Tasks Per User Story:

User Story 1- I want to have farming tools of the early 1800s.

Acceptance Tests for user story 1:

Given: User has created a garden hoe

When: User strikes a block of dirt

Then: A variable number of blocks will be harvested

Given: User has created a scythe

When: User strikes a block of grass

Then: A variable number of grass blocks will be harvested

Given:User has created a push plough

When: User right clicks an area of dirt

Then: Multiple blocks of dirt are tilled

Tasks: -Create a Scythe tool,

-Create a Reaper tool (better Scythe),

-Create a Hand Pushed Plough

-Create a more realistic garden hoe

-Create a more realistic shovel

-Have basic farm map created

Timeframe:

Week of May 19th: Create Reaper and Scythe tools

-Paired Programming during Tuesday’s meeting.

(Nasser & Robbie, Arnold & YuCheng, Jake Updating Charts)

-Consistently run JUnit tests on any algorithmic methods per object.

-Log any test data

-Update Burnup Chart

Week of May 26th: Create Hand Pushed Plough, and farm map

-Paired Programming during Tuesday’s meeting

- Consistently run JUnit tests on any algorithmic methods per object.

-Log any test data

-Update Burnup Chart

Week of June 2nd: Create Modified Garden Hoe, and Modified Shovel

- Paired Programming during Tuesday’s meeting

- Consistently run JUnit tests on any algorithmic methods per object.

-Log any test data

-Update Burnup Chart

-Ensure deliverables checklist is met

End of Iteration 1

**Iteration 2:**

User Story 2 :As a player I want to be charged fees at the end of the month for basic needs, such as food, rent, supplies, children’s needs, as this will show the difficulties facing farmers in that time period.

Acceptance Test for user story 2:

Given: 30 days past the starting date.

When: Sunrise

Then: There will a system message implying that the monthly rent is due and will be auto withdrawn.

Given: The system message for food is present

When: The user fails to have the appropriate amount of food for their family

Then: The user is penalized a strike

Given: The user is penalized with 2 strikes.

When: The user is penalized with the 3rd strike.

Then: The user loses their land thus losing the game.

Tasks:

-Create basic currency item.

-Currency item should take a recipe consisting of vegetable(s).

-Create time dependent system message for monthly rent

-Modify monthly rent system message to withdraw rent

-Add penalty strike counter to system message

-Modify penalty counter to increment only when rent isn’t paid

-Modify system message for required food/month. Keep the strike counter attribute, but change from rent dependent to vegetable number in inventory dependent.

Timeframe:

Week of June 9th:

- Create currency object and recipe

-Create basic system message object

-Paired Programming during Tuesday’s meeting.

-Consistently run JUnit tests on any algorithmic methods per object.

-Log any test data

-Update Burnup Chart

Week of June 16th:

-Create time dependent system message for monthly rent

-Modify monthly rent system message to withdraw rent

-Paired Programming during Tuesday’s meeting

- Consistently run JUnit tests on any algorithmic methods per object.

-Log any test data

-Update Burnup Chart

Week of June 23nd:

-Add penalty strike counter to system message

-Modify penalty counter to increment only when rent isn’t paid

-Modify system message for required food/month. Keep the strike counter

attribute, but change from rent dependent to vegetable number in inventory

dependent.

- Paired Programming during Tuesday’s meeting

- Consistently run JUnit tests on any algorithmic methods per object.

-Log any test data

-Update Burnup Chart

-Ensure deliverables checklist is met

End of Iteration 2

Iteration 3:

User Story 3: As a player, I want to have to deal with price changes on crops because this mimics conditions in the market of America during the mid 1800s.

Acceptance Test for user story 3:

Given: A system message for price changes is displayed

When: ----------------

Then: The price on crops is modified

Tasks:

-Add a market controller for the yield size of a stack of currency when crafting currency via vegetables (i.e. allow us to give the user 10 dollars for a carrot, or 5 dollars depending upon the market at the time).

Timeframe:

Week of June 30th + Week of July 1st:

-Add market controller for yield size

- Paired Programming during Tuesday’s meeting

- Consistently run JUnit tests on any algorithmic methods per object.

-Log any test data

-Update Burnup Chart

(ADD HERE) Burnup Chart of #of Completed Tasks vs. Timeframe of Project: