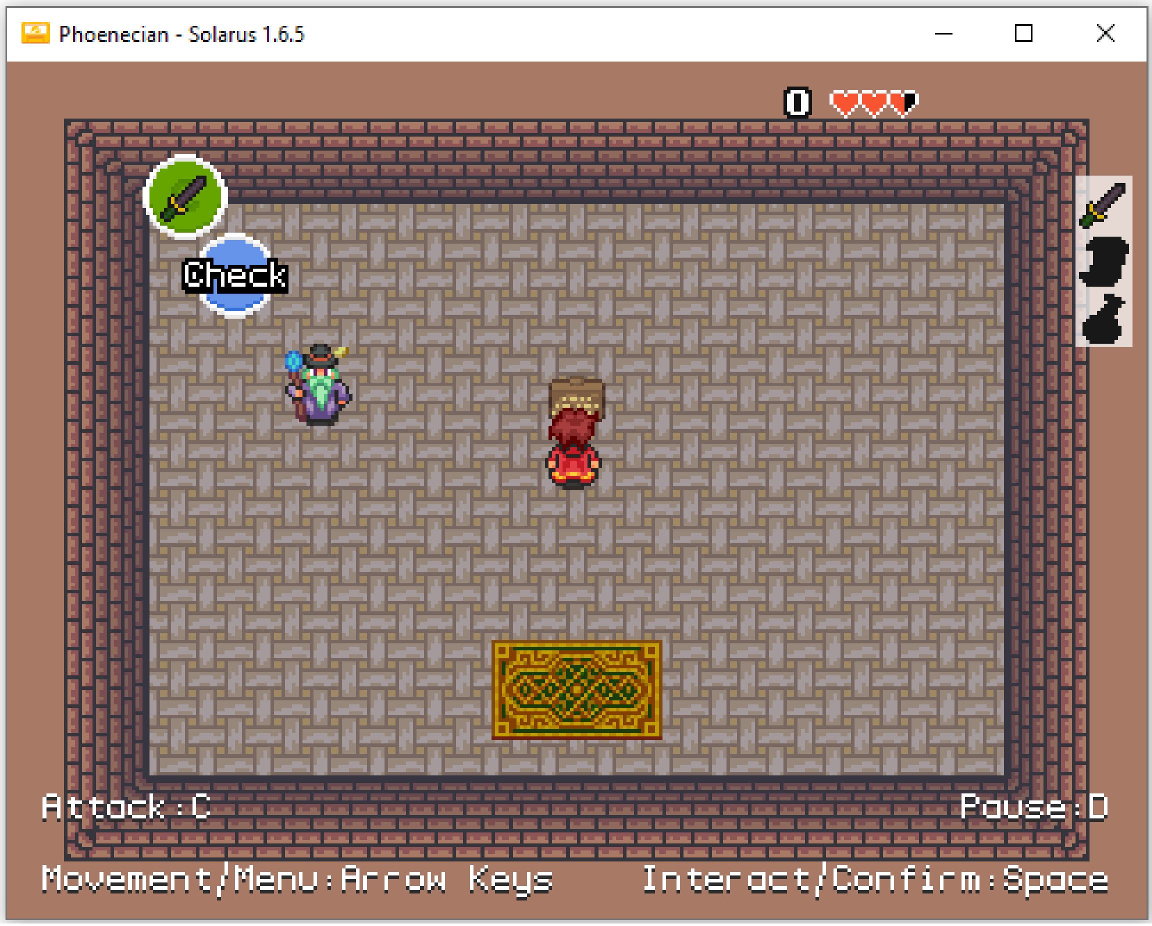
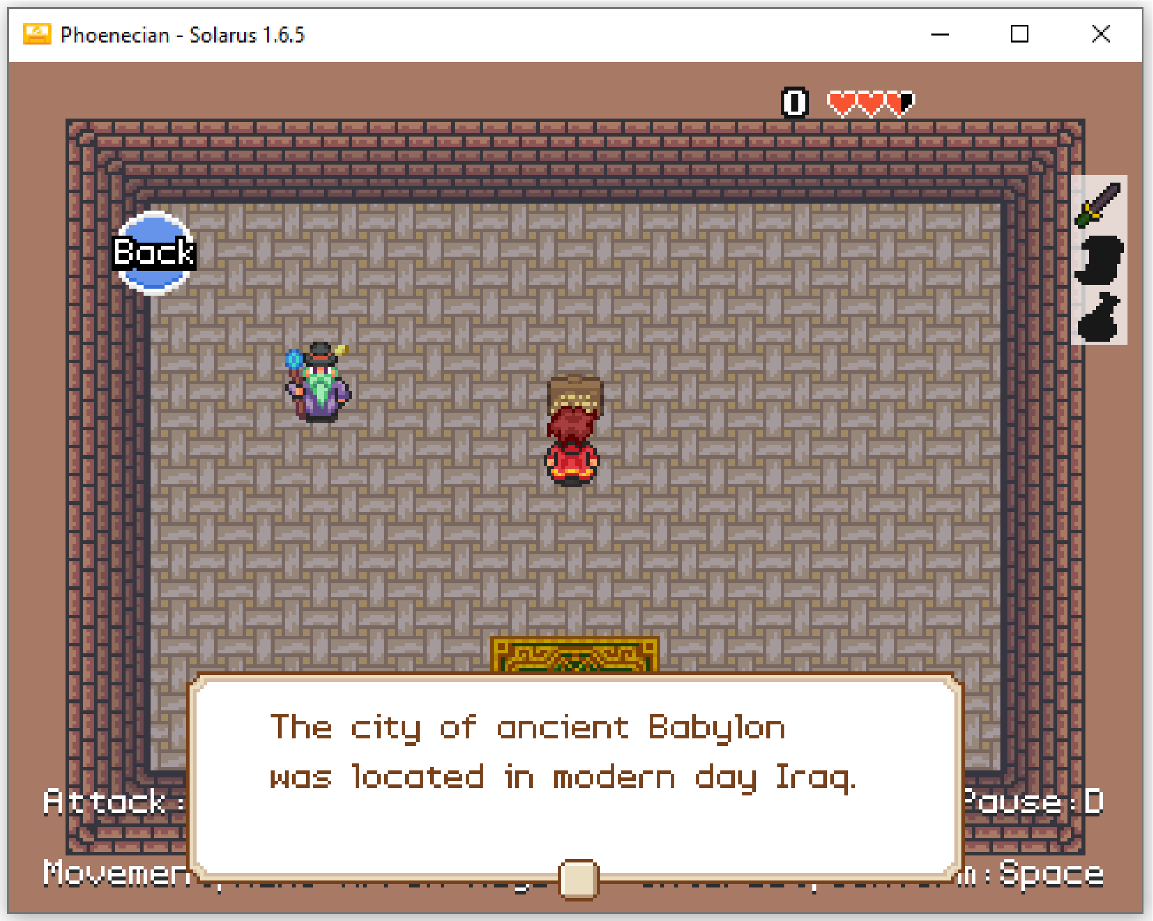
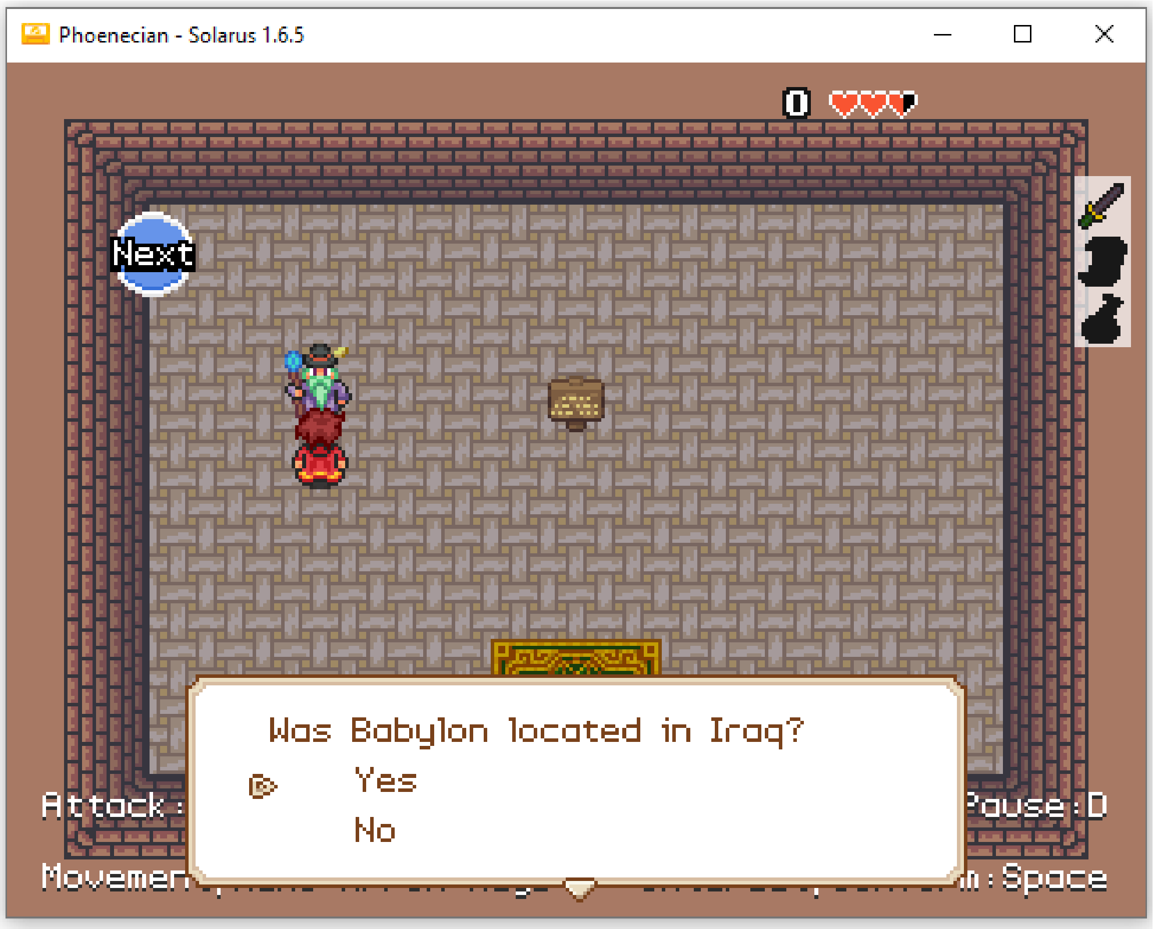


Figure 5.2.6: Main Demo Screen. Move to signpost and more Actions appear.



The city of ancient Babylon
was located in modern day Iraq.

Figure 5.2.7: Main Demo Screen. Signpost gives information about historical setting.



Was Babylon located in Iraq?

Yes

No

Figure 5.2.8: Main Demo Screen. Question meant to test history knowledge.

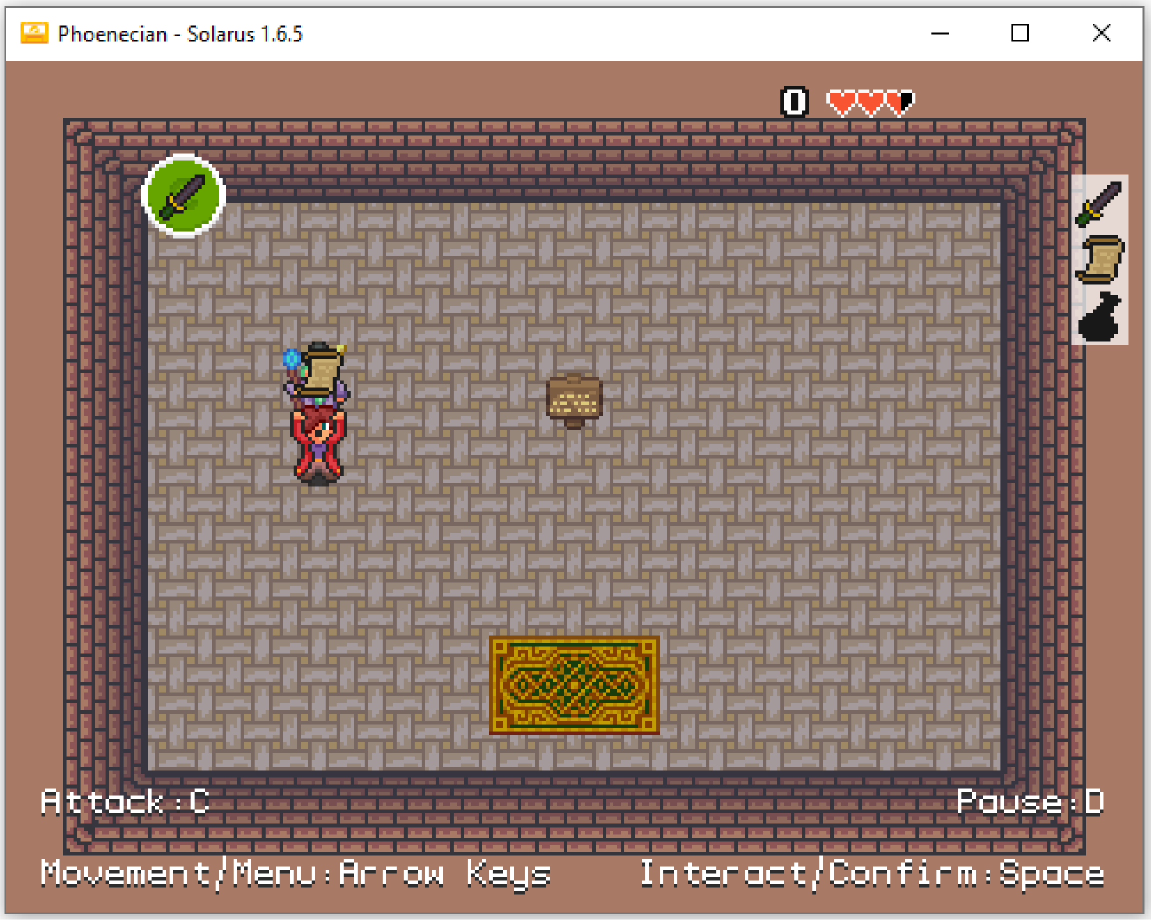


Figure 5.2.9: Main Demo Screen. Get item as a reward for answering correctly!

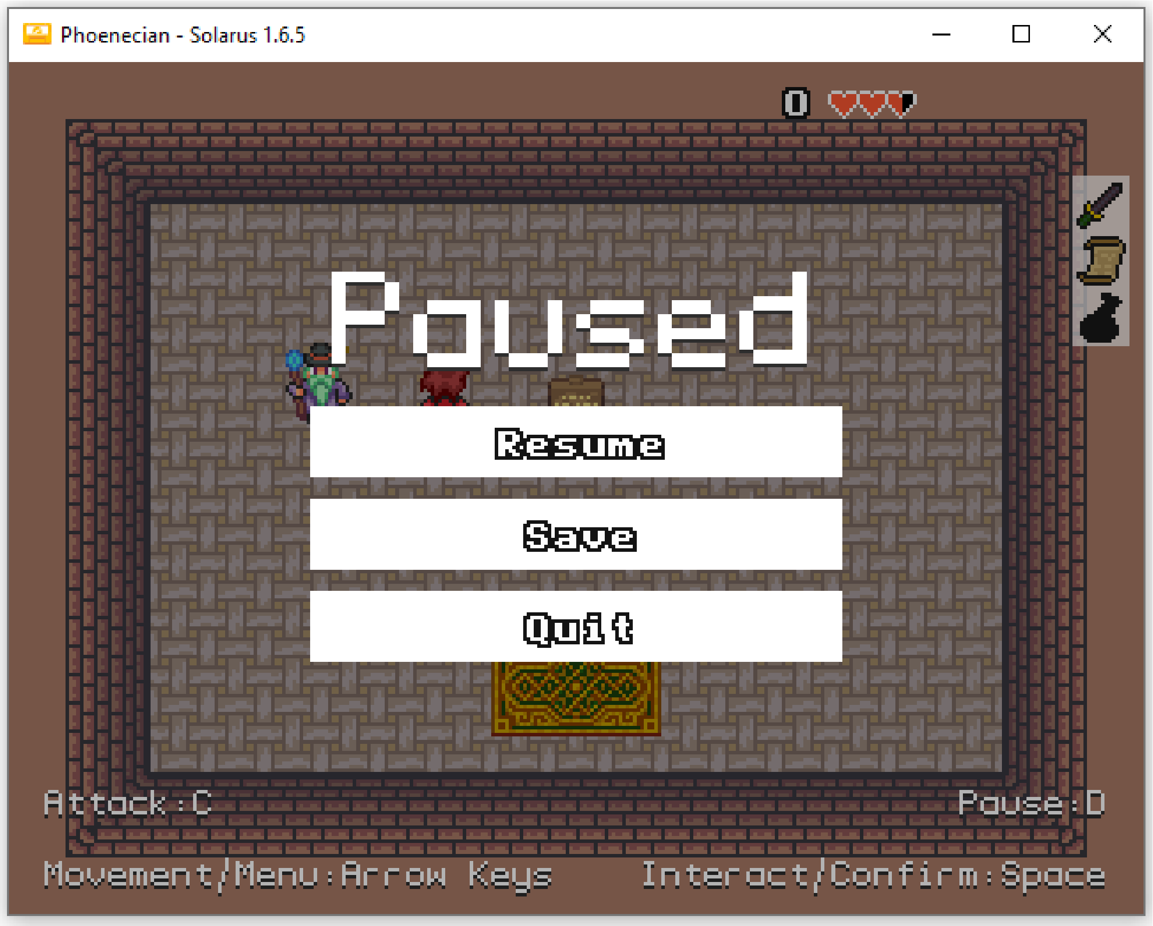


Figure 5.2.10: Main Demo Screen. Pause Menu.

6 References

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

[2] Team4.“ Phoenician.” Team4 Project Website,

https://rvekeria678.github.io/SoftwareEngineering_Team4_WEB/.

[3] 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.

[4] Solaris Project. Christolpho. https://www.solarus-games.org/. 2021.

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.