CS2113T SOFTWARE ENGINEERING AND OBJECT-ORIENTED PROGRAMMING

Semester 1 2019/2020

Group F14-2
Farmio User Guide

Authors:

Name	Matric Number
Tan Yeh Han John	A0182665W
Tan Zheng Wen	A0183615A
Tay Jing Xuan	A0185096U
Yew Jing Hui	A0184187U
Umar Moiz	A0180913H

About this Document

Purpose

This user guide is intended to introduce the interface, functionalities and various commands that you can use during gameplay.

Audience

This user guide is intended, in general, for any user who wants to learn about the game functionalities and in-game commands. In particular, this user guide is meant for:

- Teachers and parents who wish to teach their children to play the game
- Children who are playing the game
- Developers who wish to see the commands that players are able to use

Note the following symbols and formatting used in this document:

	This symbol indicates important information
`start`	A grey highlight indicates that this is a command that can be typed into the command line and executed by the game
Enter	Grey text with no highlight indicates that this is a keypress

1. Introduction

1.1. About Farmio

Farmio is a game that targets children aged between 11 and 14. We aim to teach children basic computing methodology in a fun and interactive way. Through this gameplay, students are exposed to problems that can be solved using programming concepts such as “if-else” statements, “for” and “while” loops.

What makes Farmio special is that the students will be able to see the execution of the code blocks that they have written. The game will simulate what happens on the farm according to the code blocks that the students have written. Students can watch and see whether their code blocks function as intended, and make any corrections if needed.

Interested? Jump to Section 2**,** “Starting a Game” to get started. Enjoy!

1.2. Purpose of Farmio

At the Ministry’s work plan seminar held on 10 July 2019, Singapore’s Minister for Communications and Information, S Iswaran, announced that coding will be compulsory for Upper Primary students through a program called Code For Fun which will be introduced by 2020.

Farmio aims to help these groups of students to aid them in learning the fundamentals of programming. Through interactive gameplay, we hope that Farmio can help children who are new to programming understand and learn to apply the basics of programming while having fun.

2. Getting Started

Before launching the game, ensure that the latest version of Java software installed on the computer on which you are running the game.

It is recommended to use a Linux based terminal for the best gaming experience.

Your user experience may differ from the images shown below if you are using the Windows command line terminal. Take note that colors can only be seen using a Unix-based terminal, such as Linux and Mac. However, the game commands will function as described.

To start the game, open terminal console, and type `java -jar` followed by the location of the game file that ends with ‘jar’ file extension.

```
java -jar your/file/path/farmio.jar
```

Figure 2.1 - Launch Farmio Command

2.1. Quick Start Guide

This guide serves as a summary of all the in-game commands for Farmio.

Game Creation Commands

`new game` - starts a new game

`load game` - loads previously saved games.

`quit game` - Quits the game

Story Segment Commands

`skip` - skip narratives

`Enter` - to proceed with the narrative.

Interactive Segment Commands

`do ACTION` - creates a new DoTask

`if CONDITION do ACTION` - creates a new IfTask

`insert POSITION TASK` - inserts a new task at the specified position

`edit POSITION TASK` - replaces the current task at the position with your new task

`delete POSITION` / `delete all` - deletes the task at the specified position / all tasks

`hint` - displays hints for the level

`menu` - displays the in-game menu

Simulation Segment Commands (Enter these commands only when prompted)

`reset` - resets the level

`Enter` - proceeds to the next day if there are no errors, otherwise, resets level

Feedback Segment Commands (Enter these commands only when prompted)

`Enter` - proceeds to the next level if objectives are completed, otherwise, resets level

2.2. Game Console Navigation Guide

This guide serves as a quick orientation to Farmio's Command Line Interface. The figure below serves as an example of the game console during gameplay.

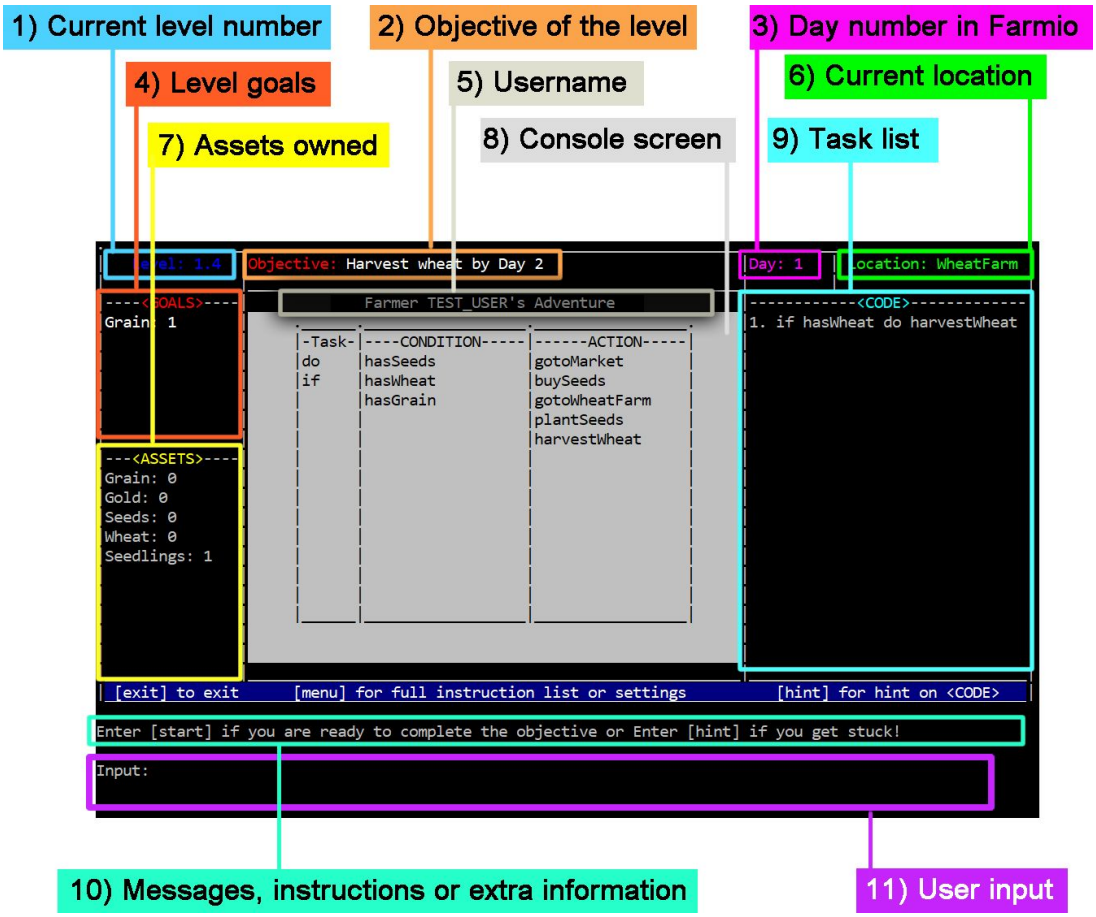

Figure 2.2.1 Example of Farmio's Command Line Interface

1. **Current level number:** Shows your current level. In Figure 2.2.1, the current level is 1.4.
2. **Objectives of the level:** Shows what must be done to complete the level as well as the level deadline. In Figure 2.2.1, this deadline is day 2. If it is not stated, the objective has to be completed after the task list is executed once.
3. **Day number in Farmio:** Shows the current day. In Figure 2.2.1, the current day is 1.

4. **Level goals:** Shows the number of assets required for you to complete the objective. From Figure 2.2.1, you need 1 grain at the end of day 2 to pass the level.

5. **Username:** Shows the username you entered on game creation. In Figure 2.2.1, it is ‘TEST_USER’

6. **Current location:** Shows the current location of the farmer. In Figure 2.2.1, it is ‘WheatFarm’.

7. **Assets Owned:** Shows a list of your assets owned. When the task list is executed, assets owned may change. In Figure 2.2.1, only 1 seedling is owned.

8. **Console screen:** Shows a visual representation of commands or tasks when executed. In Figure 2.2.1, It can also show other important information such as the list of useful task types, conditions, and actions you can use to solve the current level.

9. **Task list:** Shows you all the tasks you have added. The task in the task list executes in ascending order either after the user starts the simulation or at the start of each day. In Figure 2.2.1, there is only one task; ‘if hasWheat do harvestWheat’.

10. **Messages, instructions or extra information:** Shows you acknowledgments of commands, instructions, the narrative of each level and other messages.

11. **User input:** Shows your input.

3. Gameplay

Farmio is a game centered around the story of a Farmer. In each level, a narrative and objective will be presented to you. You will have to use the computational concepts taught to create a set of tasks that can fulfill the objectives. But beware - you cannot change the task list once the Farmer's day has started! So put on your thinking caps, and create a task list that will help the Farmer fulfill his objectives!

3.1 Game Segments

Farmio is split into five main segments:

1. **Game Creation**

   This segment occurs once when you start up the game and prompts you to either create a new game or load a previous save.

2. **Story Segment**

   At the start of every level, the Story Segment will continue the narrative, introduce new concepts and present your objectives for the level.

3. **Interactive Segment**

   In this segment, you can create and edit tasks for the Farmer.

4. **Simulation Segment**

   This segment will simulate the execution of your code, and after your task list has been completed, you can choose to either continue to the next day or restart the level

5. **Feedback Segment**

   After the objectives or deadlines have been met, this segment will recap the level, as well as give any hints if you have failed the objectives.

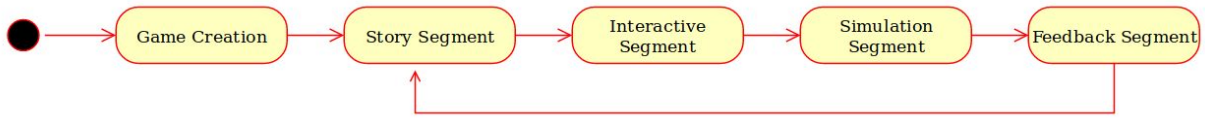

Figure 3.1.1. Game Segments

3.2 Game Creation

3.2.1. Starting a New Game

Step 1: Open terminal console, and type `java -jar` followed by the location of the game file that ends with ‘jar’ file extension.

```
java -jar your/file/path/farmio.jar
```

Figure 3.2.1.1. Launch Farmio Command

Step 2: Upon the successful game launch, you see a welcome message.

Figure 3.2.1.2. Welcome Screen

Step 3: Press `Enter` to see the menu.

Step 4: Next you will see the game menu.

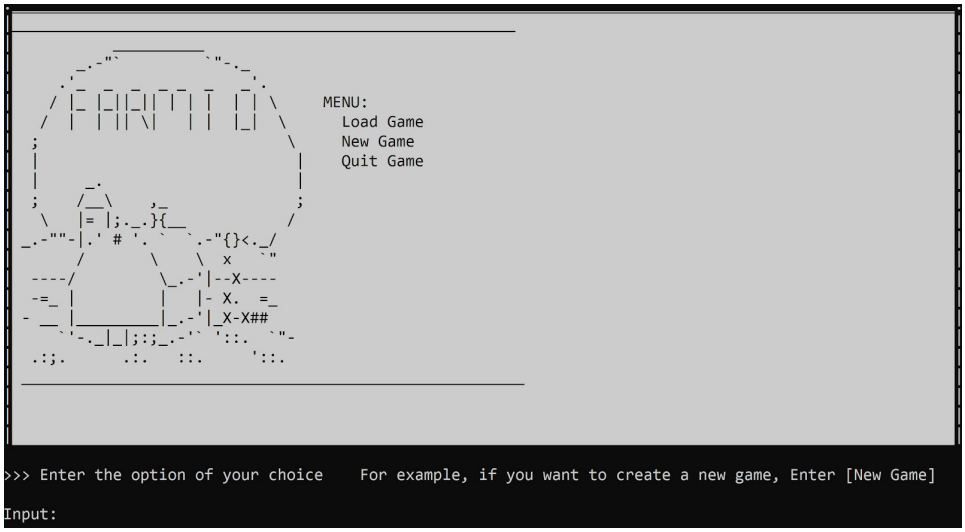

Figure 3.2.1.3. Start game menu

Step 5: Type `new game` and press `Enter` to start a new game.

You can also enter the `new game` command at the welcome screen.

Please ensure that you are in a non-permission restrictive folder before running the game to allow the game to create and save game data in the folder.

Load game option in Figure 2.3 will only appear if the game save file is in the same folder in which the program is running.

3.2.2. Loading a Saved Game

Step 1: Open terminal console, and type `java -jar` followed by the location of the game file that ends with ‘jar’ file extension. As shown in Figure 3.2.1.1.

Step 2: Upon the successful game launch, you will see a welcome message, as shown in Figure 3.2.1.2.

Step 3: Press `Enter` to proceed.

Step 4: Next you will see the game menu, as shown in figure 3.2.1.3.

Step 5: Type `load game`, and press `Enter` to start a previously saved game session.

> `load game` command will only function properly if the game save file is in the same folder as the folder the game is running from and if the save file is not corrupted.
>
> You can enter the `load game` command in the welcome screen as well.

3.2.3. Adding Your Name

Step 1: After creating a new game, the game will prompt you to enter a name for your farmer.

Figure 3.2.3. Entering Farmer Name

Your name cannot be more than 15 characters long, empty, ‘menu’, or contain any special characters with the exception of ‘_’.

3.3. Story Segment

This section discusses the story segment, as well as the commands available in this segment.

3.3.1. Narratives

The game enters a story segment at the start of every level. In this stage, there are two available commands

- `Enter` - Move to the next frame in the narrative
- `skip` - skips to the end of the narrative

For example, when a new game has been created and you have input your username, the game will prompt you to either press `Enter` or enter `skip`.

Figure 3.3.1.1. Game options to begin or skip the tutorial

If you choose to press `Enter`, the game will then go through a short tutorial explaining the game rules. You would then have to press `Enter` go through the set of instructions.

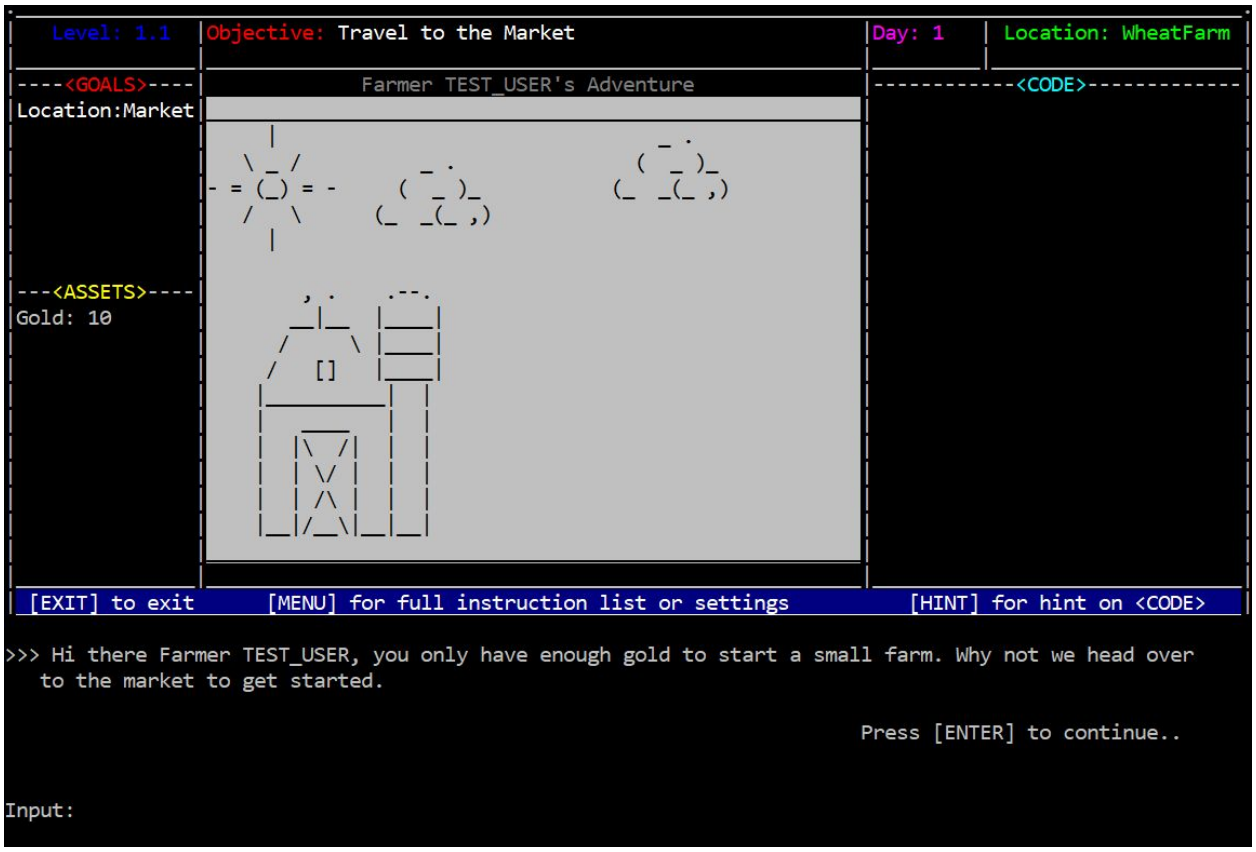

Figure 3.3.1.2. The game tutorial

You can choose to use the skip command at any point of a narrative to skip to the end of the narrative. Otherwise, you can press `Enter` to view the next frame.

If you entered the `skip` command instead, the game allows you to skip the narratives which are being displayed. This saves you time if you have already seen them.

Figure 3.3.1.3. Using the `skip` Command

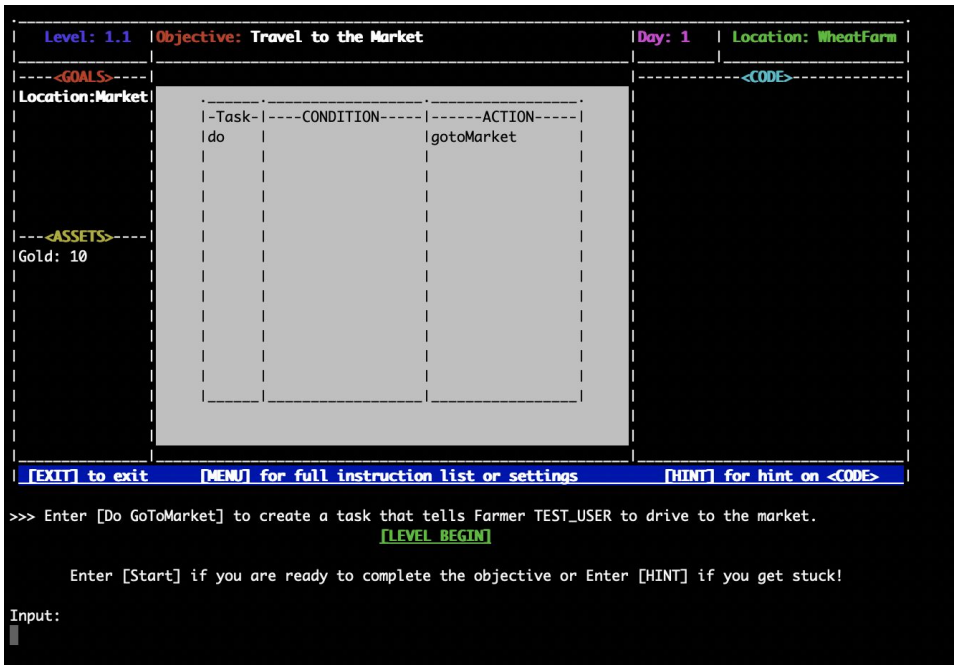

Figure 3.3.1.4. Skipping to the end of the narratives

3.4. Interactive Segment

The interactive segment comes after the story segment. This stage of the game allows you to write code to create tasks to fulfil the level objectives. This section describes the commands that can be used during the interactive segment.

3.4.1. Creating a New Task

Creates a new task and adds it to the bottom of your task list. There are two types of tasks:

1. **DoTask**

   Farmer always executes the task

2. **IfTask**

   Farmer executes the task only if the condition is fulfilled

Creating a DoTask:

Format: `do ACTION`

Examples of valid DoTask creation commands:

- `do sellGrain`
- `do buySeeds`

Creating a IfTask:

Format: `if CONDITION do ACTION`

Examples of valid task creation commands:

- `if hasWheat do harvestWheat`
- `if gold greater than or equals 10 do buySeeds`

You can only have a maximum of 18 tasks in your task list

The command is case insensitive

For example, to create a task for the farmer to go to market, you would enter

`do goToMarket`

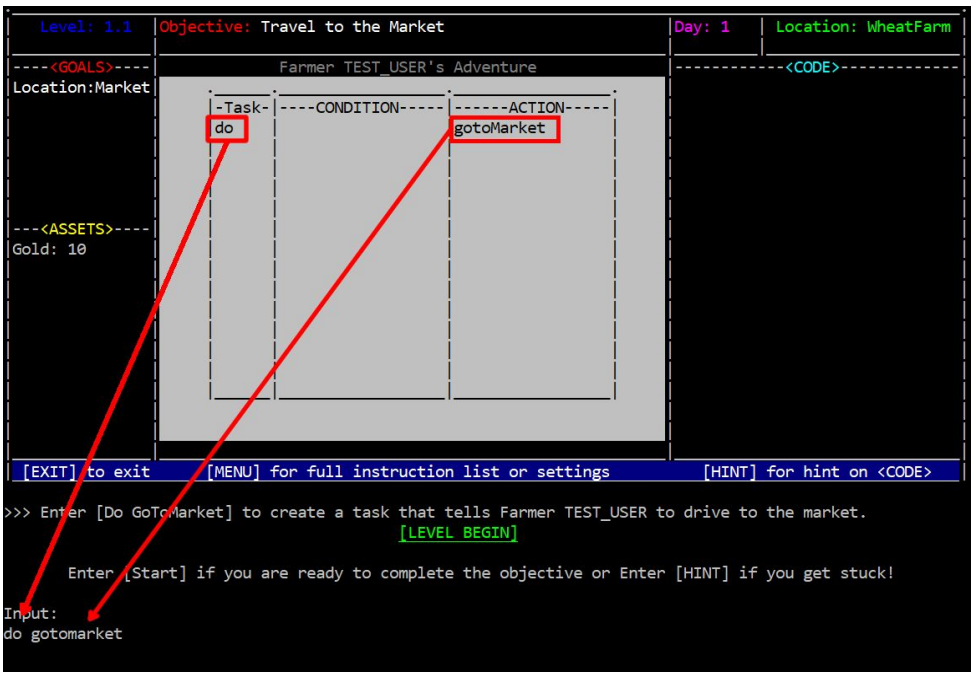

Figure 3.4.1.1. Creating a New Task

After the task has been successfully created, it will be added to the `CODE` section.

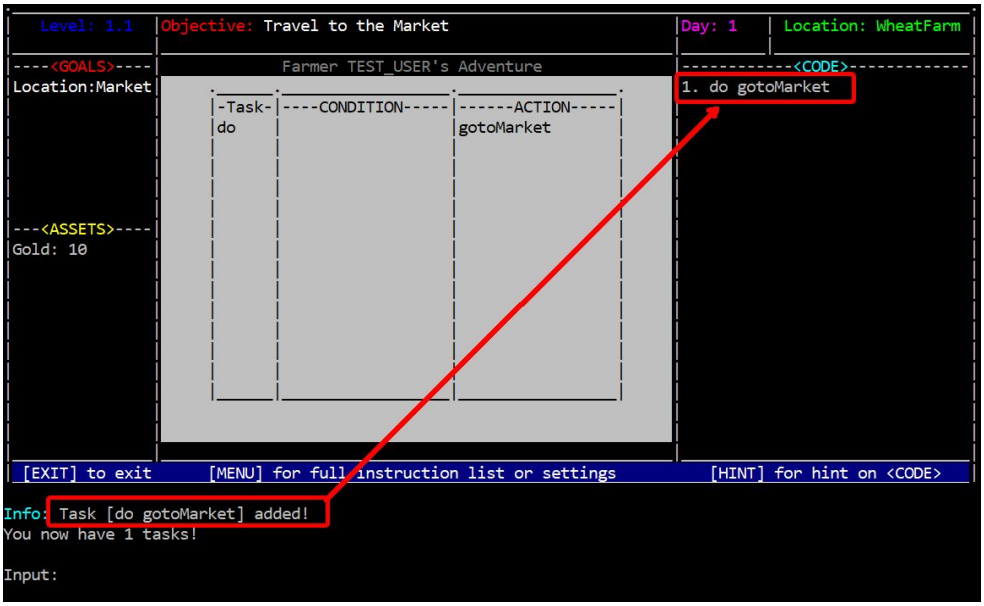

Figure 3.4.1.2. Successful Creation of Task

3.4.2. Inserting a New Task

Inserts a new task into a specified position in your task list.

Format: `Insert POSITION TASK`

> `POSITION` Must be a number between 1 to (size of task list + 1)
>
> You cannot insert more tasks if you already have 18 tasks in your task list

Examples of valid commands for a tasklist with 2 tasks:

- `insert 2 do gotoWheatFarm`
- `insert 3 if hasSeeds do plantSeeds`

For example, if you want to insert a task to go back to the wheat farm before planting seeds, you would enter the following command:

`insert 2 do gotoWheatFarm`

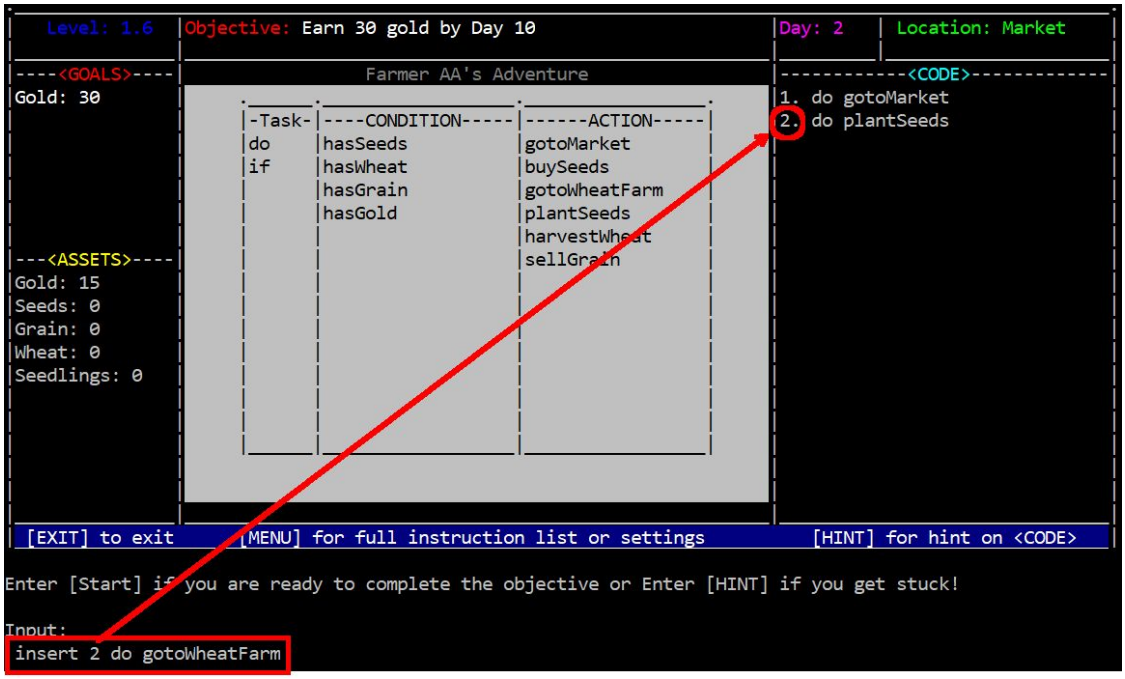

Figure 3.4.2.1. Inserting a new task at position 2

After the command is executed, your new task will be added at position 2. Tasks that were previously at position 2 and after will be pushed behind.

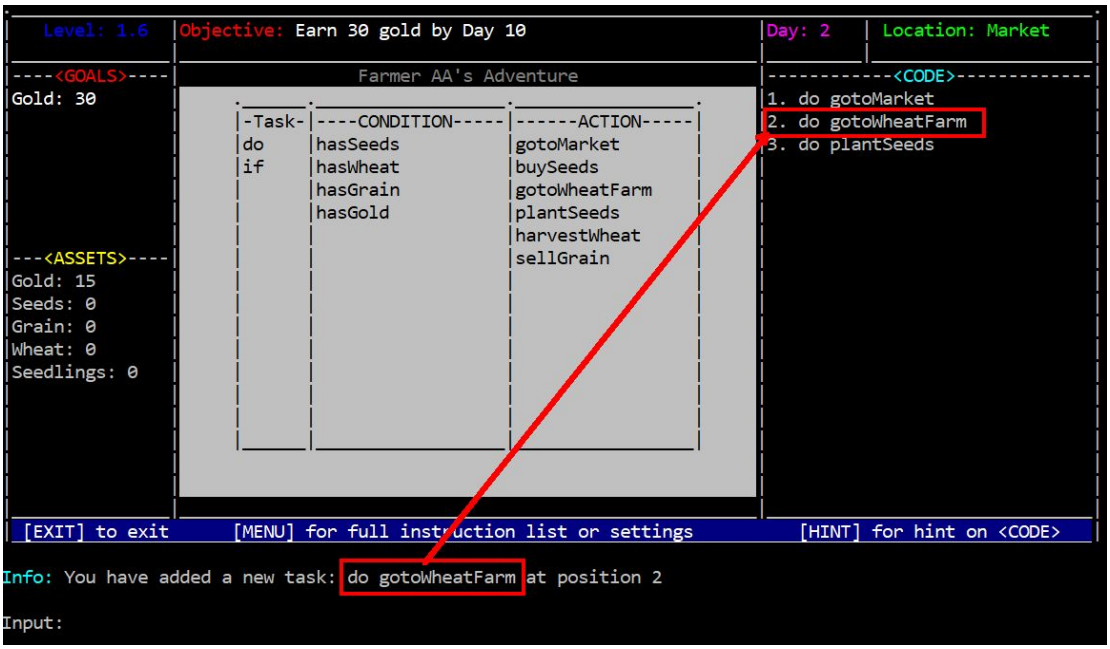

Figure 3.4.2.2. Successful insertion of a new task

3.4.3. Editing an Existing Task

Edits a task inside the task list.

Format: `edit POSITION TASK_DESCRIPTION`

> ⓘ `POSITION` Must be a number between 1 to the size of the task list

Examples of valid commands for a tasklist with 3 tasks:

- `edit 2 do buySeeds`
- `edit 3 if hasSeeds do plantSeeds`

For example, if you want to edit the task at position 2 to `do buySeeds`, you would enter:

`edit 2 do buySeeds`

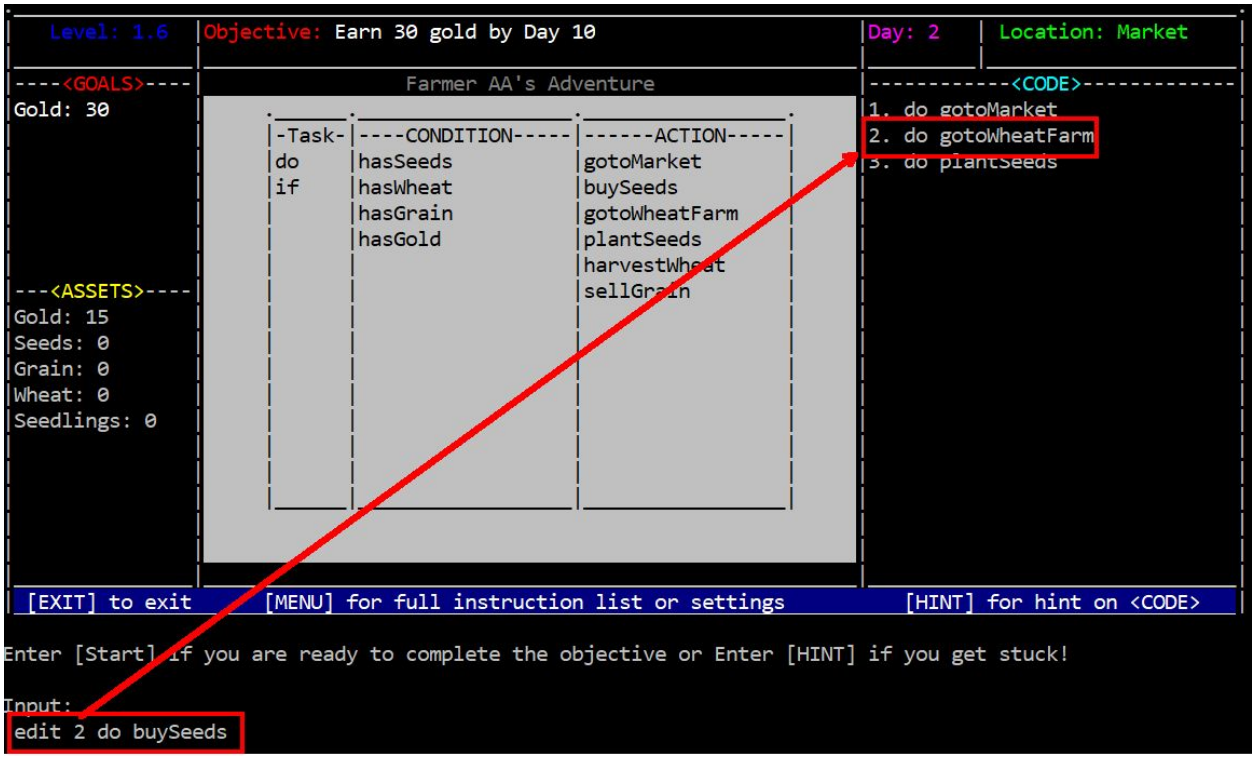

Figure 3.4.3.1. Editing a task in the task list

After the command is executed, the task at position 2 has been edited to `do buySeeds`.

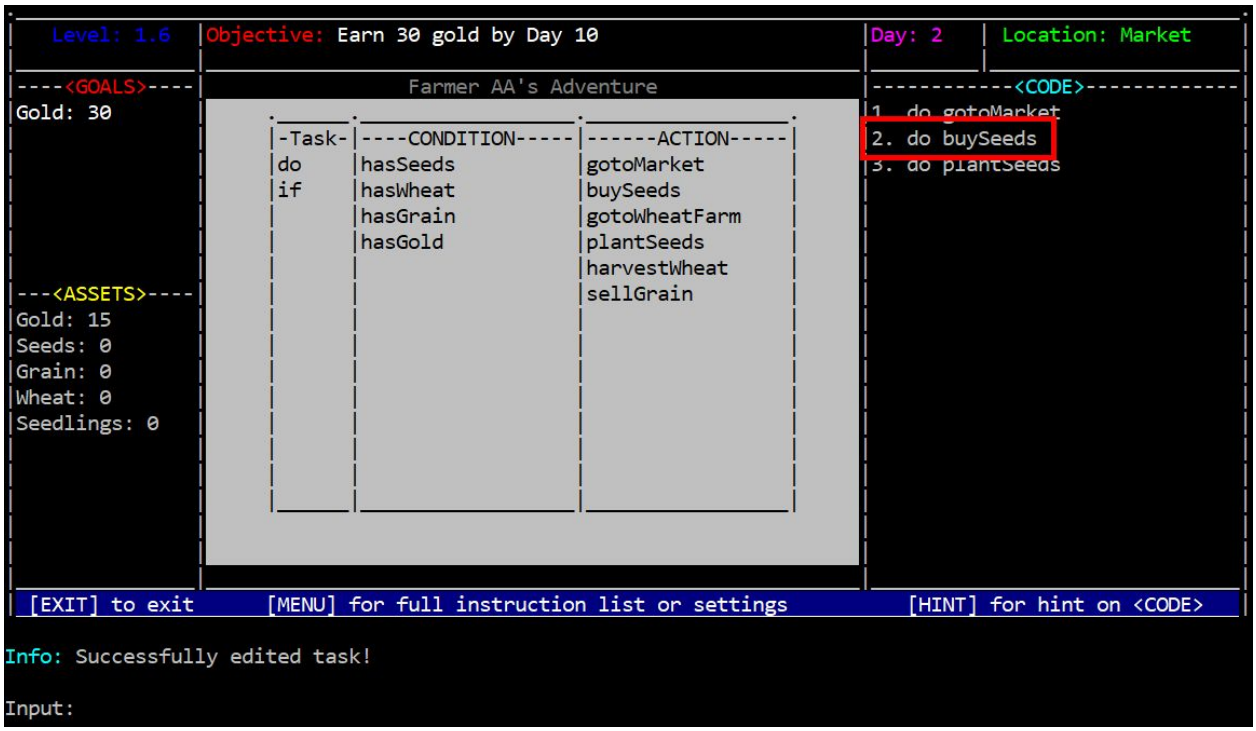

Figure 3.4.3.2. Successful editing of a task

3.4.4. Deleting existing Tasks

Delete one or many tasks in the task list.

Delete one task:

Format: `delete POSITION`

> `POSITION` Must be a number between 1 to size of task list

Delete all tasks:

Format: `delete all`

For example, to delete the task at position 3, you would enter:

`delete 3`

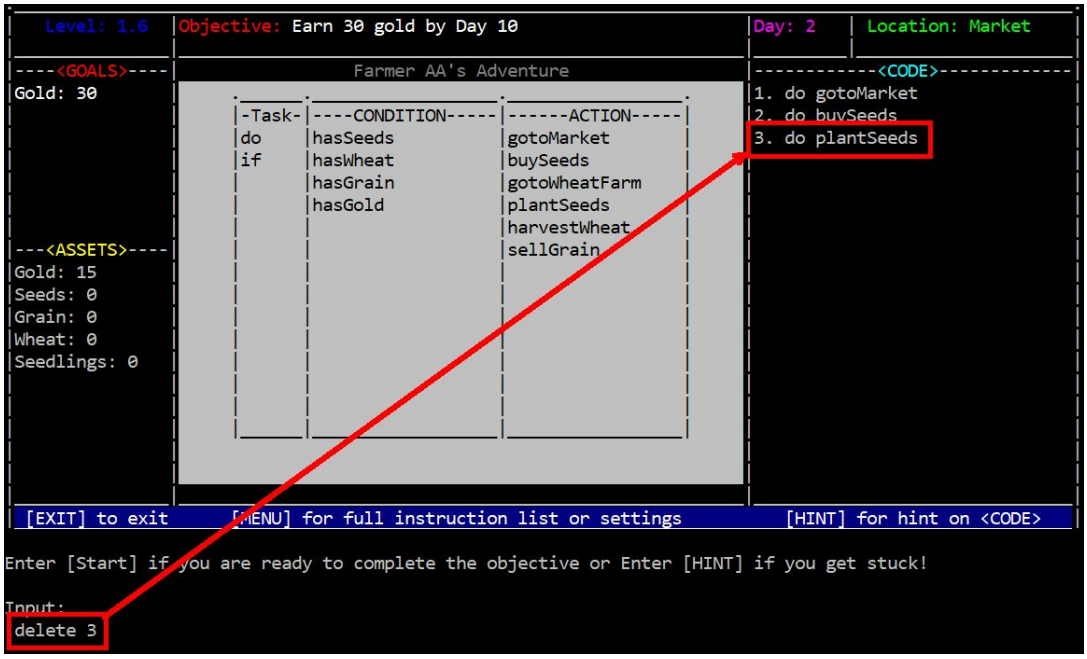

Figure 3.4.4.1. Command to delete a single task

After the command has been executed, the specified task will be removed.

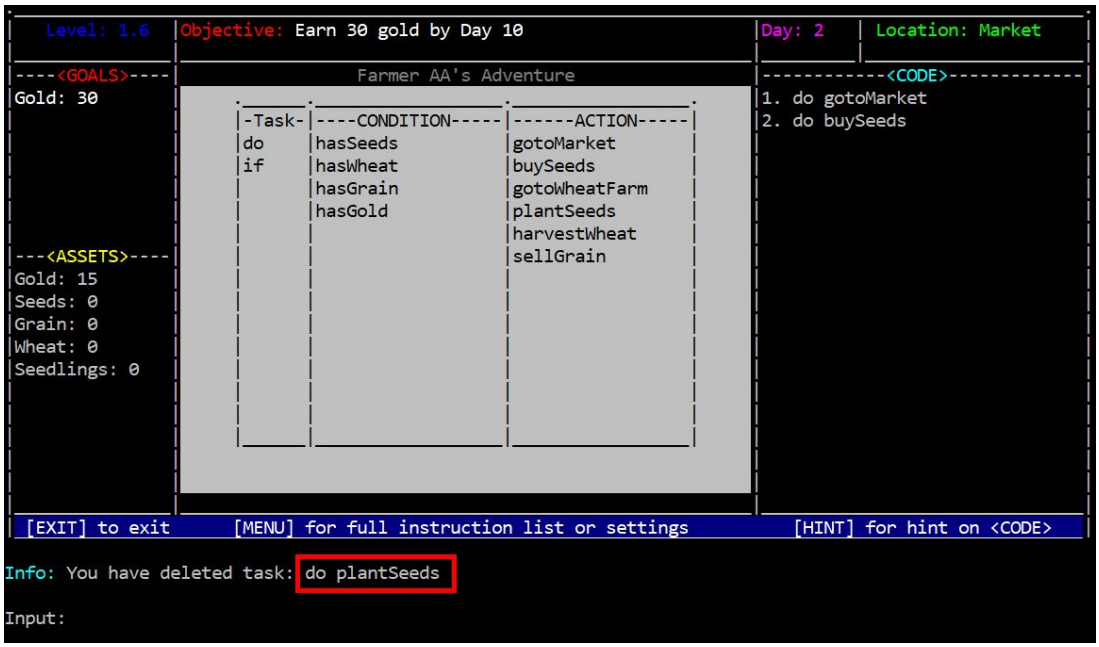

Figure 3.4.4.2. Successful deletion of the task

3.4.5. Hints for the level

You may enter `hint` in the terminal to look up for any hints for that level.

Format: `hint`

- During the tutorial (levels 1.1 to 1.4), the hint also serves as the model solution.
- After the tutorial, hints will be less direct and not reveal the model solution.

For example, in figure 3.3.5.2, entering `do gotomarket` followed by entering `start` is sufficient to pass level 1.1. Hence, this is shown in the hint.

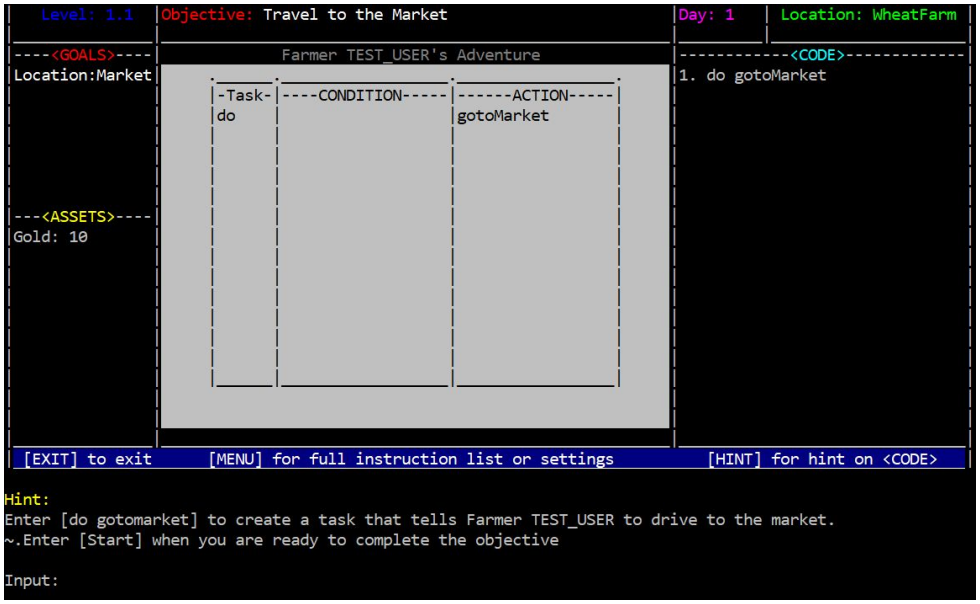

Figure 3.4.5.2. Show hint

3.4.6. Menu

The menu command allows you to access the in-game menu.

Format: `menu`

This in-game menu will show you a list of commands that can be executed to help you in the game. These commands will be discussed in detail from section 3.4.7 to section 3.4.10.

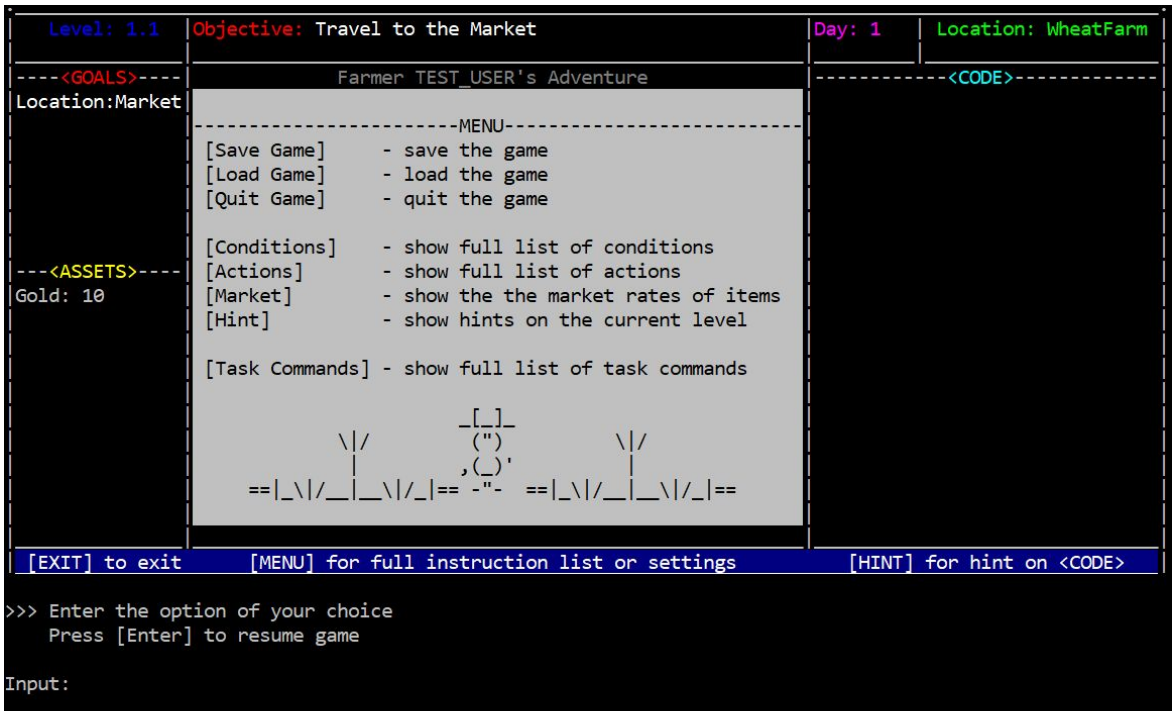

Figure 3.4.6.1. In-game menu

All commands displayed on the menu can be used at any point in the interactive segment. You do not need to go to the menu first.

3.4.7. Task Command List

The Task Command List allows you to view all task-related commands

Format: `task commands`

- As seen in Figure 3.4.7.1, these instructions are a summary of the commands documented from section 3.4.1 to section 3.4.4 of this user guide.

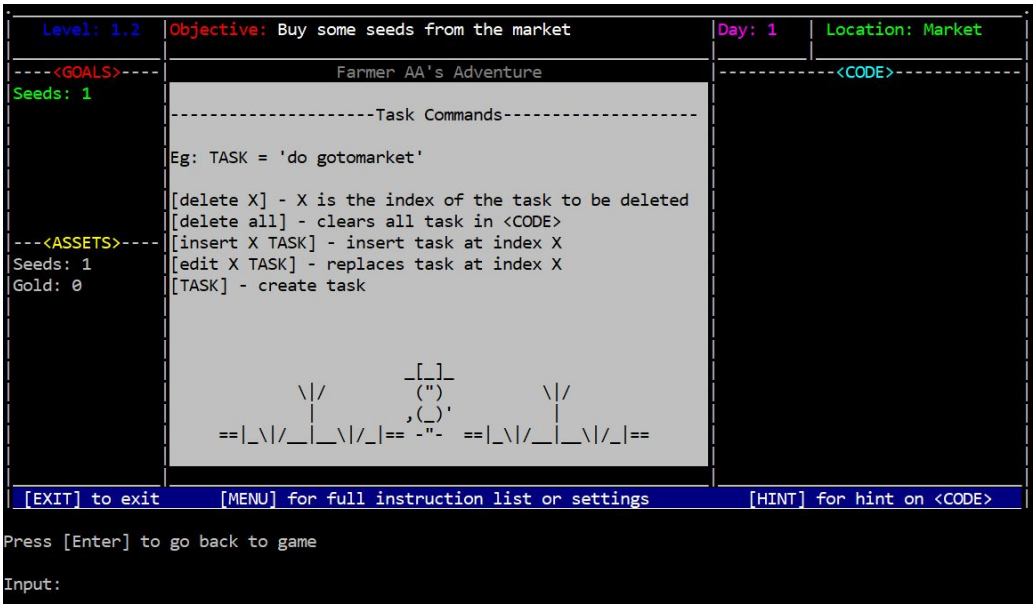

Figure 3.4.7.1. The Task Command List

3.4.8. ActionList

ActionList displays all the actions that can be used at the current level.

Format: `action`

Displayed action categories:

- Transportation actions. These actions can be used to change the location of the farmer at the current level. You can choose to enter input and create a task at the ActionList screen.
- Market actions that can only be used at the market. These actions mainly consist of buying and selling actions.
- Farm actions that can only be used at the WheatFarm. These actions mainly include planting seeds and harvesting wheat.

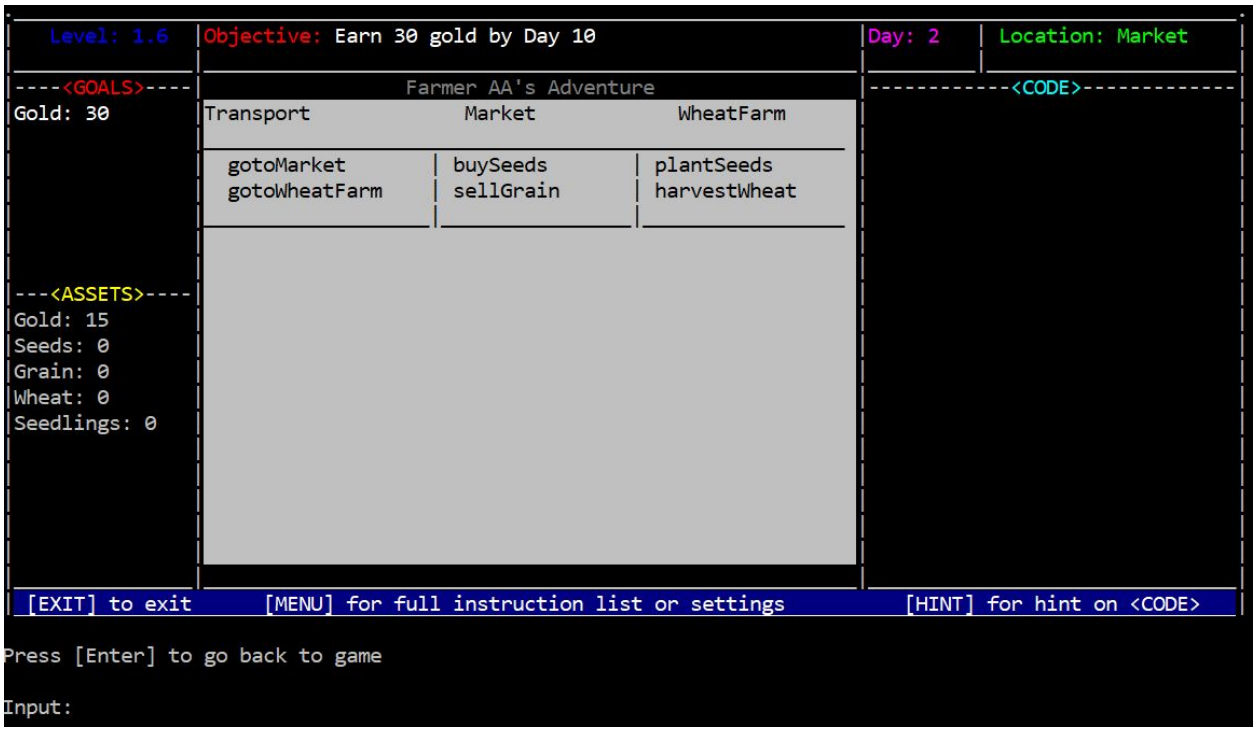

Figure 3.4.8.1. Action List

3.4.9. ConditionList

The Condition List displays all of the conditions that can be used at the current level.

Format: `condition`

Displayed condition categories:

- Gold conditions. The conditions listed in this segment can be used to compare the farmer's gold with a number. The conditions are case insensitive whitespace insensitive.
- Boolean conditions. These conditions are either true or false. The conditions listed in this segment can be used to complete objectives in the current level.

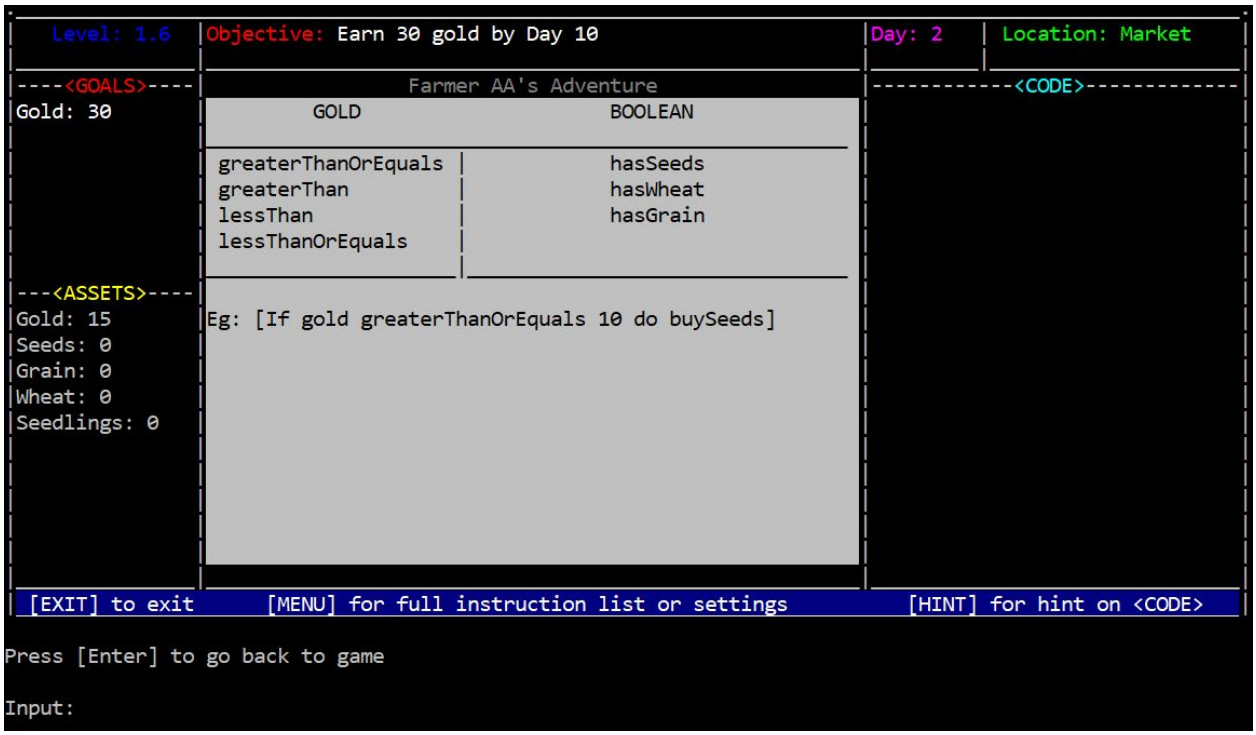

Figure 3.4.9.1. Condition List

3.4.10. MarketList

The marketlist command allows you to see the current market price that goods are currently being bought or sold at the current level.

Format: `market`

Displays:

- Market Items. This refers to the kind of items that can be bought or sold at the market.
- Price. This refers to the price of the item that is bought or sold at.

Figure 3.4.10. List of Market Rates

3.4.11. Log

The log command allows you to see the task sequence that was run in the previous simulation.

Format: `log PAGENUMBER`

> `PAGENUMBER` Must be a positive non-zero integer

Each log page displays a maximum of 15 tasks per page. You can navigate through the log pages using the log command. For example, to view page two of the log, enter `log 2`. If there was an error during code execution, the log does not display the particular task which caused the error

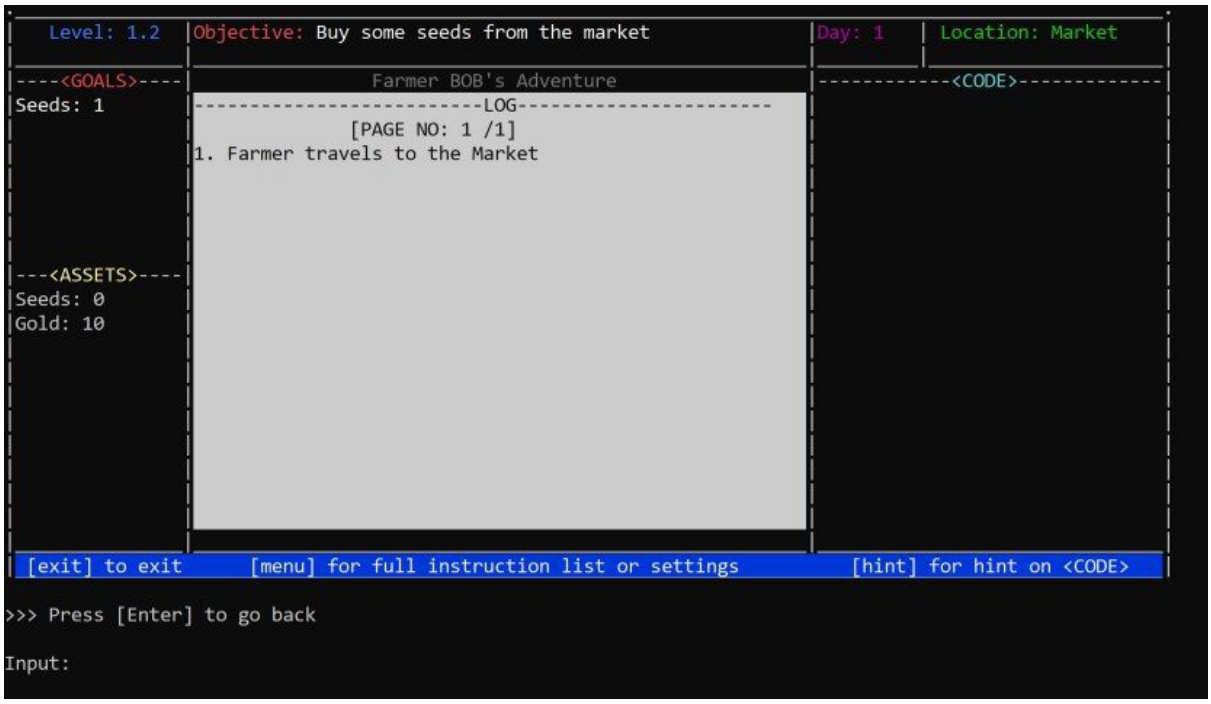

Figure 3.4.11.1. Example log page

3.4.12. Starting the Simulation

Starts the Simulation.

Format: `start`

Enter this command after you have added the relevant tasks to solve the objective.

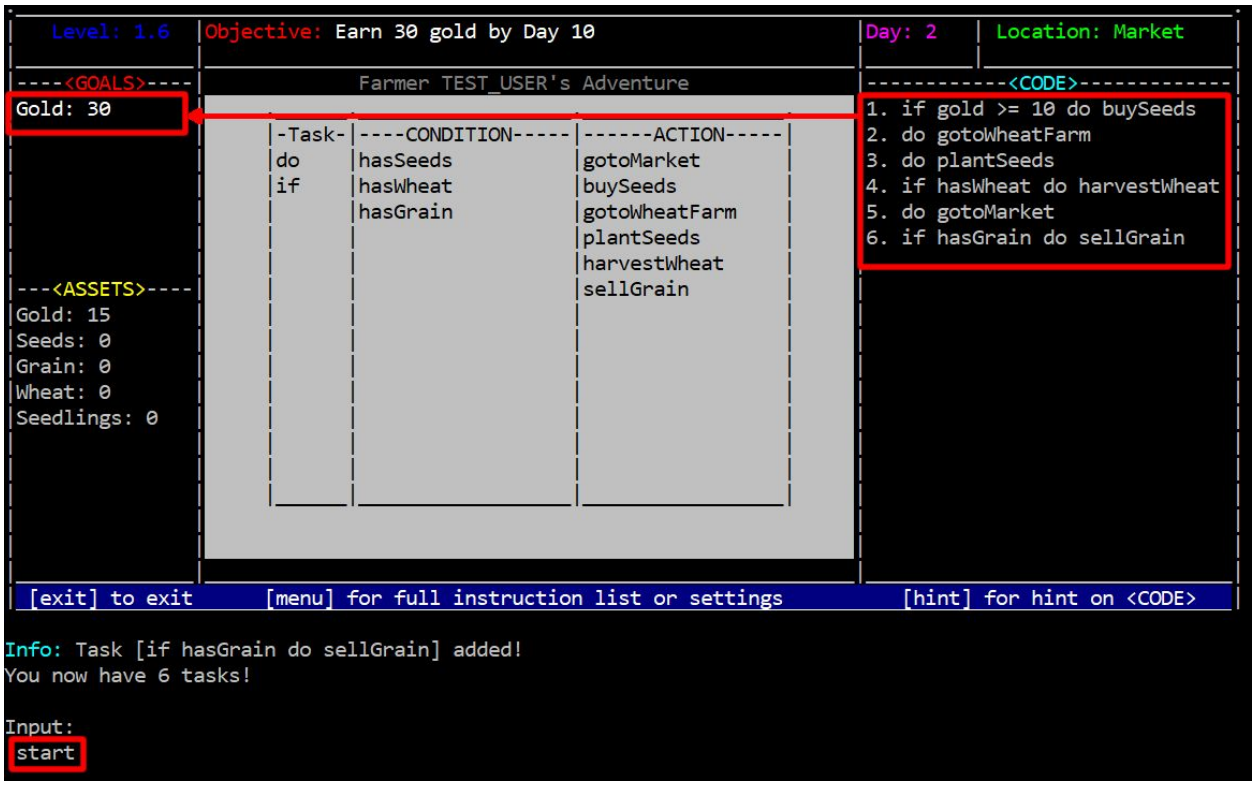

Figure 3.4.12.1. Using the start command

If you wish to speed up the simulation, you can use the fast mode command before you start the simulation.

Format: `fastmode`

Entering this command will set the game simulations to be in fast mode. Only the task description, as well as changes in assets, will be highlighted. To exit fast mode, simply enter `fastmode` again.

3.5. Simulation Segment

After you started the simulation, the game will now be in the Simulation Segment. During this segment, you should not enter any input until you are prompted to. If you do, your input will be carried forward to the next time the game requests for input.

This section explains what is happening during the simulation segment.

> Do note that colors are not supported in the windows command prompt. As such, symbols are also used to help you understand the simulation.

3.5.1. Currently executing task

As seen in Figure 3.5.1.1, the task that is currently being executed is indicated by the '<<' symbol on the right of the simulation. Visual animation of the currently executing action is also shown in the center of the game console.

On color enabled terminals, the currently executing task is highlighted in green.

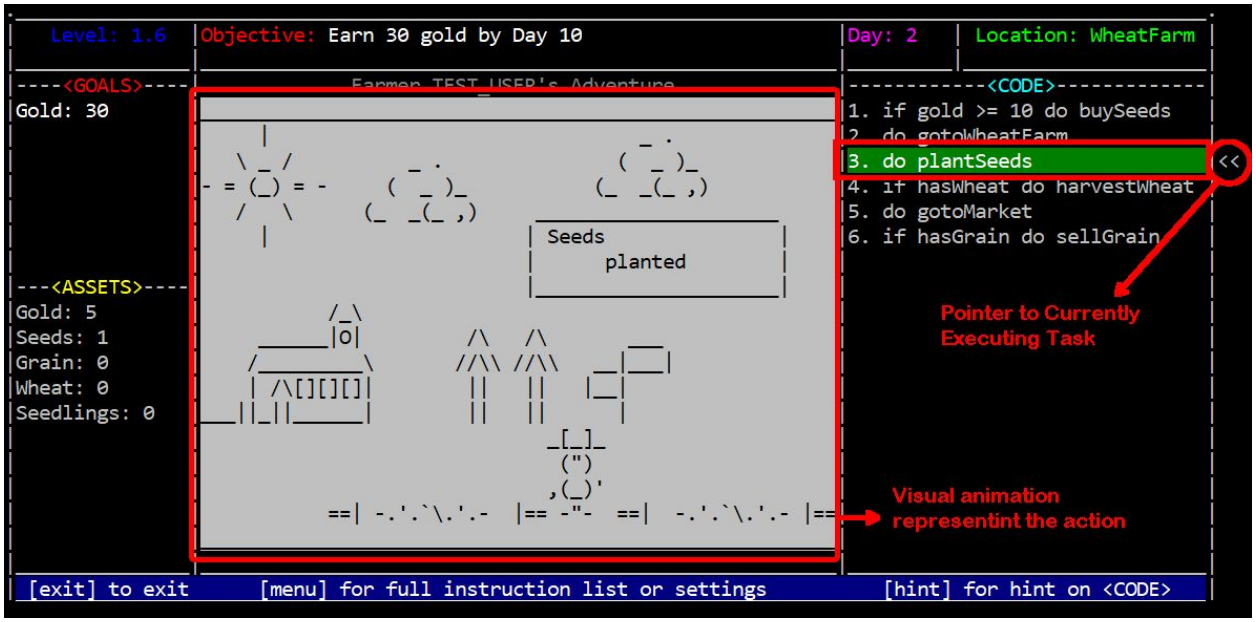

Figure 3.5.1.1. Execution of 'plantSeeds' action

If the condition for the task is not fulfilled, there would be a message saying ‘Condition not fulfilled, not executing task!’ below the console. This can be seen in Figure 3.5.1.2.

Figure 3.5.1.2. Task not being executed

3.5.2. Asset and Location tracing

After a task has been executed, there would be a change in either location or assets.

A change in location can be observed at the top right corner of the game console. Figure 3.5.2.1 shows the change in location before, during and after `[do gotoMarket]` is executed.

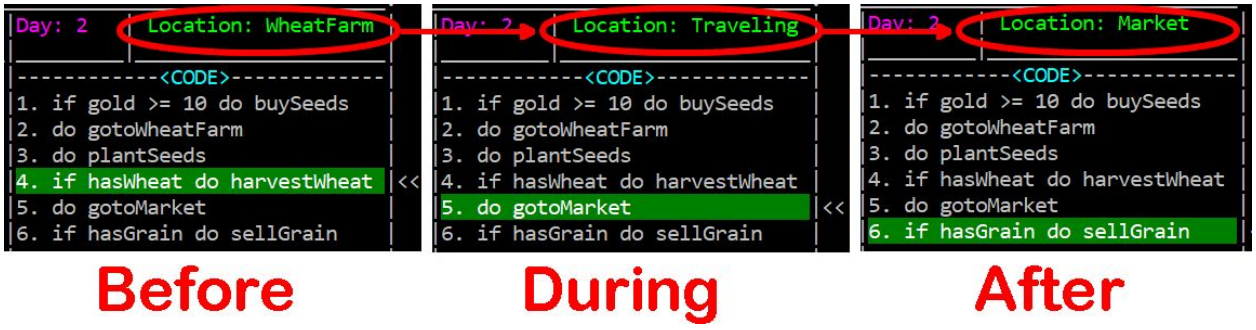

Figure 3.5.2.1. Location change

A change in assets will be signaled with the '->' symbol. The quantity of the asset before the change is on the left of the arrow while that after the change is on the right of the arrow. This can be seen in Figure 3.5.2.2. This signal would occur anytime there is a change in assets, and not only during the simulation segment.

On color enabled terminals, an asset change is highlighted in magenta.

Figure 3.5.2.2. Highlighting of Change in Assets

3.5.3. Error in executing a task

If an error is encountered during task execution, you can easily identify which task caused the error by finding the task marked with a symbol ‘X’ on the right.

On color enabled terminals, the task with the error is highlighted in red.

You can also find out the reason for the error by looking at the message below the console. This can be seen in Figure 3.5.3.1 by the text marked as ‘Reason for error’.

Press `Enter` to proceed to reset the level. You will be brought back to the story segment of that level.

In the event that you have an error in execution, the game will reset the level for you. The existing list of tasks will be retained for easy modification.

Figure 3.5.3.1.

3.5.4. End of day

After all, tasks have been completed for the day, if the level objective has not been completed and the deadline has not been reached, you will be prompted to either continue the game or reset to the start of the level.

Continuing the game

You should continue the game if you are confident that you can complete the objective.

Press `Enter` to proceed.

Restarting the Level

You should restart the level if you realize that you are going to fail the objective.

Format: `reset`

You will restart the level from the story segment.

> When you restart the level, your existing list of tasks will be retained for easy modification. Only your assets, day count and location will be reverted to what it was before you started the simulation segment.

Figure 3.5.4.1. End of day

3.6. Feedback Segment

At the end of each level, feedback will be provided to you.

3.6.1. Completed Objective

If the objective was met within the stipulated deadline for that particular level. A recap on the list of actions that were required to complete the level will be brought up for you.

Figure 3.6.1 Successful feedback

3.6.2. Failed Objective

If the objective was not met within the deadline allocated for the level, feedback will be provided to you to assist in interpreting on how you could have done better for the level. You will be provided with two pieces of information, the percentage of the correct number of tasks and the percentage of correct tasks. The percentage of the correct number of tasks allows you to see whether you are close to the number of tasks used for the optimal answer. The percentage of correct task will be used to show how close you are to achieving the optimal answer but will only be shown when the percentage of the correct number of tasks is at a hundred percent.

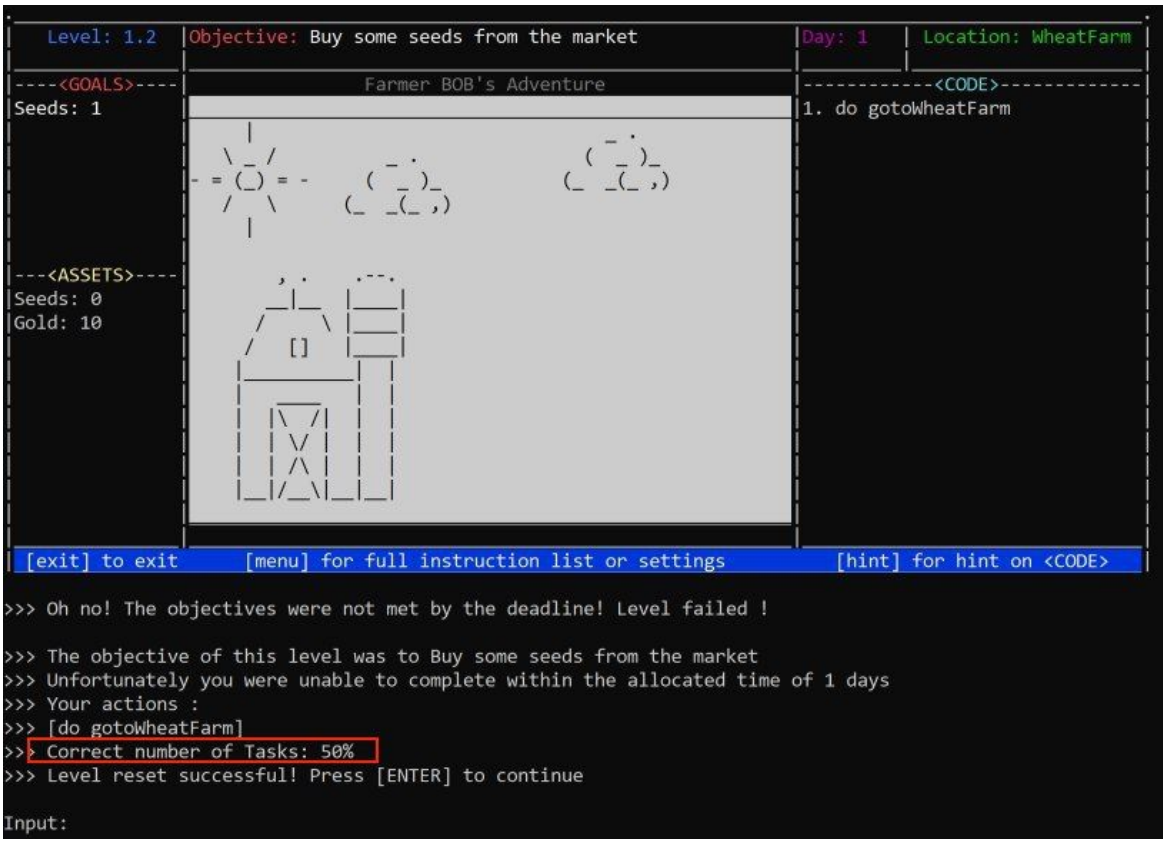

Figure 3.6.2.1. Correct number of tasks percentage

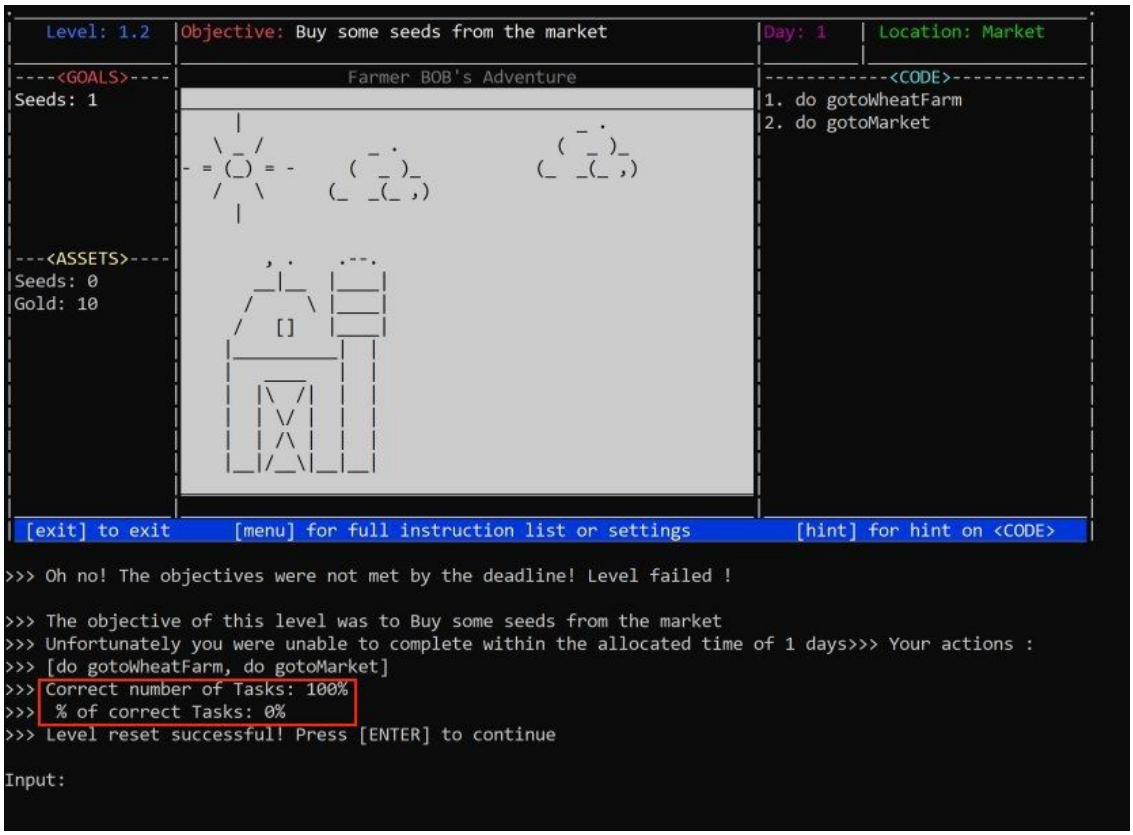

Figure 3.6.2.2. Percentage of correct tasks

4. Saving Your Game

4.1. Save Game

Format: `save game`

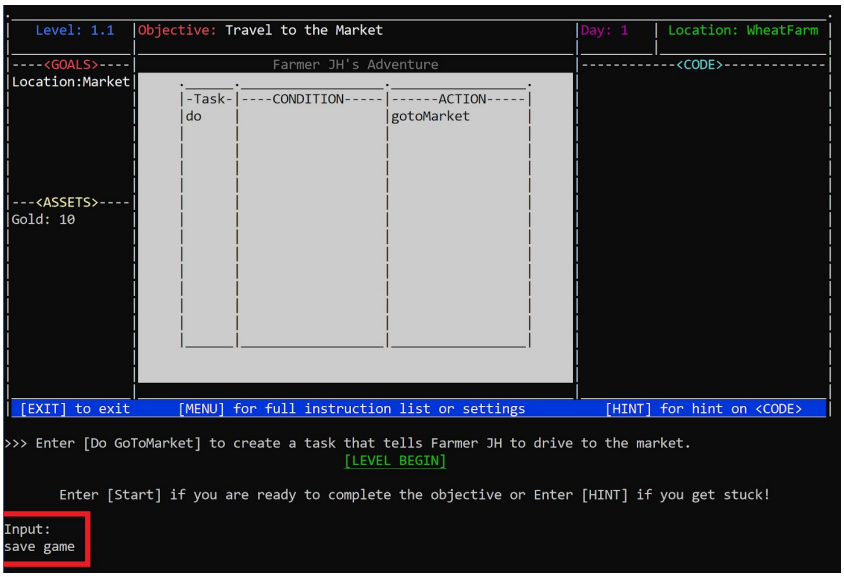

Figure 4.1.1 - Save Game Command

Upon successful game save, you will see a success message and the save file location on the screen.

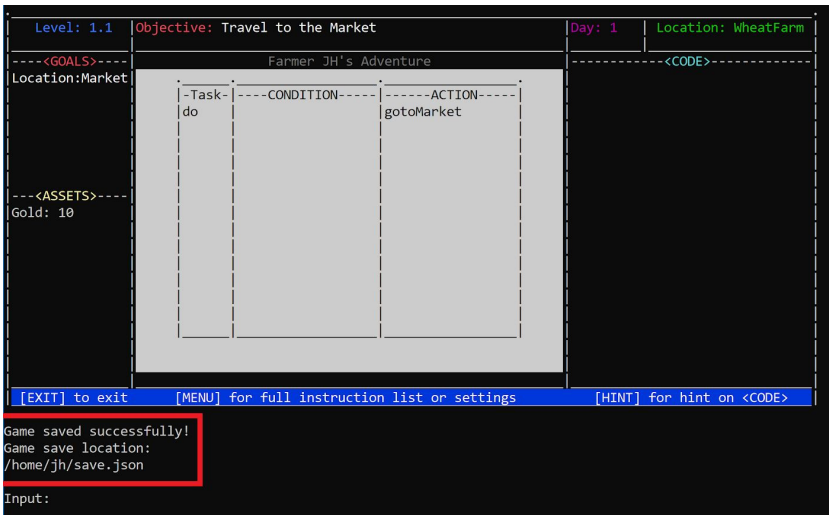

Figure 4.1.2 - Save Game Message

You can only use the save game command in the interactive segment

If you are in the story segment please use the skip command to enter the interactive segment to save your game.

If you are unable to save your game, you may be executing the game in a permission restrictive path. Please ensure that you are in a non-permission restrictive path before running the game.

4.2. Quit Game

This command allows you to quit the game

Format: `quit game` or `exit`

Step 1: Enter `quit game` or `exit` in the terminal.

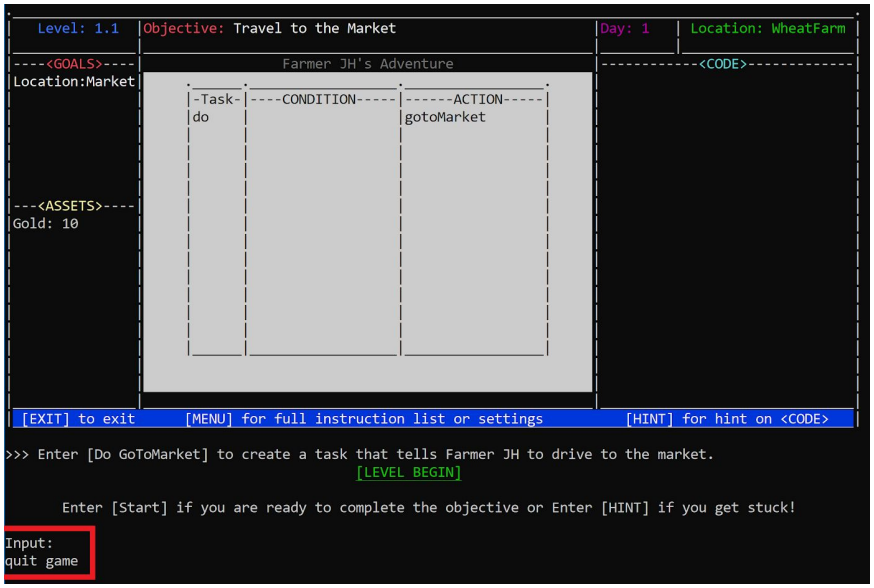

Figure 4.2 - Quit Game Command

Step 2: Upon successful game quit, you will see an animation and a goodbye message.

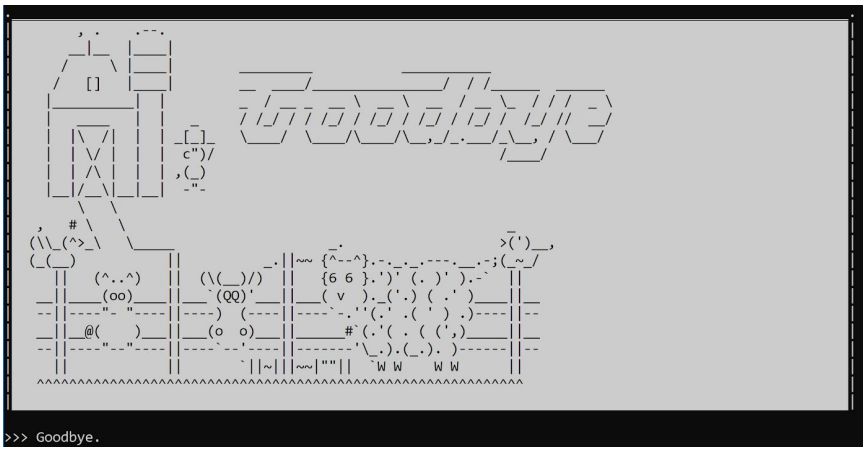

Figure 4.2 - Quit Game Animation

You cannot use the `quit game` command during the feedback and simulation segment.

5. FAQ

Will there ever be a GUI version of this game?

Farmio was envisioned with an objective to teach students computational thinking. As such we believe that it is crucial for our target audience to familiarise themselves with the command line hence we do not intend to develop a GUI version of this game.

How do I export my data to another computer?

As all the game files are stored in the directory where the game is saved, simply copy the entire game file into the other computer, and you can resume your game on the other computer.