Gaming for the Greater Good Studios

Tree Simulator 2016
Game Reference Manual

Prepared by:
Davis Masterman, Event Design
Sotirios-Aris Mitropoulos, Art Design
Michael Wong, Game Design
Haoran Xiao, Quality Control

Prepared for:
Gloria Geo, Chief Operating Officer
David Chesney, Chief Technology Officer
Mary Jane Northrop, Chief User Experience Officer
Mindy Misener, Chief Communications Officer

Date Prepared:
12 December 2016

Table of Contents

Executive Summary

As part of G3 Studios's objective to improve society through game design, the current project is focused specifically on helping children affected by ASD. ASD(autism spectrum disorder) is a widespread disability affecting many people. It is accompanied by numerous symptoms that vary in severity and occurrence from person to person. G3 Studios assigned the current project to our team and we developed the game Tree Simulator 2016 to help improve the lives of those with ASD.

Our team focused specifically on the ASD symptoms of impatience, routinized behavior, and a lack of focus. We hoped that by playing Tree Simulator 2016, the player would improve their patience and focus and would be more able to break out of routines. Thus the game would train the player and allow them to apply the skills they learn in-game to their daily lives.

Tree Simulator starts out relatively simple, with the player clicking to water trees. However, distractions appear on screen forcing the player to split their attention and train their ability to focus. Random events also occur to force the player to break out of any routines they established. The game overall takes time to complete and thus improves the player's patience.

This manual focuses on the purpose of the game in greater detail, how the game plays from a player perspective, how the game is designed from a developer's perspective, how the game works in detail, and how to install the game and what is needed to install the game. In the appendix, a selection of the sprites used and an example of how the game is coded is shown.

Introduction

Gaming for the Greater Good Studios is dedicated to the bettering of society through game design. One of G3 Studio’s current target audiences is young children with Autism Spectrum Disorder (ASD). Given the broad range of symptoms in ASD, the latest product of G3 Studios, Tree Simulator 2016, will specifically target the treatment of routinized behavior in children with ASD. The game is designed to aid young children around the age of 10 with high functioning autism due to their presumed motor control and language capabilities. Tree Simulator 2016 will accomplish G3 Studio’s goal by providing an interactive tool for parents of children with ASD to reduce repetitive behaviors in the children.

Manual Purpose & Audience

This reference manual presents detailed information about the purposes and components of Tree Simulator 2016. It can be reviewed by different audiences for several purposes:

- **G3 Studios management** can review the Overview for a forecast of the finished product and how the game benefits ASD treatment and G3 Studios.
- **ASD researchers** can refer to the Overview, and Gameplay Experience Narrative sections for an in-depth description about the behavioral treatments this game offers for children.
- **Developers of the game** will code specific parts according to the Game Modules presented. They can refer to the UML diagrams in Coding Approach for development directions and clear communication. The Gameplay Experience Narrative can also be used to help focus the design for the intended audience.
- **New members of the game design team** can review the Coding Approach and the Appendices as a resource for quickly understanding the overall design of the game and expected programming contributions.

Overview

Tree Simulator 2016 is designed to target problem areas such as lack of patience, dependence on routines, and an inability to deal with distractions which are often displayed in children with high functioning autism. The target audience will need to have a usable understanding of written English and should be able to improve in the above skills while playing Tree Simulator 2016. The varied gameplay elements challenge the player and force the player to adapt. While the player tries to accomplish the main objective, distractions appear in the form of background images or enemies. This teaches multitasking and trains the ability to change focus on the fly.

Overall the game is designed to be slightly addicting to keep the player interested and help lengthen the user's patience.

The player starts out in the position of a pseudo-god, benevolently watering the trees which sprout. In order to win in Tree Simulator 2016, the player needs to gain enough points. The player can gain points by growing and taking care of trees.

When the player clicks, a water droplet falls from the sky. If the water hits a tree, the tree is watered. If a tree is watered enough it will eventually grow from a sapling to a tree. The tree will grow fruits that can be clicked on for points. When all fruits are removed, the tree will die and grant the player points. As the game is played, random events can occur such as lumberjacks, who cut down trees. The lumberjack can be clicked on to remove him.

The player can spend his points in the point shop that will improve the gameplay experience with upgrades such as more water. Further information on game mechanics can be found in the Game Screen module.

Game Experience Narrative

I open the game and I can see the main menu. Out of the five buttons I can see below the logo, I choose the one on the left that says "New game"; I have now entered the game. I am currently in a room that describes the story of the game. *Hahaha this is a quite humorous story line.* I now must choose a difficulty level. My choice is the central button, a medium difficulty setting.

By clicking it the gameplay begins. As I can see at the bottom of the screen, I start the game with a score of zero. My trees are located at the bottom half of the monitor. I drag my mouse and place it above a green tree. I click and a raindrop falls on it. I repeat this process a few times and I can see my tree growing from a sapling to a fully grown tree. To reach its final stage the tree has passed through three stages. Three red fruits exist on my tree. I click on them and gain points. *This seems rather pedestrian and boring.*

All of a sudden a message pops up over a yellow tree, saying "Help!", a lumberjack is attacking it. I look at the bottom right part of the screen and see a hint box; it says that if I click on the lumberjack, he will disappear. I do so and *poof* he's gone; I gain a few more points. A red bird flies over my trees, since they are, currently, no fruits for it to eat. Nevertheless, I click on it and kill it; this way, I gain a few more points. As I continue to grow my trees, I see a few of them are chopped down by lumberjacks and some fruits are eaten by the birds. This results in my score diminishing. As I gather the fruits of the blue tree, a rain cloud comes over and drops water all over the grass. *Wahoo this rain cloud is over powered it helps my trees grow so much quicker.*

I have been playing for some time now and *I have grown tired of the awesome, yet repetitive, background music*. I drag my mouse to the top of the screen and I see that the view shifts; I can now see the sun and a small, yellow, “MENU” button on the top right corner. I click on it and the game is paused. I now click on the note and speaker buttons on the bottom of the menu. All music is now muted. I then decide to visit the shop by clicking the “Open Shop” button. I now see my three upgrade options and I choose to spend 100 points and buy the “More water” upgrade.

I then press the “Resume game” button and return to the gameplay. My upgrade is working and the value of the water dropped has increased. I continue growing my trees, when one of them catches fire. I keep clicking on it, dropping water, until the fire stops. *That was a close one...* I gather a few more fruits and have now reached the threshold of 450 points. As a pop up message informs me, I have won the game. I am sent to a victory screen with lots of fireworks. I feel accomplished for beating the game. *This was… not as boring as it sounded.*

Coding Approach

To create Tree Simulator 2016, the design team chose to use GameMaker: Professional Edition. GameMaker offers grouping by sprites, sounds, objects, and rooms. This offers easy and clear organization of game elements as we build the game. The internal sprite editor became the convenient engine for most sprite and animation creations in the game. The included functions for room allows for creation of different screens in the game, such as the main menu and the instruction screen. GameMaker also can be coded either with scripts or drag and drop; this allows team members without programming experience to contribute efficiently.

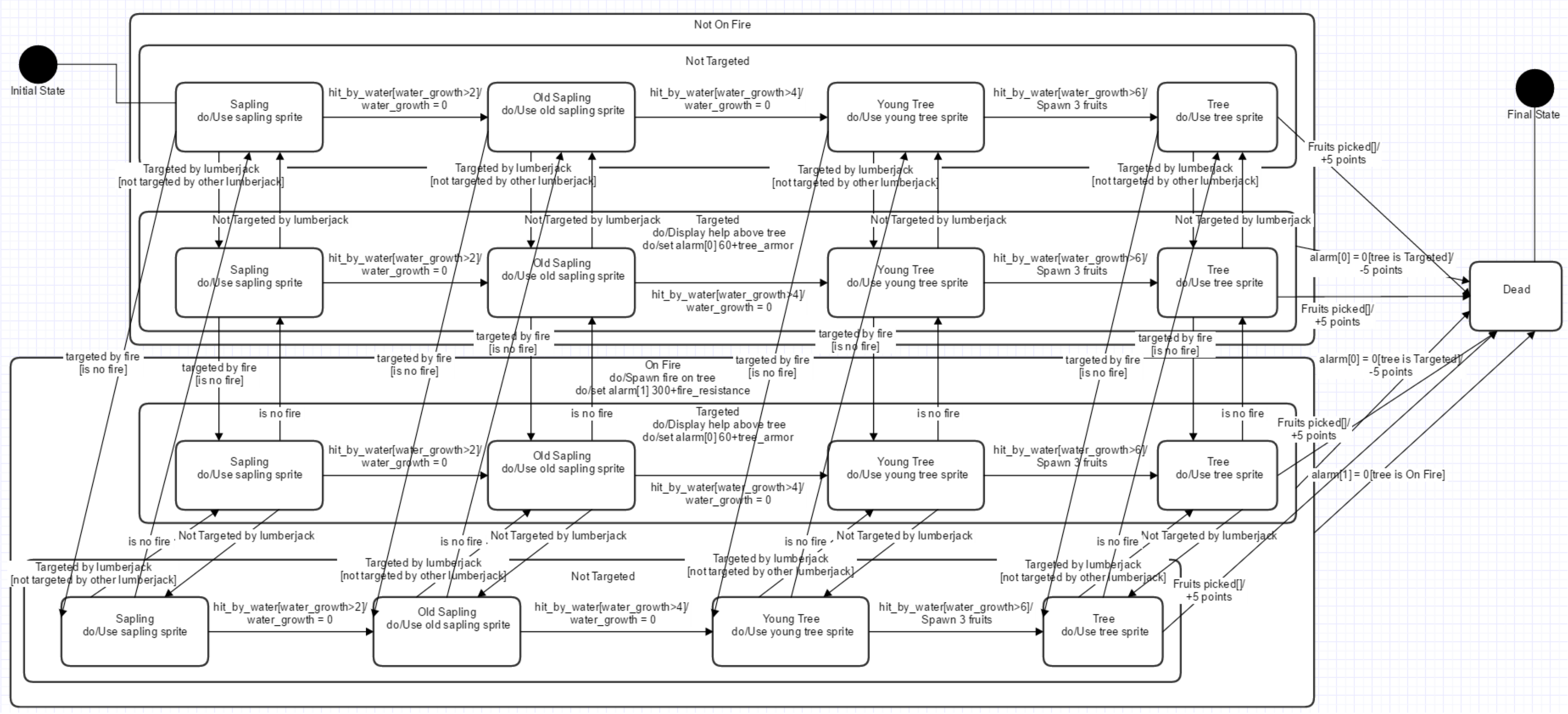
As a team, we developed a systematic approach to naming conventions and version control. All sprites have the prefix “spr_”, all objects start with the prefix “obj_”, and all scripts are named with “scr_”. All game files are uploaded to a Google Drive folder for working as a team. The files are named with the same format: “<release_version>_<date>_<uploader_name>.gmz”. Individual members are responsible for merging their current game file with the most recent uploaded file. Version control allows for both consistency in workflow and a way to backtrack in case of discovered bugs or merge errors.

UML Diagrams

The following two UML diagrams are a statechart of a green tree and a class diagram of main game mechanics, respectively.

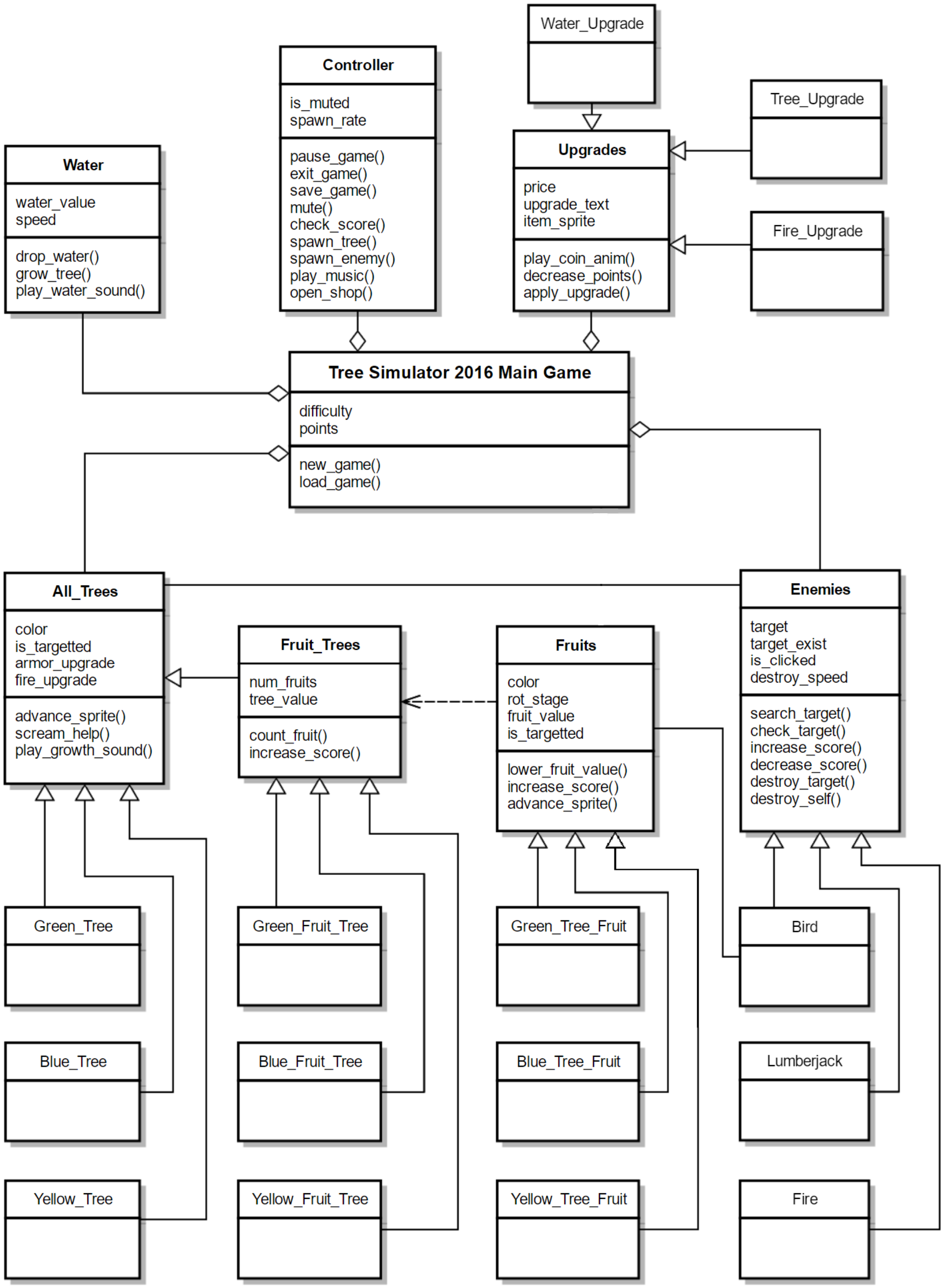
Statechart Diagram

The tree varies between three different types of states: the state of tree growth, the state of tree on fire, and the state of tree being targeted. While growing into a full tree from a sapling, the green tree can switch between being on fire and not being on fire and being targeted by a lumberjack and not being targeted by a lumberjack.


Initial State
Not On Fire
Not Targeted
Sapling
do/Use sapling sprite
hit_by_water[water_growth>2]/
water_growth = 0
Old Sapling
do/Use old sapling sprite
hit_by_water[water_growth>4]/
water_growth = 0
Young Tree
do/Use young tree sprite
hit_by_water[water_growth>6]/
Spawn 3 fruits
Tree
do/Use tree sprite
Fruits picked[]/
+5 points
Final State
Targeted by lumberjack
[not targeted by other lumberjack]
Not Targeted by lumberjack
Targeted
do/Display help above tree
do/set alarm[0] 60+tree_armor
alarm[0] = 0[tree is Targeted]/
-5 points
Dead
targeted by fire
[is no fire]
On Fire
do/Spawn fire on tree
do/set alarm[1] 300+fire_resistance
is no fire
alarm[1] = 0[tree is On Fire]
Not Targeted

Class Diagram

The class diagram contains the most essential elements of game mechanics: game controller, shop system, water droplet, trees, fruits, and enemies. The diagram shows interaction between trees, fruit trees, fruits, and specific types of enemies. Some of the interactions of the controller are seen in Appendix B.



Module - Main Menu

The main menu is the first screen the player sees when he/she enters the game. It includes the game title and logo. The main menu allows the player to start the game, load a game, check instructions, check credits, or exit the game through use of the mouse. (Figure 1)

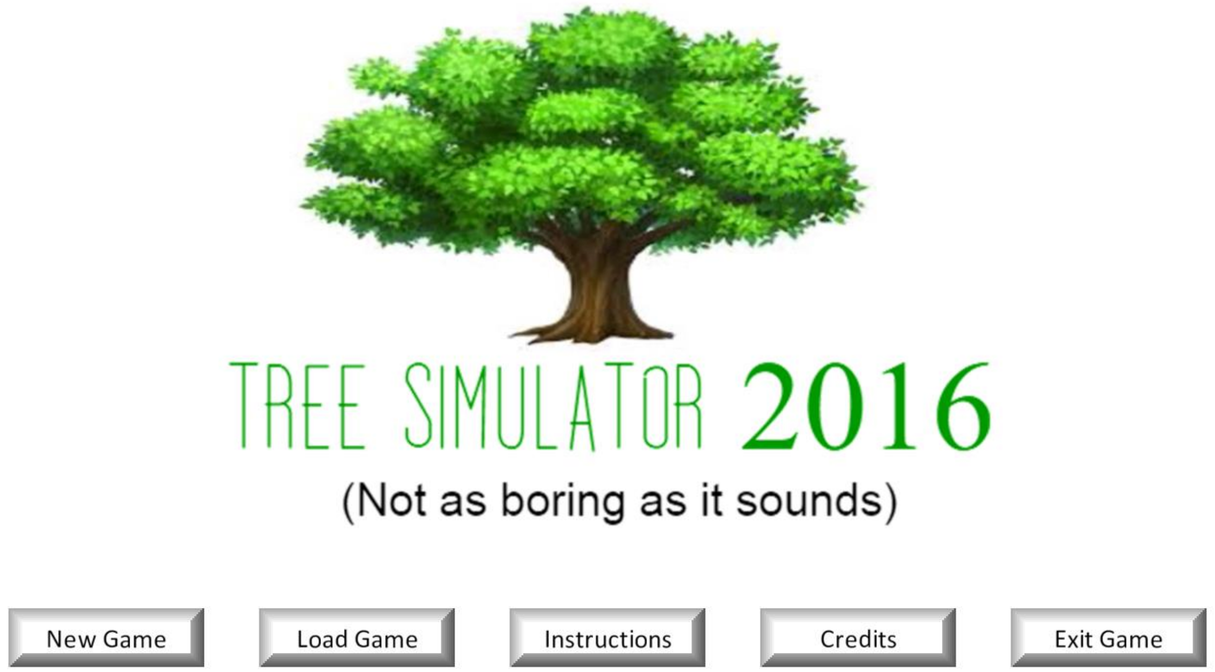


Figure 1. Main menu for Tree Simulator 2016

Main Menu Components

New Game- Button to start a new game for the player by going to the difficulty select screen

Load Game- Button to starts a game with the score, difficulty, and upgrades from the previous saved game

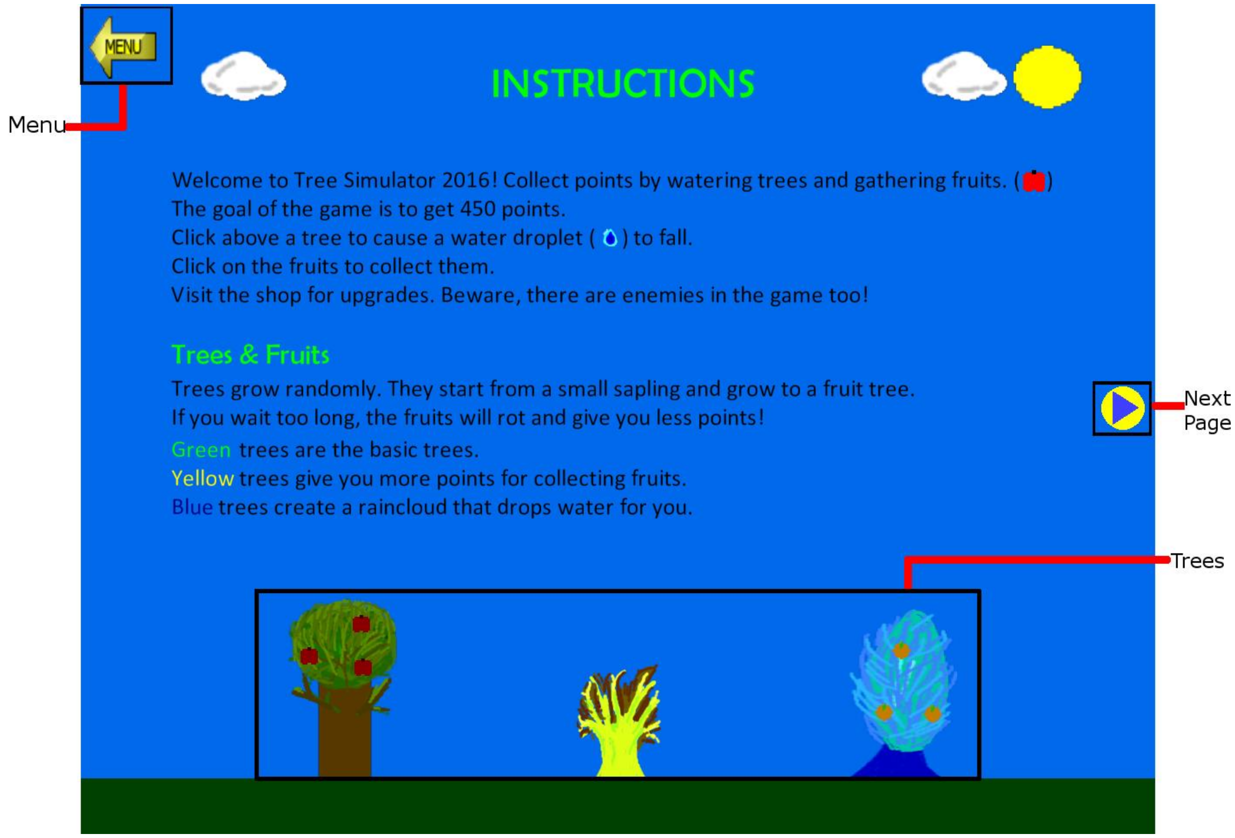
Instructions- Button to show the main objective for the player as well as how to play

Credits- Button to show the resources used and special thanks to outside helpers

Exit Game- Button to quit the game

Module - Instructions Screen 1

The instruction screen (Figure 2) provides information on the goal of the game, and a simple description of the gameplay. Players will be able to learn the controls for gaining points. The page also includes instruction on returning to the main menu.



INSTRUCTIONS

Welcome to Tree Simulator 2016! Collect points by watering trees and gathering fruits. (🍎)
The goal of the game is to get 450 points.
Click above a tree to cause a water droplet (💧) to fall.
Click on the fruits to collect them.
Visit the shop for upgrades. Beware, there are enemies in the game too!

Trees & Fruits

Trees grow randomly. They start from a small sapling and grow to a fruit tree.
If you wait too long, the fruits will rot and give you less points!
Green trees are the basic trees.
Yellow trees give you more points for collecting fruits.
Blue trees create a raincloud that drops water for you.

Figure 2. First instruction screen for Tree Simulator 2016

Instruction Screen Components

Background Object- Displays the instruction of the game in text, which explains the game objective, controls, and gameplay elements. Includes some images of gameplay elements

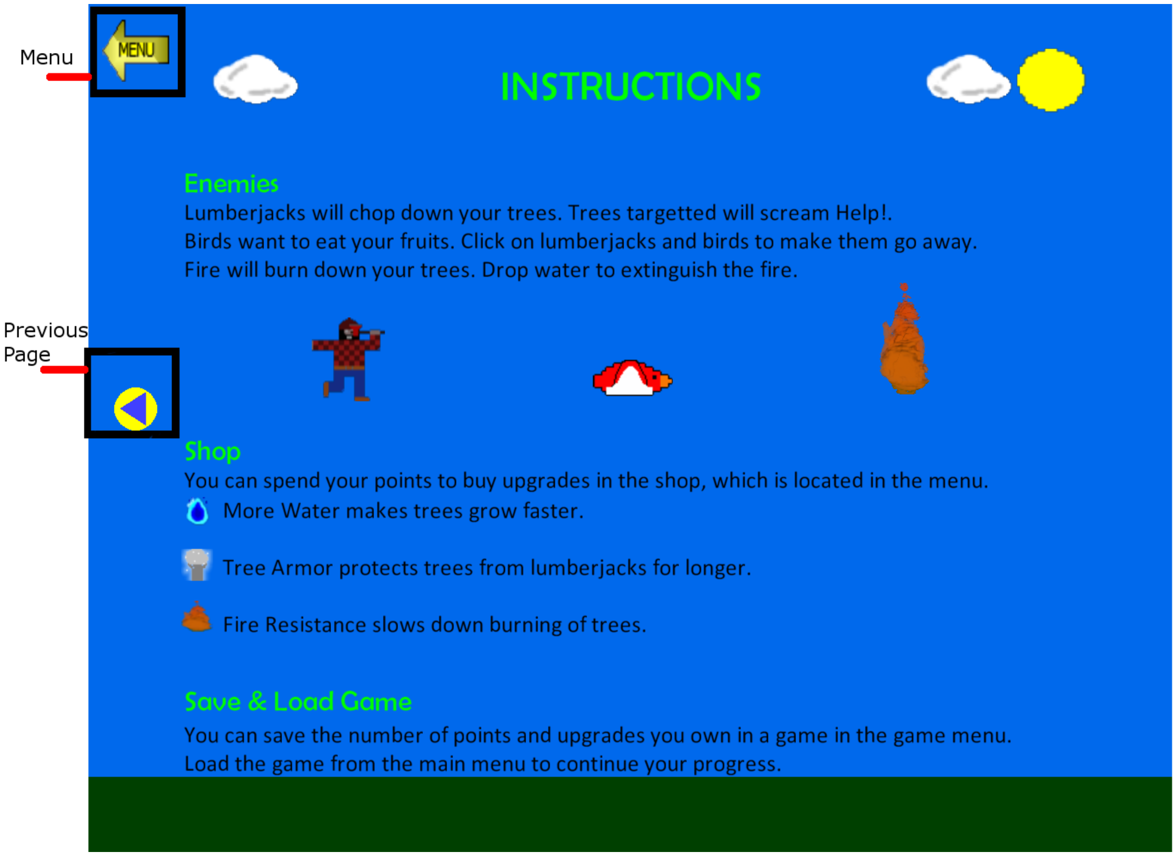
Trees- Gifs that display the three types of trees (discussed in-depth later) in all their stages

Menu- Button to return to the main menu

Next Page- Button to go to the next page of instructions

Module - Instructions Screen 2

The instruction screen (Figure 3) displays information of enemies, the shop upgrades, and the save/load system.



INSTRUCTIONS

Enemies

Lumberjacks will chop down your trees. Trees targetted will scream Help!.
Birds want to eat your fruits. Click on lumberjacks and birds to make them go away.
Fire will burn down your trees. Drop water to extinguish the fire.

Shop

You can spend your points to buy upgrades in the shop, which is located in the menu.

More Water makes trees grow faster.

Tree Armor protects trees from lumberjacks for longer.

Fire Resistance slows down burning of trees.

Save & Load Game

You can save the number of points and upgrades you own in a game in the game menu.
Load the game from the main menu to continue your progress.

Figure 3. Second instruction screen for Tree Simulator 2016

Instruction Screen Components

Background Object- Displays the instruction of the game in text, which explains the enemies and their purpose, the shop and its upgrades, and the save and load system

Enemies- Displays the three enemy types, lumberjacks, birds, and fire (discussed in-depth later)

Shop- Displays the three upgrades, More Water, Tree Armor, and Fire Resistance (discussed in-depth later)

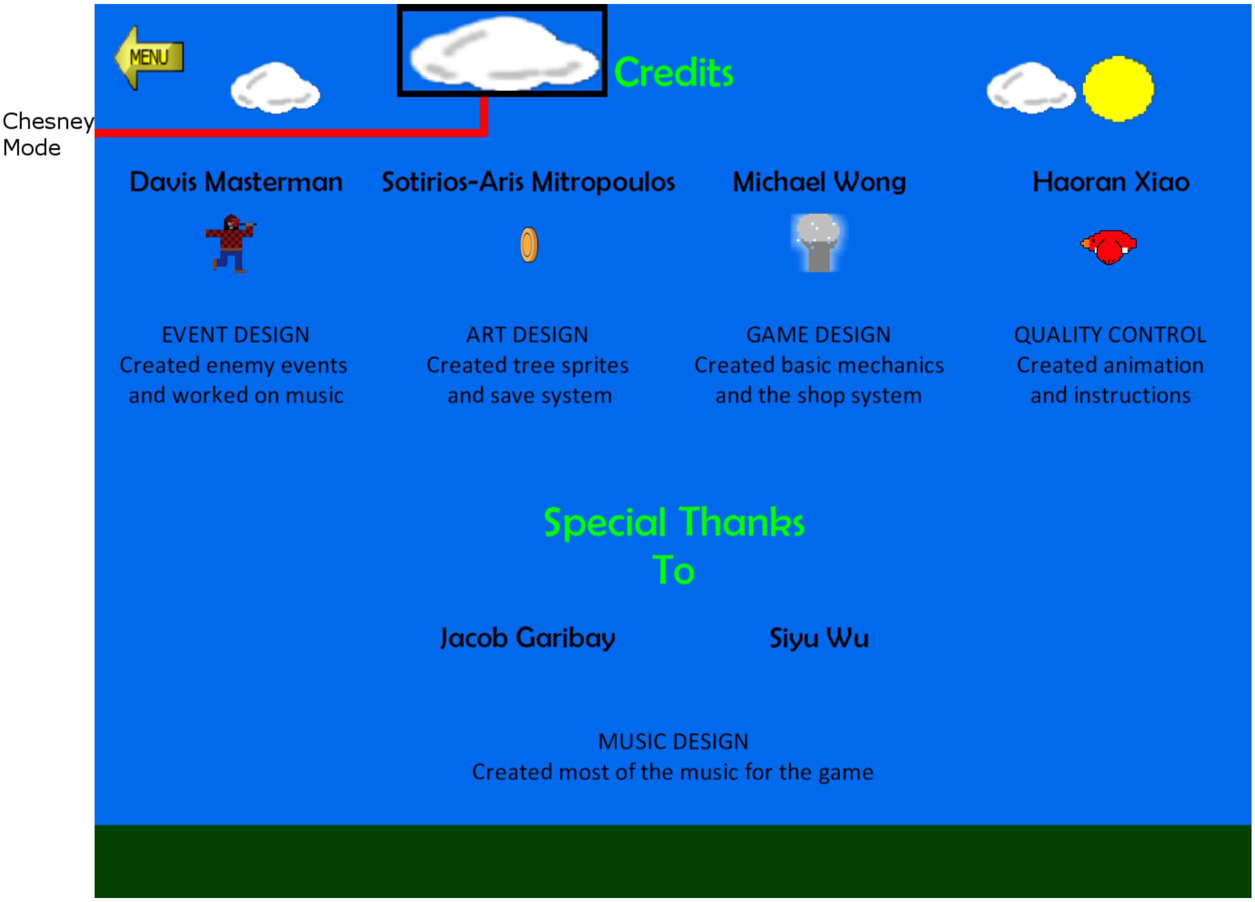
Save & Load- Displays that the number of points and the upgrades are saved and loaded

Menu- Button to return to the main menu

Previous Page- Button to return to the previous page of the instructions

Module - Credits

The credits screen provides information on the members of the team, as well as outside workers who worked on the game. It also provides the activation of Chesney Mode. (Figure 4)



Credits

Davis Masterman

EVENT DESIGN
Created enemy events
and worked on music

Sotirios-Aris Mitropoulos

ART DESIGN
Created tree sprites
and save system

Michael Wong

GAME DESIGN
Created basic mechanics
and the shop system

Haoran Xiao

QUALITY CONTROL
Created animation
and instructions

Special Thanks
To

Jacob Garibay

Siyu Wu

MUSIC DESIGN
Created most of the music for the game

Figure 4. Shows contributions of the team members and the contracted workers

Credit Screen Components

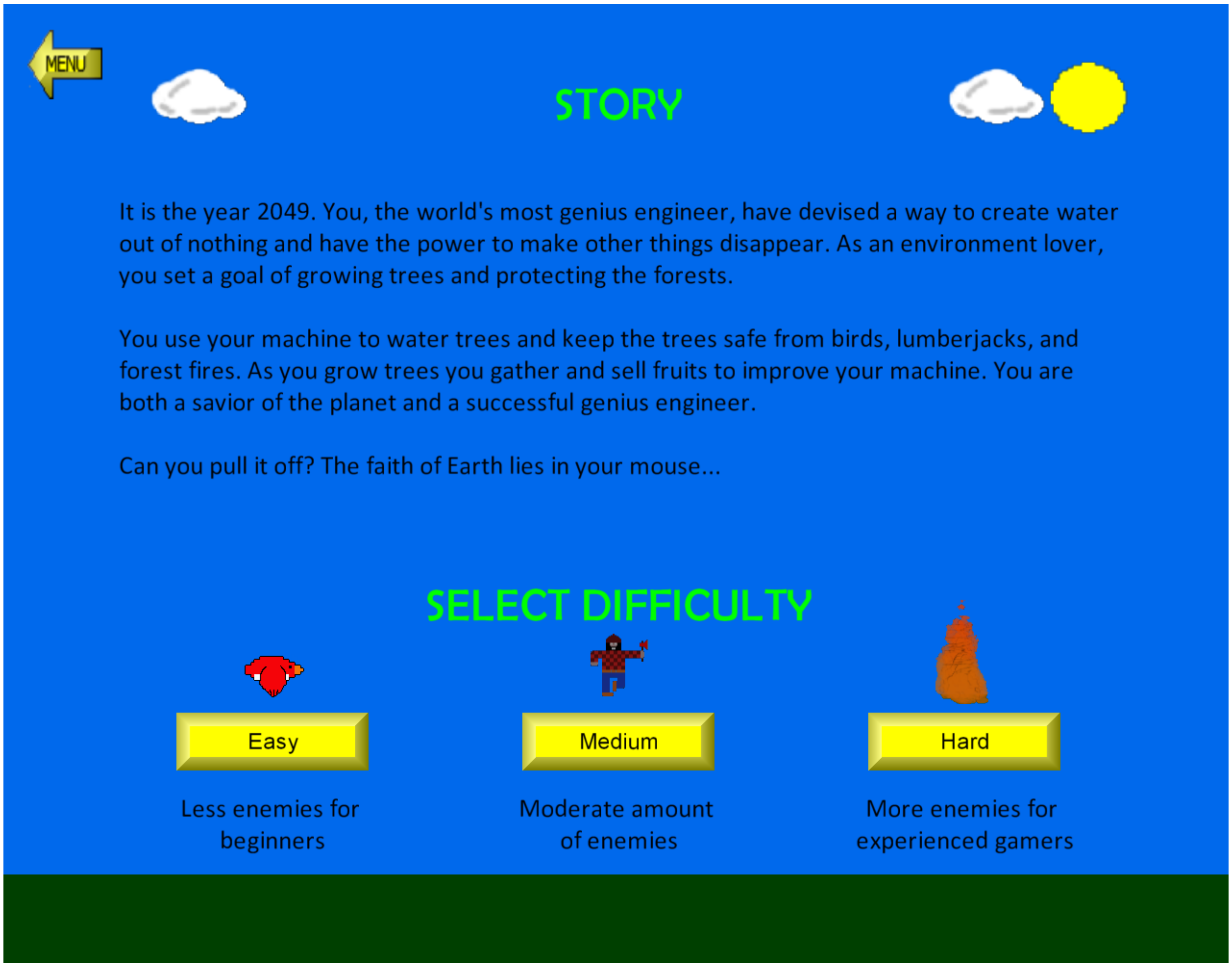
Background Objects- Shows the roles of the four members and the work done by all contributors to the game

Sprites- Shows each member’s best sprite contribution to the game

Cloud- When clicked, a porpoise image is revealed. Click porpoise image to enter Chesney Mode.

Module - Difficulty Select Screen

The difficulty select screen can be opened by selecting the New Game button on the main menu screen. This screen offers the backstory of the game, and allows the user to select the difficulty he/she wants to play at. (Figure 5)



It is the year 2049. You, the world's most genius engineer, have devised a way to create water out of nothing and have the power to make other things disappear. As an environment lover, you set a goal of growing trees and protecting the forests.

You use your machine to water trees and keep the trees safe from birds, lumberjacks, and forest fires. As you grow trees you gather and sell fruits to improve your machine. You are both a savior of the planet and a successful genius engineer.

Can you pull it off? The faith of Earth lies in your mouse...

Less enemies for beginners

Moderate amount of enemies

More enemies for experienced gamers

Figure 5. Shows how to select the difficulty of the game

Difficulty Select Screen Components

Background- The same static sky background as shown in the instruction and credits screens.

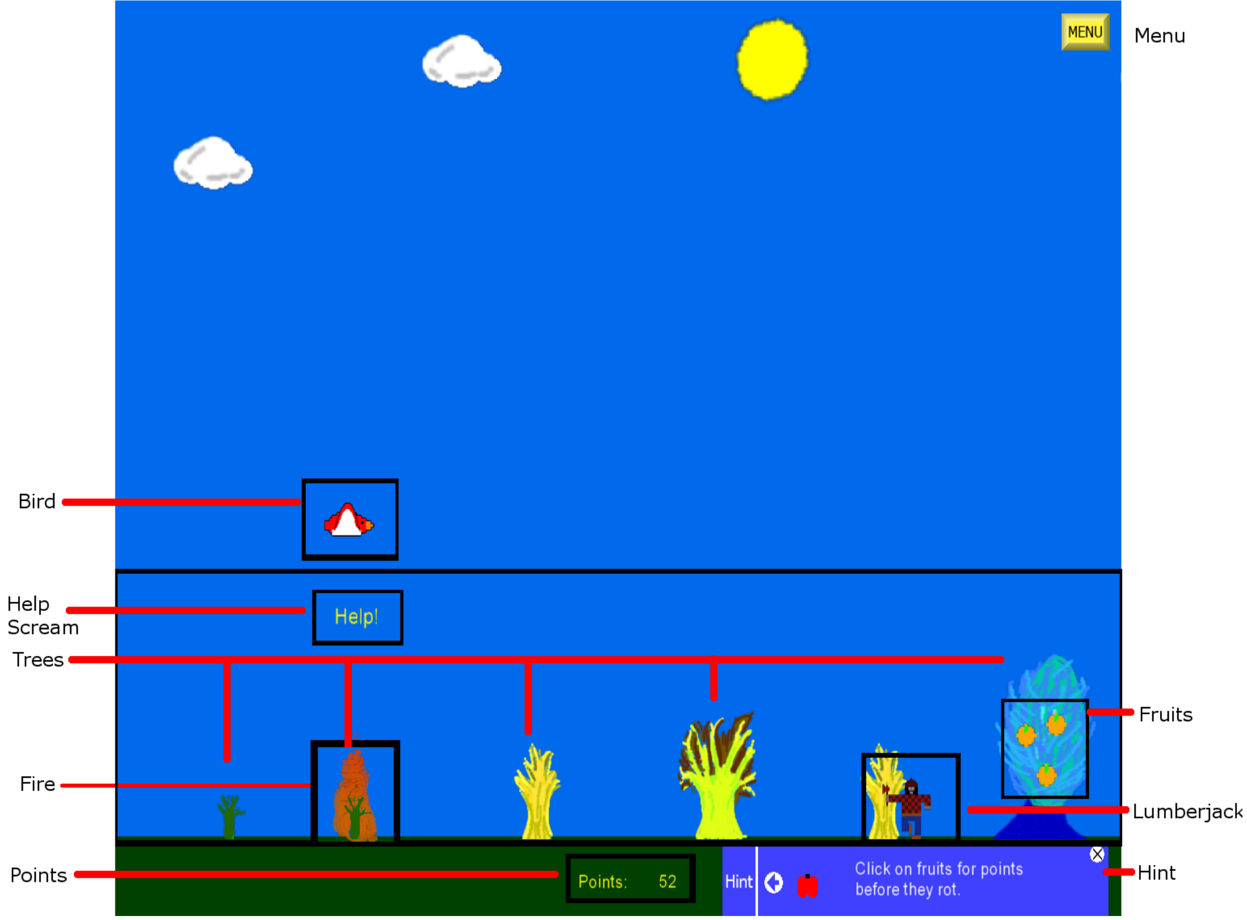
Introduction Text- A short paragraph that describes the backstory and scenario of the game.

Menu- Button to return to the main menu

Difficulty buttons- Three buttons with the text “Easy”, “Normal”, and “Hard”. They set the main game difficulty for the player. The text underneath the buttons describes what each difficulty contains. The sprites above the buttons (bird, lumberjack, and fire) express the difficulty in order of perceived dangers.

Module - Main Game Screen

The game is played on one screen (Figure 6) where all gameplay elements occur simultaneously. This is where the player earns points and attempts to win the game. Detailed information on individual in-game sprites can be found in Appendix A - Sprite Index.



Click on fruits for points before they rot.

Figure 6. Gameplay screen with multiple interactions between objects and player.

Game Screen Elements

Background- Animated sun and clouds with a bright blue sky

Menu- Button that pauses the game and allows access to the menu

Trees- Object that grows when watered and when fully grown grants points. When targeted by lumberjack, the trees display a “help!” message. There are three types of trees with different attributes. The colors and attributes of each tree are listed in Appendix A - Sprite Index

Bird- Object that seeks out fruits and attempts to eat them. They randomly spawn and can be destroyed by clicking on them

Hint- Object that aids a player with understanding the game

Fire- Object that randomly spawns if it has not rained from a raincloud recently

Lumberjack- Object that seeks out a tree and attempts to cut it down and deny points. They randomly spawn and can be destroyed by clicking on them

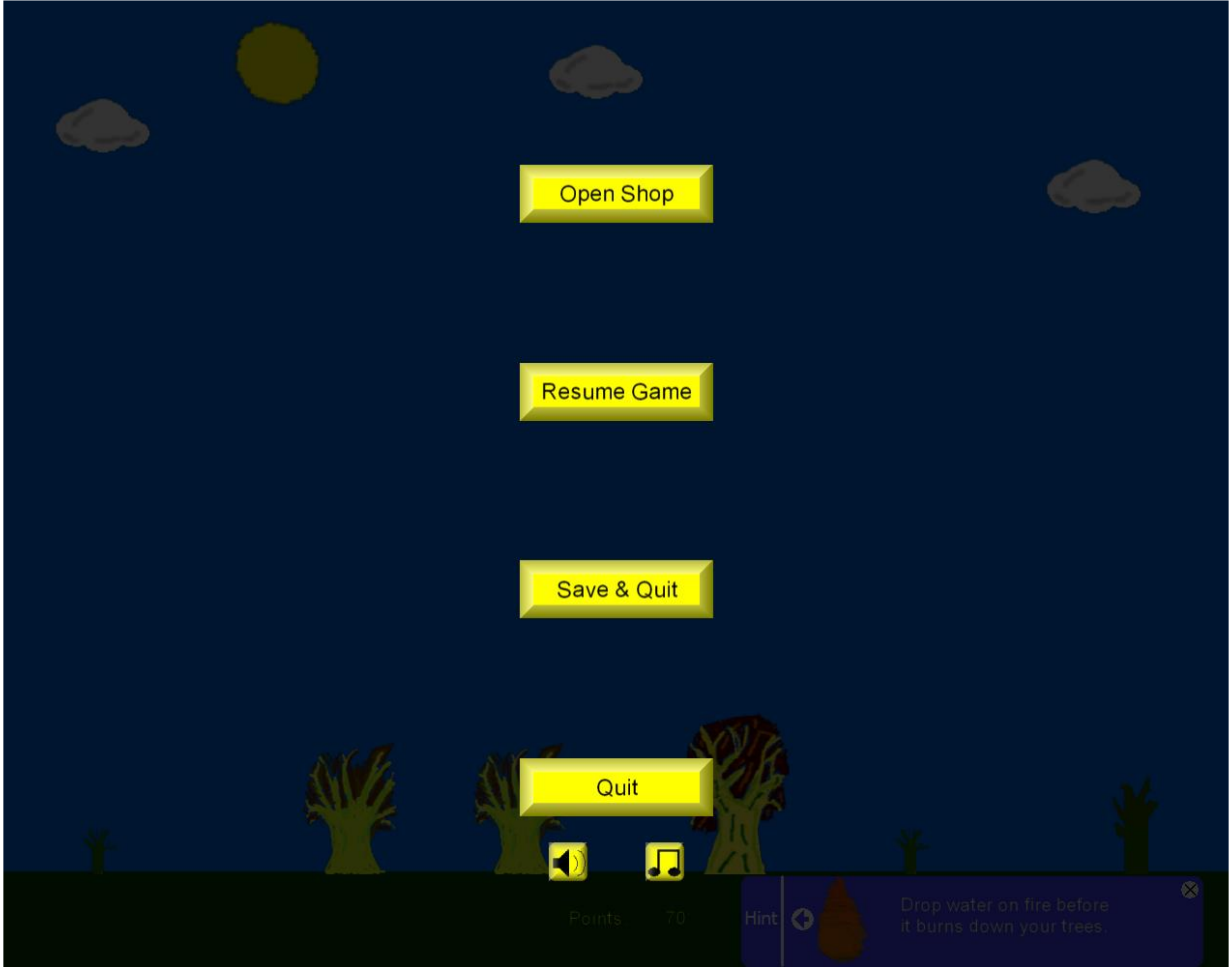
Water Drops- Spawn when player left clicks on the screen. If it hits a tree, the tree gets water

Raincloud- Spawns when a blue tree is fully grown and its fruits are picked

Points- When a tree is fully grown it grants points. When a fruit is picked it grants points. When a lumberjack is killed, it grants points. The points UI element displays the current amount of points of the player.

Module - In-Game Menu

The in-game menu (Figure 7) pauses the game at the current state and displays the game screen as a darkened background. The player can enter the shop from here or access game functions.



Points 70

Drop water on fire before
it burns down your trees.

Figure 7. In-Game Menu screen in gameplay

Menu Components

Background- Darkens gameplay screen to indicate that the game is paused

Shop button- When clicked on, the game enters the shop menu

Resume- Button that resumes the game and closes the menu

Save & Quit- Exits the game and saves the current points and upgrades

Quit- Exits the game without saving the game

Sound- Button that mutes sound effects

Music- Button that mutes the background music of the game and the menu

Module - Shop Menu

The Shop Menu (Figure 8) is accessed through the shop button in the pause menu. In the shop, player can spend points to purchase upgrades that changes elements of gameplay.

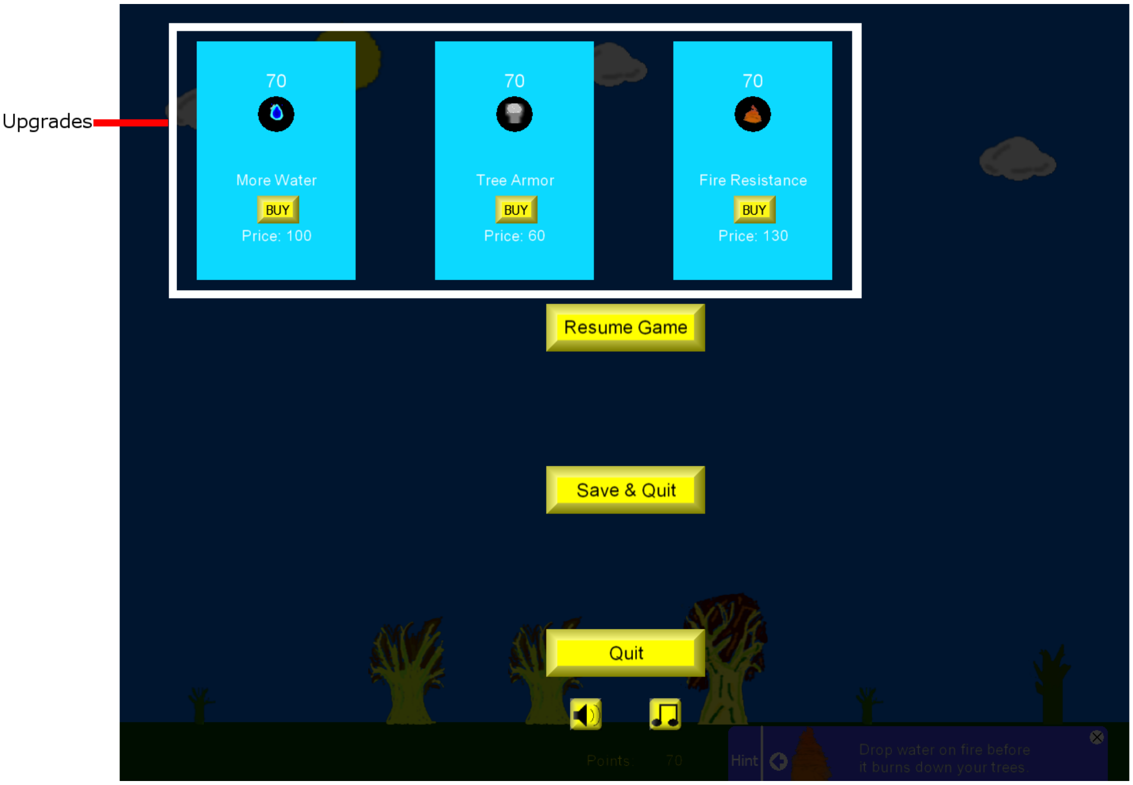


Figure 8. The shop menu screen with available upgrades

Shop Menu Components

Background- Darkens gameplay screen to indicate that the game is paused

More Water Upgrade- Increases the effectiveness of water, decreasing the amount of water needed to grow a tree

Tree Armor Upgrade- Increases the toughness of the tree, increasing the amount of time lumberjacks take to cut down a tree

Fire Resistance- Increase the fire resistance of the tree, allowing it to last longer when set on fire

Resume- Button that resumes the game and closes the menu

Save & Quit- Exits the game and saves the current points and upgrades

Quit- Exits the game without saving the game

Sound- Button that mutes sound effects

Music- Button that mutes the background music of the game and the menu

Module - Victory Screen

When the player wins the game, there is a suitable reward, a victory screen that emphasizes the player's accomplishments. (Figure 9)

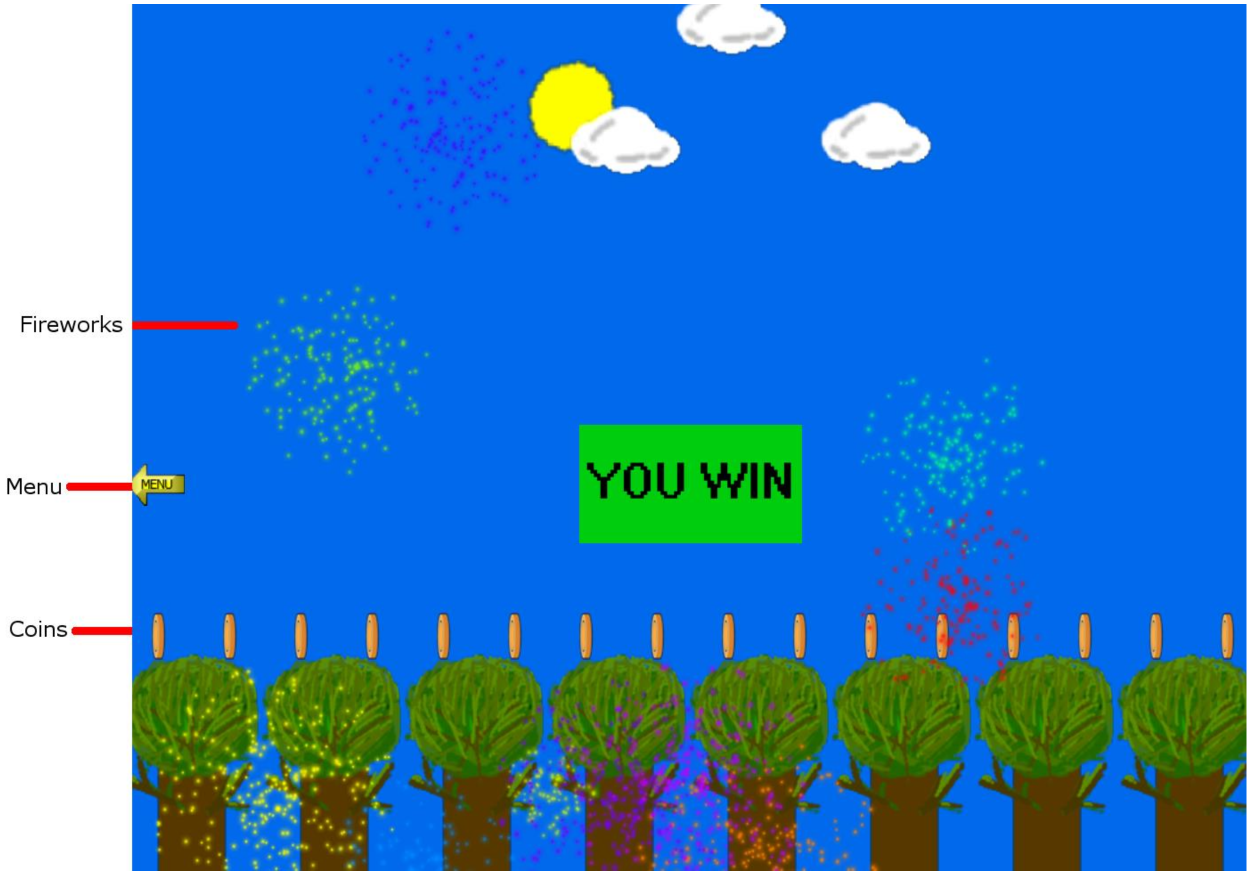


Figure 9. This displays victory screen and animations.

Victory Screen Components

Background- Animated sun and clouds with a bright blue sky

Fireworks- Animated firework explosions of random colors and positions, reminiscent of a celebration

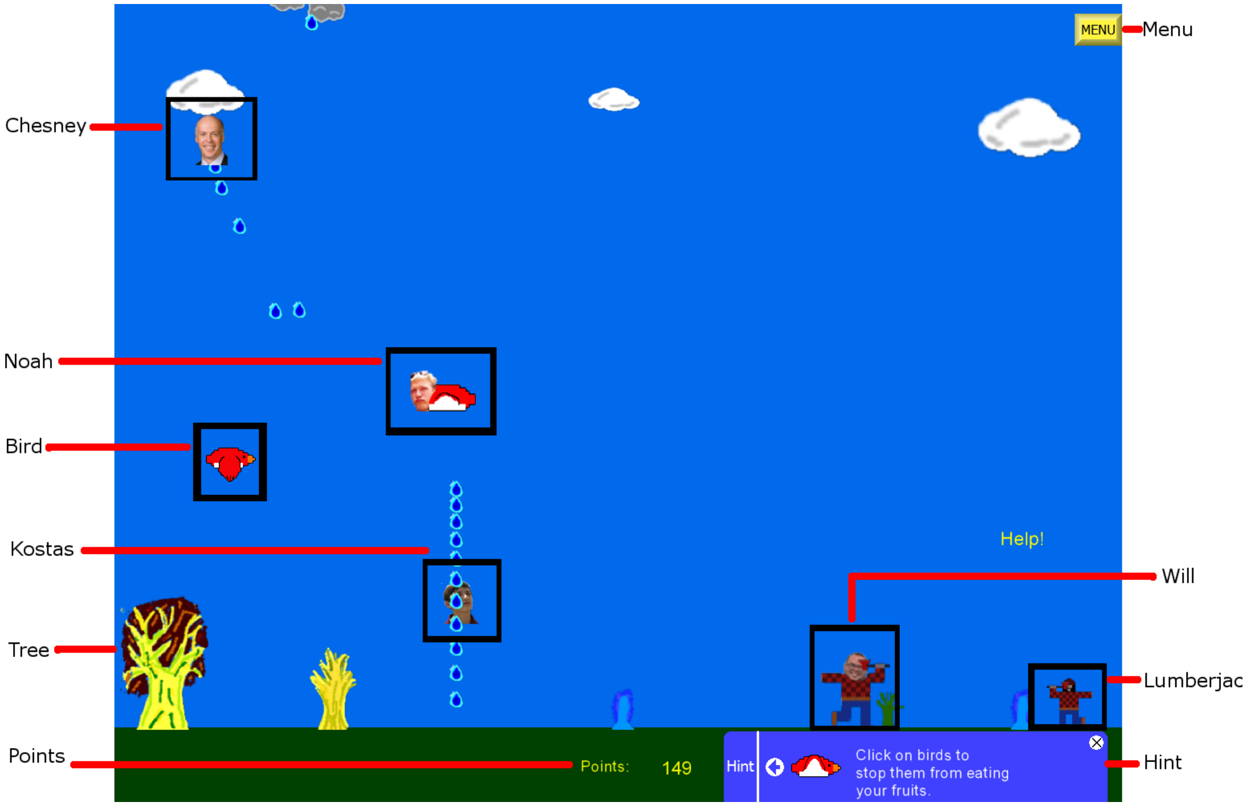
Menu-Button that returns the player to the main menu

Coins- A representation of the riches the player has earned by growing trees

Trees- A representation of all the trees the player has grown and saved over the course of the game

Module - Chesney Mode Game Screen

For an easier gameplay experience without stress, there is a game mode that is easier than easy. This is Chesney mode where most actions are automated. Chesney mode is accessed through the credits screen. The credit screen module details the process of which cloud to click in order to access Chesney mode. (Figure 10)



Click on birds to stop them from eating your fruits.

Figure 10. Shows basic gameplay of Chesney Mode

Chesney Mode Game Screen Components

Background- Animated sun and clouds with a bright blue sky

Menu- Button that pauses the game and allows access to the menu

Trees- Object that grows when watered and when fully grown grants points. When targeted by lumberjack, the trees display a “help!” message. There are three types of trees with different attributes. Details of the color and attributes of each tree are listed in Appendix A - Sprite Index

Bird- Object that seeks out fruits and attempts to eat them. They randomly spawn and can be destroyed by clicking on them

Hint- Object that aids a player with understanding the game

Fire- Object that randomly spawns if it has not rained from a raincloud recently (not shown)

Lumberjack- Object that seeks out a tree and attempts to cut it down and deny points. They randomly spawn and can be destroyed by clicking on them

Water Drops- Spawn when player left clicks on the screen. If it hits a tree, the tree gets water

Chesney- Object that automatically drops a stream of water and follows the mouse cursor

Noah- Object that automatically kills birds

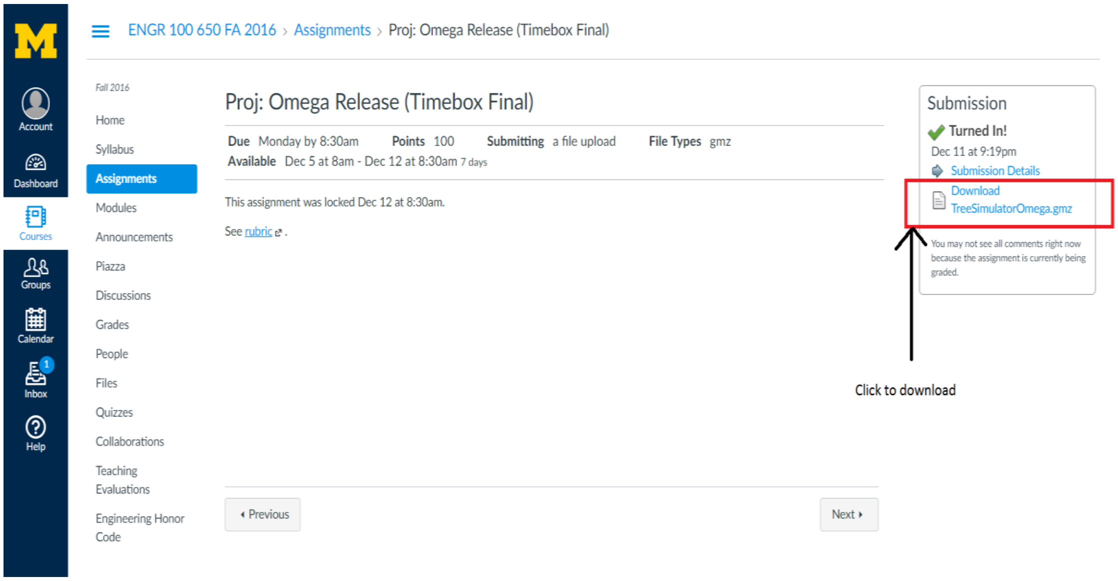
Kostas- Object that automatically picks fruits from trees

Will- Object that automatically kills lumberjacks

Points- When a tree is fully grown it grants points. When a fruit is picked it grants points. When a lumberjack is killed, it grants points. The points UI element displays the current amount of points of the player.

Installation Instructions

Navigate to the Canvas website for University of Michigan. In the Dashboard, click on the window for ENGR 100 650 FA 2016, next select the Assignments tab. Under Past Assignments, select “Proj: Omega Release (Timebox Final)”. On the next screen, click “Download TreeSimulatorOmega.gmz”, and save the file to a convenient location. Depending on user permission, the screen for downloading might vary. Figure 11 shows a screenshot of the download screen from a team member’s perspective.



ENGR 100 650 FA 2016 › Assignments › Proj: Omega Release (Timebox Final)

Proj: Omega Release (Timebox Final)

Due Monday by 8:30am **Points** 100 **Submitting** a file upload **File Types** gmz
Available Dec 5 at 8am - Dec 12 at 8:30am 7 days

This assignment was locked Dec 12 at 8:30am.

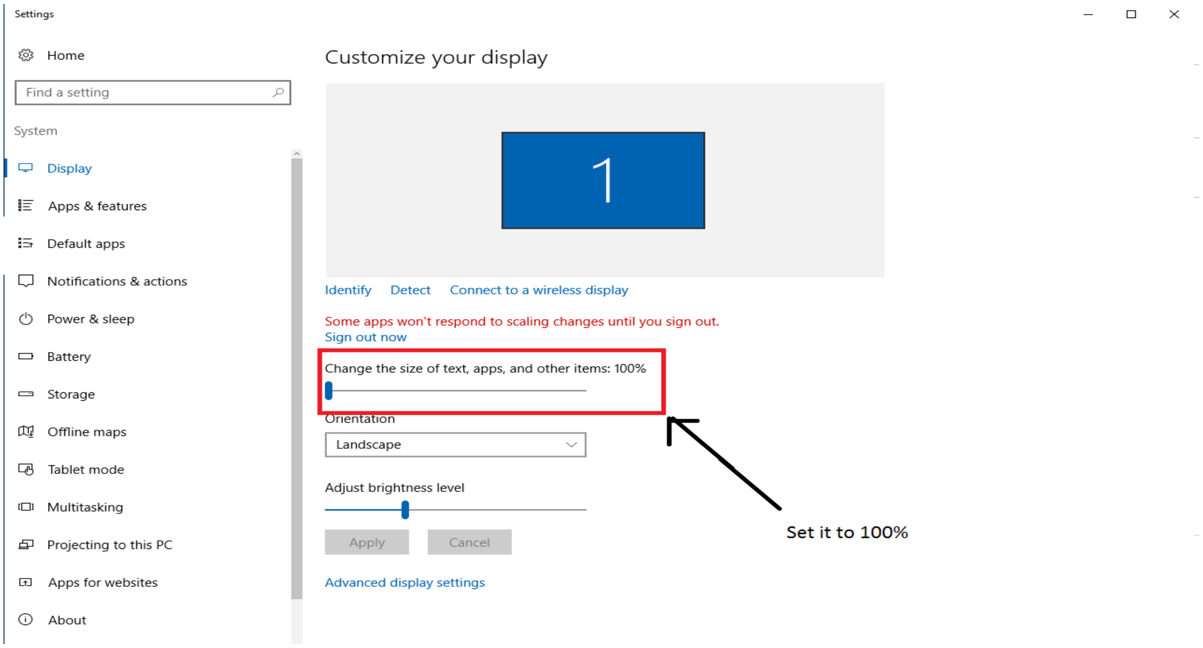
See rubric .

Submission
✓ Turned In!
Dec 11 at 9:19pm
Submission Details
Download TreeSimulatorOmega.gmz

You may not see all comments right now because the assignment is currently being graded.

Figure 11. Screenshot for downloading the game from Canvas

Return to the Windows desktop, right click on the desktop and select “Display Settings” on Windows 10 (settings may differ for other versions of Windows). Underneath “Change the size of text, apps, and other items:” adjust the slider such that it is at 100%. This step is crucial as it ensures the game is functional in the display. This step is shown in Figure 12 below.



Customize your display

Some apps won't respond to scaling changes until you sign out.
Sign out now

Change the size of text, apps, and other items: 100%

Orientation
Landscape

Adjust brightness level

Figure 12. Screenshot for setting the screen display

Navigate to the download folder, and double click on the downloaded .gmz file (the installation instruction assumes that the user has correctly installed GameMaker onto their computer). This will create a new folder at the current directory and open up GameMaker. In the GameMaker interface, simply click the green play button on the top bar, as shown in Figure 13, to start Tree Simulator 2016.

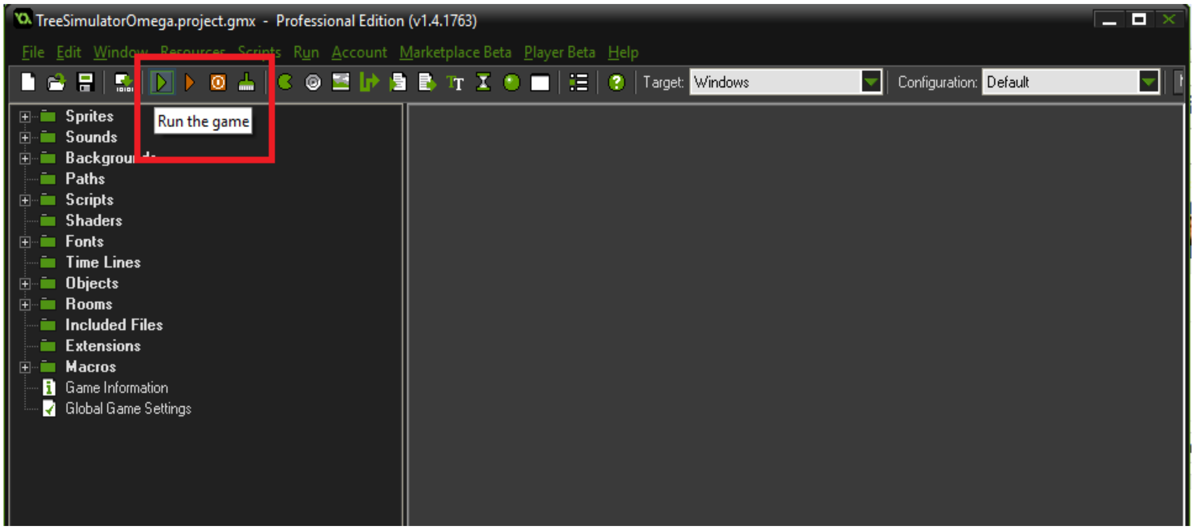


Figure 13. Screenshot for running Tree Simulator 2016 in GameMaker

System Requirements

Recommended Requirements

OS: Windows Vista, 7, 8, 10
Memory: 2 GB RAM or better
Storage: 1 GB available space or better
Monitor: 1920x1080 resolution
Processor: 2GHz dual core or better
Additional Requirements: Mouse or other input device, Gamemaker v1.0 or later

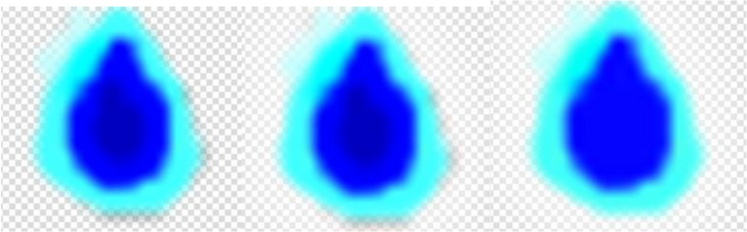
Appendix A - Sprite Index

Appendix A is an index of most sprites used in the game. It includes the pictures and descriptions of the sprites. The appendix will also describe the attributes and operations of the objects using each sprite in the game. All sprites are created by the team using the sprite editor in GameMaker.

The sprites are indexed and grouped in a logical order by their functionality and importance in the game. The "spr_" prefix will be omitted to avoid redundancy and improve clarity. Some sprites have irregular names that include the suffix "_new". The suffix has no special meaning; it exists due to merging conflicts or sprite update in older versions of the game. The suffix is not removed because such action might cause sprite reference errors in multiple objects.

drop_new, drop_new_1, drop_new_2

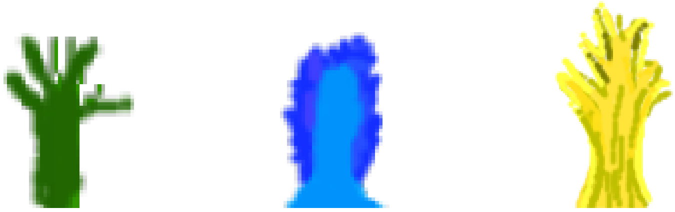
The drop sprites are used by the water drop object. It is spawned when the player left clicks on the main game screen.



The drop sprite starts from the top of the screen and accelerates downward. When the drop comes into contact with the ground or a growing tree, it disappears and emits a droplet sound effect. The tree's growth will be advanced by the water value of the drop. The water value can be increased by buying the "More Water" upgrade in the shop. The default sprite for the drop is drop_new. As the water value increases, the sprite changes to drop_new_1, which has a glowing effect around the droplet (the effect is not very visible against an empty sprite background, it is more apparent against the blue background of the game). Higher water values will change the sprite further to drop_new_2, which glows the most out of the three sprites.

sapling_new, blue_sapling, yellow_sapling

The sapling sprites are the default sprites assigned to obj_<color>_tree_1 with the corresponding colors. The sapling sprites are spawned at a set rate randomly along the ground level.



When the sprite collides with water droplets, it will increment the growth value of tree_1 objects. Green sapling sprites are spawned with the highest probability, followed by yellow sapling sprites, and the blue saplings are the rarest sprites. Sapling sprites can be set on fire. When targeted by lumberjacks, the sprite will display “Help!” above itself.

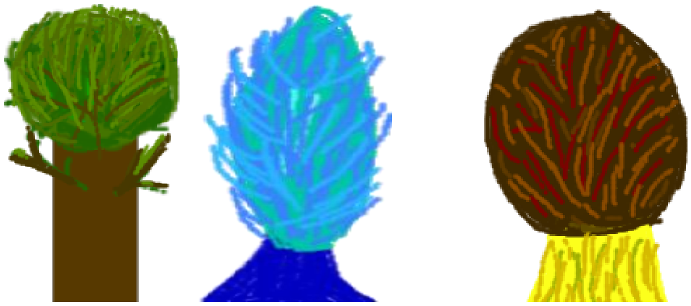
old_sapling_new, blue_old_sapling, yellow_old_sapling
The old sapling sprites are the secondary sprites assigned to obj_<color>_tree_1 with the corresponding colors. Old saplings sprites are larger in size than sapling sprites.

When the growth value of the tree_1 object reaches 2, the tree’s sapling sprite changes to the old_sapling sprite at the same position. Then the growth value is reset to 0. Old sapling sprites can be set on fire. When targeted by lumberjacks, the sprite will display “Help!” above itself.

young_tree_new, blue_young_tree, yellow_young_tree
The young_tree sprites are the tertiary sprites assigned to obj_<color>_tree_1 with the corresponding colors. Young tree sprites are larger in size than old sapling sprites.

When the growth value of the tree_1 object reaches 4, the tree’s old_sapling sprite changes to the young_tree sprite at the same position. Then the growth value is reset to 0. Young_tree sprites can be set on fire. When targeted by lumberjacks, the sprite will display “Help!” above itself.

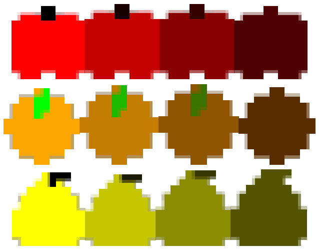
tree_new, blue_tree, yellow_tree
The full tree sprites (shown on the next page) are the assigned to obj_<color>_tree with the corresponding color. Full tree sprites are the largest sized sprites for all tree sprites.



When the growth value of tree_1 object reaches 6, the tree_1 object creates a tree object at the same location, and then it destroys itself. Tree sprites cannot collide with water drops. Tree sprites can be set on fire. When targeted by lumberjacks, the sprite will display “Help!” above itself. The tree objects are destroyed when there are no more fruits left on the tree, or when it is killed by lumberjacks or fires.

fruit_1, fruit_2, fruit_3

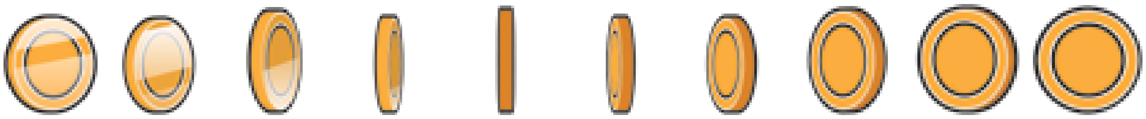
Fruit sprites are assigned to corresponding obj_fruit<number>, and they appear on trees.



Fruit_1 spawn on green trees, fruit_2 spawn on blue trees, and fruit_3 spawn on yellow trees. Three fruits spawn on an imaginary circle of random radius centered at the tree. As time increases, the fruits advance their sprites, meanwhile the points given for collecting the fruits decrease. If the tree is destroyed, the fruits will disappear as well. The fruits can be targeted by bird objects. Once all three fruits are collected, the tree the fruits grow on disappears.

coins

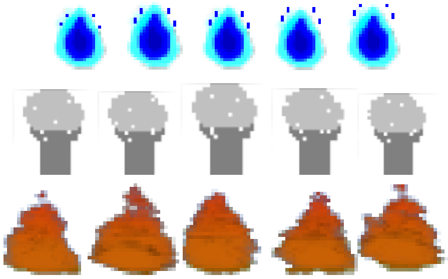
The coins sprites are assigned to obj_coin. Obj_coin is created above the Buy button in the shop when it is pressed. It is also shown on the victory screen.



The sprites of the coin create a short animation for buying an upgrade. It serves as a visual indicator for applying the upgrade. The coin disappears after one loop of the animation. On the victory screen, the coin objects add to the aesthetics as well. The coin sprites here remain permanent on the screen.

drop_icon, tree_armor_icon, fire_icon

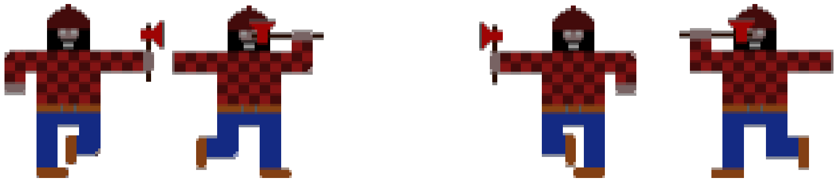
The <upgrade>_icon sprites are assigned to obj_shop_icon. They are created in the corresponding upgrade slots in the shop menu of the game.



The shop_icon object uses the appropriate sprite from the icon sprites for each of the three shop upgrades. They display particle effects for the drop and tree icons, and animation for the fire icon. The icon sprites help add a visual representation of the upgrades in the shop.

lumberjack, lumberjack_left

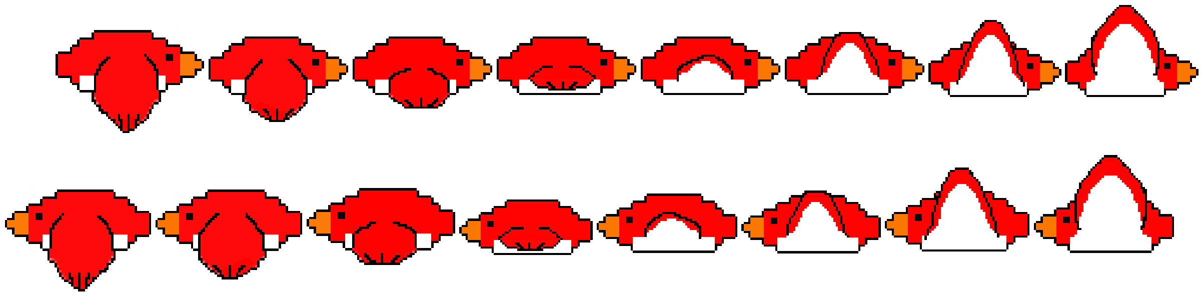
Lumberjack sprites are assigned to obj_lumberjack or obj_lumberjack_left, depending on the direction of movement. They are spawned according to a timer and difficulty settings.



Lumberjack starts from the left side of the screen near the ground level and moves right, lumberjack_left moves left from the right. The two sprites of each object creates a continuous animation as the lumberjacks move. When the lumberjack is spawned, it searches for a tree that is not yet targeted by another lumberjack. The lumberjack will stop in front of its target and destroys the tree in 2 seconds. Then the lumberjack continues moving out of the screen, without destroying other trees. If the tree is destroyed before the lumberjack arrives, the lumberjack will not target another tree. Clicking on the lumberjack sprite will destroy it and give the player points.

bird, birdleft

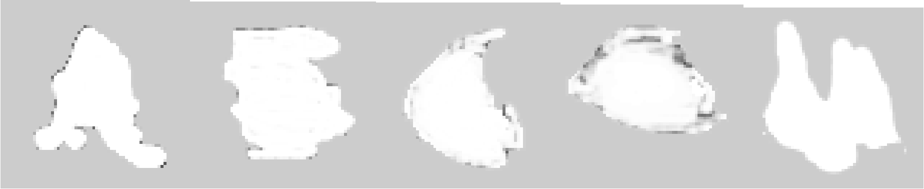
Bird sprites are assigned to obj_bird or obj_birdleft, depending on the direction of movement. They are spawned according to a timer and difficulty settings.



Bird sprite starts from the left side of the screen in the sky and moves right, birdleft moves left from the right. The sprites of each object creates a continuous animation as the birds move across the screen. When the bird is spawned, it searches for fruits on the screen. Then the bird will move directly towards the fruit at a faster movement speed and animation speed. It destroys the fruit immediately. Then the bird flies upward and continues moving out of the screen, without destroying other fruits. If the fruit is collected by the player or eaten by another bird before the bird arrives, it will not target another fruit and flies away. Clicking on the bird sprite will destroy it and give the player points.

fire

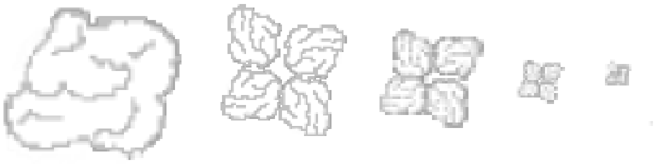
The fire sprite is assigned to obj_fire. Obj_fire is spawned at a random tree after enough time has passed since the last collection of blue tree fruits.



The fire sprite is assigned to obj_fire. Obj_fire uses each of the five individual fire sprites in a particle system, part_fire. The sprites are randomly drawn shapes with random shading. The system continuously emits fire particles. The sprites will change size, speed, and color as they move upward to mimic the motion of flames. The fire will destroy the trees after a certain amount of time. Once enough water is dropped onto the fire, the sprites stop spawning.

poof

The poof sprite is assigned to obj_poof. Obj_poof appears when a lumberjack or a bird object is clicked on by the player.



The sprites make up a short animation when the enemy is destroyed as a clear indicator to the player. The poof sprite is created at the last location of the enemy. The sprite disappears after one full cycle of animation.

raincloud

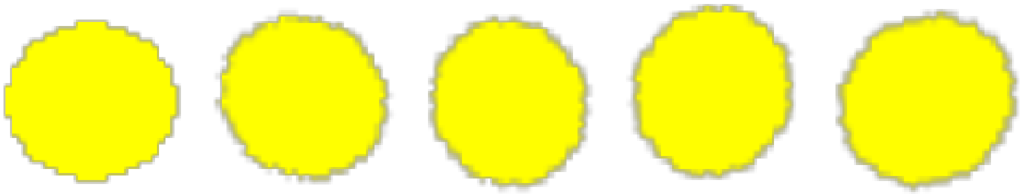
The raincloud sprites are assigned to obj_raincloud. Obj_raincloud spawns on the top left of the screen when all fruits on a blue tree are collected by the player.

The raincloud sprites move across the top of the screen to the right. It is animated with the given sprites. The object helps the player by dropping water in short intervals. The water dropped by the raincloud has the same behavior as the water dropped by players. The raincloud is destroyed when it goes outside the game screen.

cloud_new
The cloud sprite is assigned to obj_cloud. Three obj_cloud are placed within the game room during creation.

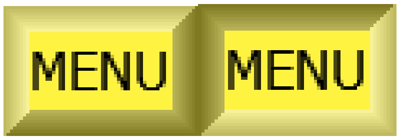
The cloud sprite moves across the top of the screen to the right. When the entire cloud sprite becomes outside the game screen, it will warp back to the left end and reenters the screen. It serves as part of the soothing background in the game. Additionally, it is the sprite for obj_fakecloud, which is used to hide the entrance to Chesney Mode in the credits room.

sun_new
The sun sprites are assigned to obj_sun. Obj_sun is placed within the game room during creation.



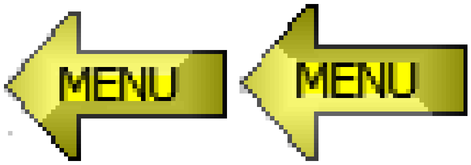
The sun object moves across the top of the screen to the right. When the entire sprite becomes outside the game screen, it will warp back to the left end and reenters the screen. The sun sprites create a spinning animation as part of the soothing background.

menu
The menu sprites are assigned to obj_menu. Obj_menu is placed on the top right of the game room during creation.



The menu object is used to access the pause menu. The default sprite for the button is the first of the two sprites. When the player hovers the mouse over the button, the sprite is updated to the second one. The sprite changes back to the default one once the mouse leaves the button. The menu object pauses the state of all in game objects, it darkens the background and creates the buttons for the pause menu.

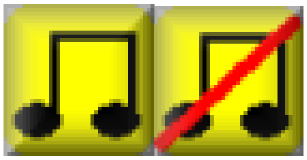
back
The back sprites are assigned to obj_back. Obj_back is placed in the top left of instruction screens, credit screen, and difficulty select screen.

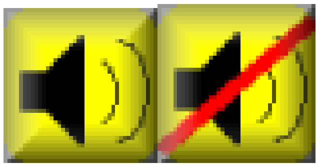


The back object is a button used to access the main menu. The default sprite for the button is the first of the two sprites. When the player hovers the mouse over the button, the sprite is updated to the second one. The sprite changes back to the default one once the mouse leaves the button. Clicking on the button will bring the player back to the main menu room.

music, sound

The music and sound sprites are assigned to obj_music, and obj_sound, respectively. They are spawned near the bottom of the screen by the game controller.

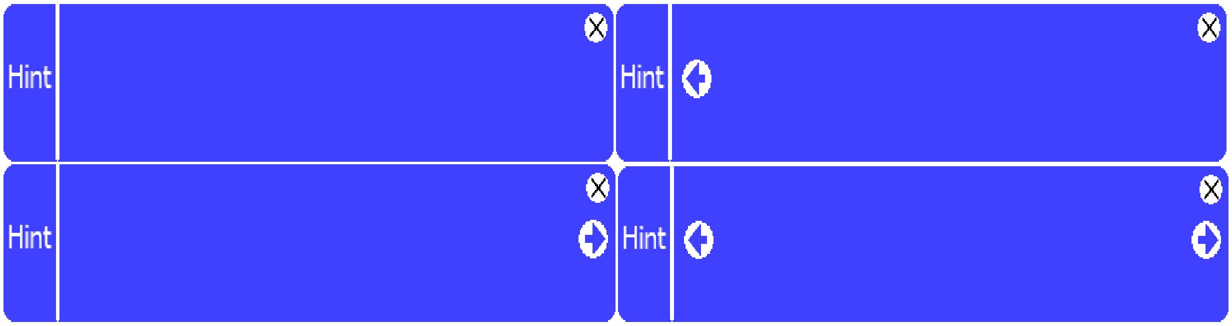




Obj_music and obj_sound control the sound effects in the game. When the game begins, the sprites are invisible to the players. The sprites become visible only when the game is paused using the Menu button. Clicking on the music sprite will either mute or unmute the in game music and shop music, depending on the current state of the sprite. The current sprite will be replaced by the other sprite at the same time. Clicking on the sound sprite will mute the water drop and tree growing sound effects. The sound control buttons become invisible once the player resumes the game.

hint

The hint sprites are assigned to obj_hint. Obj_hint is created at the bottom right of the in game screen.



The hint object is used to display text and pictures that help the player progress in the game. When new objects show up for the first time, such as enemies or fruits, the hint object will create the corresponding sprite and help text. The four sprites in the hint object are used to indicate the position of help hints, so that players can read previous hints. The sprites can be hidden by pressing the X at the top right. The hints will show up again once a new hint comes up.

purpoise

The purpoise sprite is assigned to obj_purpoise. Obj_purpoise is spawned hidden behind a fake cloud in the credit screen. The source of the sprite is the Canvas thumbnail image for ENGR 100-650, Gaming for the Greater Goods.

The purpoise object can be clicked on once the fake cloud object in front is destroyed. Clicking on the purpoise sprite will automatically start the Chesney Mode room.

Appendix B - Example Script

```
/**************************************
*Name: obj_controller_Create_1
*Author: Davis M, Aris M, Michael W, Haoran X
*Date: 11 Dec 2016
*Purpose: Initialization of essential game mechanics and variables
*Usage: When the controller is created this code will run
***************************************/
randomize(); //randomize random seed
instance_create(random_range(100,room_width-100),908,obj_green_tree1);//creates the first sapling in a random location
globalvar safe_tree; //counts how many trees are not targetted
draw_set_color(c_black);
safe_tree = 1;
globalvar tree_spawn;
tree_spawn = 30;//sets spawn rate of trees
tree_spawn_update1 = false;
tree_spawn_update2 = false;
globalvar can_water;
can_water = true;//allows player to drop water
alarm[0] = tree_spawn;//sets delay until next spawn
//sets up score display system
globalvar current_score;
current_score = 0;
alarm[2] = 5; //frame rate of score increment
instance_create(768,910,obj_hint);
score = 0;

globalvar tree_armor;//sets tree armor
tree_armor=0;

//fire particle setup
global.part_fire = part_system_create(); //creates particle system
part_system_depth(global.part_fire, 10); //makes particles go behind objects
global.part_fire_t = part_type_create(); //creates particle type
part_type_alpha1(global.part_fire_t, 0.5); //sets transparency
part_type_colour2(global.part_fire_t, make_color_rgb(200,99,5), c_red); //color shift of fire from orange to red
part_type_direction(global.part_fire_t, 90, 90, 0, 18); //fire goes up
part_type_life(global.part_fire_t, 50, 60); //how long each flames survives
//part_type_blend(global.part_fire_t, 1);
part_type_speed(global.part_fire_t,2,8,-.08,0); //how fast fire goes up
part_type_sprite(global.part_fire_t, spr_fire, 0,0,1); //uses fire sprite
part_type_size(global.part_fire_t, 3,3.5,-0.05,0); //size decreases

globalvar water_value;
water_value = 1;//sets the amount of growth each drop of water grants
mask_index = spr_drop_new; //assigns mask to object for collision checking

//sets up sound system
audio_play_sound(snd_background1,0,true);
```

```
audio_sound_gain(snd_background1,0.5,0);
//audio_play_sound(snd_menumusic,0,true);
audio_sound_gain(snd_menumusic, 1, 0);
globalvar sound_effect;
sound_effect = true;
global.musicpaused = false;
global.soundpaused = false;

//loads shop script
script_execute(scr_shop);

globalvar fire_resist;//initializes the third upgrade, fire resistance
fire_resist = 0;

//initalizes load
if(load = true)
   script_execute(scr_load);

//initializes conditions for fire event
globalvar recent_rain;
recent_rain = false;

//initializes buy button settings
globalvar buy_click;
buy_click = false;
globalvar chesney_mode;
chesney_mode=false;

alarm[4] = 300;//lumberjack alarm
lumb_spawn = 0; // decides which event in timeline to use
alarm[5] = 600;//bird alarm
bird_spawn = 0; // decides which event in timeline to use

//sets up difficulty settings
globalvar diff_mult;
diff_mult = (difficulty+1)/2;//adjust difficulty based off selection
diff_mult = diff_mult*((score+150)/150);//adjust difficulty dynamically with the current score of the player
```