**BUNN’S CHRISTMAS TREE TALE**

1. **Introduction**
2. **Background**

As a part of our Python Algorithm and Programming class, we are expected to create a comprehensive application or game to further deepen our knowledge about Python. The project has to be extended than what have been taught in class. In other words, this final project expected us to make an application that is not taught yet and it is outside our comfort zone to learn new concepts in Python and solve or debug the problems that we encountered during the creation of this specific application.

Hence, after some research, I decided to make a story game that focuses in a farm using Python, spesifically the Pygame module that isn’t taught in the course. I learnt it by reading the documentation in pygame.org and watching some Youtube videos. I also learnt how to make a map in Tiled and incorporate it in my game using PyTMX (Python Tiled Map XML) module that is also not taught in this semester’s course.

I was heavily inspired by the Youtube video that Clear Code made about Pydew Valley. I adapted some of the farming physics that he taught in his video in my game. However, my main point in making this game isn’t to let player to only experience a farm game, but I was more inspired to make a wholesome story that happens in a farm for player to experience during the Christmas holiday.

1. **Problem Identification**

The first thing that I decided to when I first thinking about my Python Algorithm and Programming Final Project is to make a simple game that can be solved only for like some minutes, but gives a rather useful information. My inspiration is to make a bite-sized simple game, like Google games (example: Google Doodle Champion Island, Pangolin Love, Black Cat Academy, etc.). So, after some thought, I decided to make a farm game, and the useful information that I picked for this game is a story to cherish your family and don’t let yourself drown in depression or other kind of sadness because life still continues. We have strived and do our best to not let the people that cares about us disappointed. The game is aimed only to be played for a few minutes, but will give players the useful information that is also fit in the holiday theme.

In the beginning of the game, player will be directed to an intro page and if player clicks the play button, then the game will be started. The game will be started by the first cutscene about the introduction of the story’s main character (Bunn’s) background story. Then, after each cutscene, there will be some simple tasks that the player has to accomplish, so that they can proceed the story. One of the tasks is to raise money and this can be achieved by farming and raising animals.

After the story, the player will get some money and a decoration will be blitted in the player’s house. Player can choose to continue playing the game (but there is no story anymore) or exit the game. The story is not too long too, so this will also let player feel that their time didn’t get wasted by playing this game.

So, my main objectives for creating this game are:

1. Let player experience a warm-hearted holiday story.
2. Let player draw a message from the story.
3. The player doesn’t have to waste that much time to experience the story.
4. **Project Specificiation**
5. **Game Name**

“Bunn’s Christmas Tree Tale”

The game is related to a Christmas tree and the main character’s name is Bunn. So, the game name is simply based on them.

1. **Game Flow Summary**

“Bunn’s Christmas Tree Tale” is a game where the player has to walk through the story of Bunn and how he get his Christmas tree. Player has to control Bunn to move around the map and doing things, like going to some places or raising money.

The player or Bunn can raise money by planting seeds, which is differentiated into two types (tomato and corn), cutting woods, or raising cows for their milks. Players can sell their crops (corns and tomatoes), woods, or milks by selling them to the merchant. Player has to use specific keys in the keyboard to moves around the map, toggle menus (inventory, help, and shop), and use specific tools.

Player also has to press only the key ‘C’ in their keyboard to start a cutscene. If the objectives or missions from the last cutscene is not completed yet, then the player can’t move to the next part of the story.

1. **Game Display**

The objects of this game are created using the sprite sheets from the Sprout Land assets, which are 2D pixel art assets. Meanwhile, the background is the map image that I created using Tiled and the tilesets that are also from the Sprout Land assets.

When player first opened this application, they will not be directed to the map instantly, but they will rather be directed to an intro menu with the background of a pixeled grass image. There will be also a play button and our main character’s Bunn. If player clicks the play button, then a cutscene will start.

Every cutscene display is divided into two parts, the main picture and the dialogue part. The main picture is about 550 – 700 pixels in height, showing the relevant pictures of what told in the story. Meanwhile, the dialogue part is also differentiated into two different types. The first type is the narrator’s line, which the background is a brown rect, while the second type is the dialogue of the character. The background of the second type of the dialogue is a dialogue box and the head profile of the character that is talking.

After that, there is the game default background, which is the map or the world as the background. There are many objects in this game background, from plants, trees, player, cows, merchant, houses, etc. However, there are also overlay display on bottom left and top right of the screen. On the bottom left, there is the display of the tool and the seed type that player currently uses. While, on the top right, there are two clickable menus. The first one is the inventory with cart symbol and the second one is the help menu about the keyboard shortcuts in this game with the question mark symbol. Both of them can also be toggled using the keyboard shortcut ‘P’ for inventory and ‘H’ for help.

1. **Game Physics**

Below are the explanations of the game physics that exists in the game:

1. Collision detection

Some objects like trees, plants, fences, and others have hitboxes, so that it can detect collisions. In this game, if a collision happens, then the player can’t move to that direction. For example, if the player crash or hit a tree when moving to the right, then the player can’t move to the right further. In this game, except objects that is derived from the class Generic, I also made some collisions in the map. The collisions can be seen through Tiled which it’s marked by red colour. With this, we can control player, so that they can’t walk on the water or walk out of the map.

1. Tree

If the player uses an axe and the target position is a tree object, then the tree will be cut. If the tree is cut, then the stump will be blitted on the screen rather than the tree because it is cut. There are two type of trees, the large one and the small one. Different trees have different sizes of stump. After that, if the player goes to sleep, then the tree will be reset. The tree will revive again (the stump surface will be replaced by the tree surface again) and player can chop it again.

1. Tilling Soil

Player can only till the soil if the player use hoe and the soil is farmable. If the soil is not farmable, example: soil tile that is near to water body or below a tree, then the soil will not be tilled. Tilled soil will lost it’s green color and changed color to brown as it is tilled.

1. Growing Plants

Every plant has four types of stage when they are planted. The first state is when they are still a little sprout. This stage occurs when the player first used a seed on a plantable soil. The second and third stage are the plants when they are still growing, while the last (fourth) stage is the stage when the plant is harvestable. Every seed has to be planted into a plantable or a tiled soil, if not, then the seed will not be planted and the amount of the seed will not be deducted. If the seed is planted, then it won’t grow even if the player resets it for several days if the seed is not watered. So, the player has to water the seed. If the seed is watered, then it will grow to become a plant on the next day. If the player watered it every day, then after several days, when the plant reach the fourth (harvestable stage), they can harvest it.

1. Harvesting Milk

The mechanism for harvesting milk is similar to the growing plants. The different is that when growing plants, we have to water it every day, but for getting milk, we have to feed the cow with grass every day. If the cow reached the max age, then we can harvest the milk by again giving the cow grass. The status of the milk (it is ready or not) can be seen by checking the milk bottle inside the cow farm. If it’s full, then the player can harvest the milk.

1. Weather (Rain and Sunny)

The weather system in this game is achieved by using the random library. The probability for a rain to happen in the game is 0.3 or 30%. If it’s raining, then the tilled soils will be wet, so the player doesn’t have to water the soil anymore. There will be also water drop that is blitted on the screen.

1. Day-night Transition

There is also a day-night transition in this game, so the sky will be darker along with the pass of time. If the player goes to sleep, the everything will be resetted, including the sky colour. The sky will become light again and if player plays for quite a long time, it will become dark again.

1. **Game Input**
2. Mouse left button, to click the play button in the intro menu, quit, or to click the overlay help and inventory menu on top right of the game when the game has started.
3. Keyboard keys:

* 'Esc' to quit the game.
* 'A' to move the player to the left.
* 'D' to move the player to the right.
* 'W' to move the player upward.
* 'S' to move the player downward.
* 'Space' to use current tools and if the merchant menu is on the screen, ‘Space’ is used to purchase or to sell something.
* 'Q' to switch current tool.
* 'J' to plant current type of seed.
* 'I' to switch seed with other type of seed.
* 'P' to open the player’s inventory menu.
* ‘H’ to open the help menu.
* 'O' to close the help menu, inventory menu, or merchant menu if they are active.
* 'Enter' to let player sleep (reset the day) if it is pressed near the player’s bed, open the merchant menu if it is pressed in front of the merchant, and to feed the cow if it is pressed inside the cow farm.
* 'Arrow' to navigate between items that are available when the merchant menu is active.
* 'C' to start a cutscene after doing some missions.
* ‘LCtrl’ to proceed to the next line of dialogue in a cutscene.

1. **Game Output**
2. Player image. The player has many kind of images based on what they are doing at the moment, from idle, using specific tools, and walking (based on the player’s input). The direction which they do something also determine which player image will be blitted to the screen. The images of the player are all contained in the ‘graphics/character’ folder.
3. Background image. The background image can be accesed through the ‘graphics/world’ folder which the name is ‘ground.png’.
4. Object images. The rest of the objects’ images like cuttable trees, flowers, fences, houses, cows, milk, stumps, and plants. are also in the ‘graphics’ folder, but in different type of folder based on their own name. The reason why they don’t exist in ‘ground.png’ because they are planned to have a hitbox, so that they can detect collisions.
5. Water and rain image. The water images can be accesed through the ‘graphics/water’ folder, while the rain image can be accessed through the ‘graphics/rain’ folder where the water and rain have several states (it is animated).
6. Button and overlay images can be accessed in the ‘graphics’ folder with their respective name. The button and overlay image is blitted on top of the screen, so that it will move around with the player.
7. Soil images. The default soils has the colour of green, but if it’s farmable and it is tilled by the player, then it will change its image. The brownish image is available in the ‘graphics/soil’ image. If the soil is further watered, then there will be also water on top of the soil. The water that is on top of the soil is one of the images in ‘graphics/soil\_water’.
8. Cutscene images. The images for the dialogue box, character’s emoticon, and cutscene picture is available in the ‘graphics/dialogue’ folder where for every cutscene, there will be a specific folder for them based on their order.
9. Player’s money.
10. Player’s inventory. The list of items that player currently has can be accessed in the game by toggling the inventory menu. There will be also image for each item that the player has. The image for those are available in the ‘graphics/inventory’ folder.
11. Background music is in the audio folder and its name is ‘bg.mp3’.
12. Sound effects are also available in the audio folder. The sound effects in this game are

* When player using an axe to chop a tree (axe.mp3)
* When player buy something (buy.mp3)
* When player click on the play button in the intro menu or toggle the help and inventory menu (click.mp3)
* When the player press LCtrl to proceed to the next line of dialogue in a cutscene (cutscene.mp3)
* When the player feed the cows with grass (grass.mp3)
* When the player uses hoe to till soils (hoe.mp3)
* When the player harvest milk (milk.mp3)
* When the player plants a seed (plant.mp3)
* When the player sells something (sell.mp3)
* When the player acquired crops or woods (success.mp3)
* When the player waters a plant (water.mp3)

1. **Game Libraries (Modules)**
2. Pygame

Pygame is a set of Python modules to create a game. Used for creating the game window, the game objects and blitting them to the screen by using surface and rect, drawing text, playing sounds, and detecting collisions.

1. PyTMX

PyTMX or Python Tiled Map XML is a set of Python modules specified to load and handle map for games. So, we can easily edit our map in Tiled and use it in Pygame by utilizing PyTMX. In this game, PyTMX helped me to get the tile layer’s and object layer’s name and properties based on Tiled, so that I can use it in the program.

1. random

The random module in Python is a built-in module that is used to generate random choices or numbers. In this game, I used the randint and choice from the random module. The randint function is used to help generating the probability of a rain happening, while the choice function is used to choose one of the elements inside a list. The choice function is used when generating the water on top of the soil tiles and the water drop and water floor when it is raining.

1. os

Python has a built-in module, called the os module that contains methods to handle and interact with the operating system. In this game, I use the walk function from os module that generates the file names in a directory tree by walking the tree. I used it to help me to return the surfaces from the images in specific folder.

1. sys

The sys Python module provides various functions and variables to manipulate Python runtime environment. In this game, I use the exit function to exit the program when the player quit the game or pressing the ‘Esc’ key.

1. **Game Important Files and Folders**
2. ‘audio’ folder, contains the background music and the sound effects for this game.
3. ‘code’ folder, contains the Python code of this game.

* cutscene.py, contains the class CutSceneOne, CutSceneTwo, CutSceneThree, and CutSceneFour. Each of them is used to handle their own respective cutscene. cutscene.py also contains the class CutSceneManager to manage cutscene and update the status of a completed cutscene.
* game\_display.py, contains the class Display that is used to manage all the game functions and displays if the player has started the game and the class CameraGroup to move the camera based on the player movement, so that the player is still on the center of the screen.
* game\_settings.py, contains the constant values of some variables that are used in the game, like the screen width, screen height, and many more.
* intro.py, contains the class Intro that is used to draw and manage the intro menu screen (the display screen before the player starts the game).
* main.py, contains the class Game that is used to handle methods to run the game (contains the game loop).
* merchant\_menu.py, contains the class Menu that is used to handle the merchant shop, so that the player can buy and sell something through the shop menu.
* overlay\_menu.py, contains the class Overlay\_Menu that handles the help menu display and the class Inventory that handles the inventory menu. Both of them are displayed overlay on the screen, that’s why they are in the same file.
* overlay.py, contains the class Overlay that is used to handle the images on the bottom left of the screen that represents the current used tool and current type of seed.
* player.py, contains the class Player that handles the player’s movement, animation, tool usage, collisions, and many more that is related to the player.
* sky.py, contains the class Sky that draws a darker sky by the pass of time, the class Drop to handle the movement of the rain drops and its lifetime, and the class Rain to create the rain drops on the screen and update it.
* soil.py, contains the class Plant to handle methods for growing plants, the class SoilTile as the tilled soil tiles after player uses a hoe, the class WaterTile as the water on top of the soil tiles after player waters the soil tiles, and the class SoilLayer to handle methods of tilling and planting in the game.
* sprites.py, contains the class Generic for every kind of objects in the game. Besides Generic, there are also the class Interaction, Water, WildFlower, Particle, Tree, and Animal which are all inherited from Generic.
* The class Interaction is used to handle non-visible objects but player can interact with them, for example, in front of the merchant, besides the bed, etc.
* The class Water is used to draw waters and animate them.
* The class WildFlower is used to draw wildflowers.
* The class Particle is used to draw Particle when a crop is harvested.
* The class Tree is used to handle method that is related to a cuttable tree.
* The class Animal is used to draw and animate the cows.
* support.py, contains the class Timer to know and handle the duration of a player when doing something and the class Button to handle and draw button on the screen.
* transition.py, contains the class Transition to draw the transition if the player goes to sleep and reset the day.

1. ‘data’ folder, contains the .tmx file map for this game’s ground background and several .tsx files inside the tilesets folder. The .tsx files are used in the process to create the complete map in the ‘map.tmx’ file.
2. ‘document’ folder, contains the documentation (report and diagrams) for this project.
3. ‘font’ folder, contains the font types that are used in the game.
4. ‘graphics’ folder, contains all of the game picture assets. The ‘graphics’ folder contains 16 different folders that are all the pictures or sprites for the game. The folders are ‘animal’, ‘button’, ‘character’, ‘dialogue’, ‘environment’, ‘fruit’, ‘inventory’, ‘milk’, ‘objects’, ‘overlay’, ‘rain’, ‘soil’, ‘soil\_water’, ‘stumps’, ‘water’, and ‘world’. The ‘environment’ and ‘objects’ folder contains the images to create the tilesets in the previous ‘data’ folder, meanwhile the rest besides of them are images that are going to be blitted on the screen when the game runs.
5. ‘story’ folder contains four .txt files that are the story or the narration for each cutscene of this game.
6. **Solution Design**
7. **Design Choices**

For this game, I wanted to have a light-hearted and cute character as the main character. The game atmosphere should also be light-hearted and casual, so that the player can experience a warm-hearted story in this casual game.

For the image design, I decided to use the online free spritesheets assets from Sprout Land for my project because the character and the game elements are cute as I intended to. They also have more free assets than the other farm assets that I have searched. Below are the image of the Sprout Land assets.



Source: https://cupnooble.itch.io/sprout-lands-asset-pack

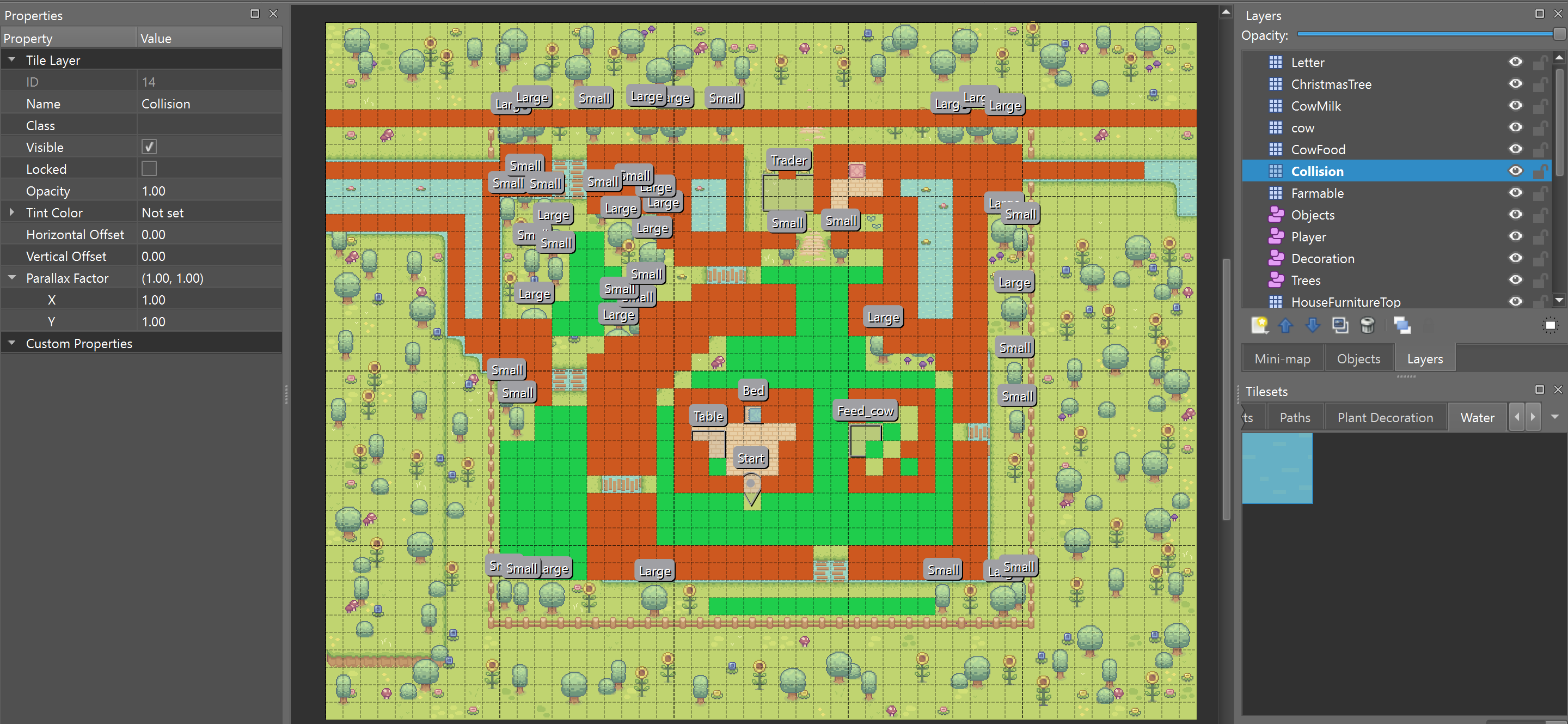
The only object that I didn’t use from this assets pack is the Christmas tree and the Christmas hat which are shown below.

Source: https://www.istockphoto.com/id/vektor/pohon-natal-seni-piksel-dengan-ikon-vektor-bintang-dan-pernak-pernik-untuk-game-8bit-gm1431396087-474006614



Source: https://www.freepik.com/premium-vector/santa-hat-pixel-art-style\_35166269.htm

From the assets pack that I have downloaded, I then create a map in Tiled for the world environment for this game. The map can be seen below. The green tiles are the farmable tiles (the tiles that can be tilled), while the red tiles are the collision tiles (player can’t walk across these tiles). The fence is not marked as collision tiles because the fences will be generated as a Generic object later and Generic object has hitbox.



Below are the map’s appearance if the collision and farmable tiles are not visible. The red and green tiles inside the cow farm and Bunn’s house is not the farmable and collision tiles.

The red tile inside the house is for the Christmas tree, while the green one is for the letter. Meanwhile, the green tiles inside the cow farm is for the cows and the milk bottle, while the red one is for the cows’ food.



Result



1. **Font and Sound**

The font that I used for this project are LycheeSoda and Rubik (Bold and Semibold). LycheeSoda is used mostly in the game because it’s in pixel style, so it is very suitable for the overall design, while Rubik is only used for the help menu because I think that the help menu should have a straighter and easier to see font, so that the player can read it more easily.

Source for LycheeSoda: https://www.dafont.com/lycheesoda.font

Source for Rubik: https://fonts.google.com/specimen/Rubik?query=rubik

Meanwhile, for the sound, as I said before, I wanted to have a light hearted song, so I chosen a light hearted background music created by MusicTown from Pixabay (https://pixabay.com/music/search/light-hearted/) and rename it to bg.mp3.

And below are the sources for the other sound effects that I have used throughout the game:

* axe.wav

Source: https://www.soundsnap.com/axe\_impact\_hit\_wood\_blastwavefx\_26204

* buy.mp3

Source: Pixabay from Pixabay (https://pixabay.com/sound-effects/coin-c-02-102844/)

* click.mp3

Source: UNIVERSFIELD from Pixabay (https://pixabay.com/sound-effects/interface-124464/)

* cutscene.mp3

Source: UNIVERSFIELD from Pixabay (https://pixabay.com/sound-effects/button-124476/)

* grass.mp3

Source: https://www.youtube.com/watch?v=4eVmKIfZFpY

* hoe.mp3

Source:https://www.videvo.net/sound-effect/hoe-chop-dirt-grub-soil-pehd069902/247265/#rs=audio-download

* milk.mp3

Source: https://www.youtube.com/watch?v=3aOnFCVz9iY

* plant.mp3

Source: Pixabay from Pixabay (https://pixabay.com/sound-effects/21-seed-46670/)

* sell.mp3

Source: Pixabay from Pixabay (https://pixabay.com/sound-effects/cash-register-fake-88639/)

* success.mp3

Source: https://www.zapsplat.com/music/retro-80s-arcade-game-jump-8/

* water.mp3

Source: Pixabay from Pixabay (https://pixabay.com/sound-effects/pouring-drink-sound-effect-by-rea-102018/)