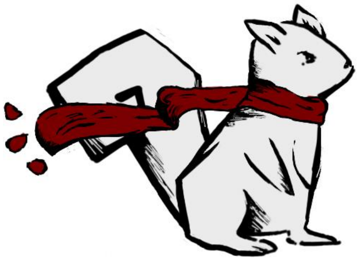
Nutty Adventures



A Senior Project

presented to

the Faculty of the Computer Science & Computer Engineering

California Polytechnic State University, San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science

by

Tony Hsu, Mandy Chan

June, 2015

Table of Contents

Abstract

This senior project's goal was to learn Unity by designing a 2.5D game with the game engine. Being a small team of two people and both being new to Unity, proved to be a challenge. Using Unity and learning about its quirks had its benefits and downfalls. Other decisions made at the start of the project seemed like a good idea at the time, but also introduced some problems later on.

Definitions

The Prince - the main character of the game
The (active/primary) Squirrel - the squirrel that the player is controlling at the moment
The (inactive) Secondary Squirrel - the squirrel that the player is not controlling at the moment
Squirrel-mate - a squirrel companion
Acorns - the primary resource collected in the game
Home Tree - the squirrels' home; where the acorns are deposited
Squirrel Kingdom - the kingdom that the Prince used to live in
Day - a time limited level or stage
Winter - the end goal (season) of the game

Introduction

Project Overview

Nutty Adventures is an Android game. The goal is to gather acorns for day-to-day survival, upgrades, and hoarding. Ultimately, the player must have enough acorns by the time Winter comes to fulfill the squirrel's quest: providing enough acorns for his squirrel kingdom during the barren Winter season. Throughout the days, the player has to keep track of time and monsters, which might influence player decisions.

Background Story

You are the eldest prince squirrel of a kingdom. However, your white fur made you an outcast. Rumors have spread that you're not actually from the royal line. Because of these bad rumors, your family is forced to prove your lineage by giving you a task to fulfill.

They decide to send you on a quest to provide the kingdom with enough food before winter begins. This quest, they tell you, would dispel the questions of royal lineage and prove your worth and competence in leadership (as all kings should have).

Your best friend decides to be your companion throughout the adventure. With your friend, a small sack of food, and a heavy heart, you leave your home.

Game Play

Gathering

The player runs around the field gathering acorns. Acorns are temporarily held by the squirrel. Squirrels can not pick up more acorns than they are able to carry (denoted by 'weight'). However, squirrels can drop off the acorns at his Home Tree and then continue collecting acorns. Players may also accumulate acorns from a tree. Gathering from a tree takes time and the squirrel is vulnerable to being attacked, but will have a higher payout than gathering acorns on the ground.

Survival

The player will need to avoid touching enemies as the squirrel who gets hit will be injured and unable to continue collecting acorns for the day. Each day, the squirrels get hungrier (indicated by 'fullness' decreasing). To prevent death by starvation, the player must feed both squirrels. Feeding a squirrel requires depleting some of the player's acorn stash.

Enemies

Currently there is one type of enemy, a snake. The snake roams around a small section of the map. However, if the snake detects either squirrels, it will chase after the closest squirrel until it can't detect them anymore. There was another type of enemy that would move randomly throughout the entire map, but was scrapped due to unfair deaths.

Environment

Every day, a randomly generated setting is created. However, the Home Tree is always on the bottom-left corner of the map. Players have only a limited time to collect acorns per day before night falls. If a squirrel is holding acorns at the end of the day, the acorns will be lost.

Night Time

After the end of each day, the player has the ability to feed their squirrels and improve them. Four different stats are available. Gather Rate improves how long it takes to search and gather acorns from a tree. Carry Capacity improves how much the squirrel can hold before needing to return to home. Efficiency improves how many acorns are received. Fullness determines the hunger level of the squirrel. If the fullness reaches zero, the squirrel has starved to death. Maximum fullness level is 100 where the squirrel is full. Each squirrel has his own stats that needs to be accounted for.

End Game

Win Condition

The player will win the game if they survived until Winter and the amount of acorns stashed away is enough to feed the kingdom during that season.

Lose Condition

The player loses if either squirrel has starved to death or if both squirrels are injured during acorn collecting.

Framed Insights and Opportunities (Original Plan)

Upon finalizing our game idea, we discussed which game engine or API to use. We eliminated our choices down to Libgdx, Cocos2D, Unity 3D and GameMaker.

Our original plan was to create a dynamic and highly strategic game involving 1 to 4 squirrels that the player controlled at a time. The game went through (time limited) days, where the player had to collect a certain amount of acorns by the time Winter approached.

Combat System

Originally, each squirrel could attack an enemy and deal some damage. When damage was taken, it would affect the individual squirrel's health gauge. Once the squirrel's health hits 0, it dies, unless it is the Prince. If the Prince's health decreased to 0, he would be incapacitated for the day.

Depending on the stats of the squirrel, players could decide to fight or run from a monster. If a squirrel can deal large amounts of damage, but can easily die, the player can pair it up with another squirrel that can "tank" or take the monster's damage. The player can customize their team to support his/her way of playing the game.

Multiple Squirrels

The player started with one squirrel (the Prince). However, throughout the days, the Prince might encounter other squirrels in distress. The player can choose to ignore or help them. By completing their quest, they will join you in your adventures (as a squirrel-mate). Each squirrel-mate would have different stats which might make them better in certain regards. However, the player could only select up to 4 squirrels to control every day. The rest of the squirrels stayed in the Home Tree for the duration of the day.

Having multiple squirrel-mates can prove to be very useful. Obviously, they can gather more resources than one squirrel. On top of that, though, is that the squirrel-mates can help fight monsters. They can mob a monster, killing it faster, which saves time and overall health among the squirrels. If the player only has the Prince, then they are more vulnerable to losing the game by dying from a monster. If the squirrel-mates all die from monsters, the day ends immediately and the player loses all the acorns each squirrel was holding. Nonetheless, if the player still has squirrels, they do not get a Game Over. A new day would come and the player can select new squirrel-mates in their Home Tree to control for the day. Therefore, it might be beneficial to hold extra squirrel-mates in case other squirrel-mates die.

Every day, each squirrel-mate would get hungrier. The ones at home would be less hungry than the ones foraging, but still must be fed at some point, effectively limiting the amount of squirrel-mates a player can have. The player must feed each squirrel some amount of acorns to fill their hunger bar. They must decide what the optimal amount of squirrel-mates is.

Each squirrel-mate has stats that could be upgraded using acorns. The player had to decide how much acorns they should spend on upgrades and food, as well as making sure there was an increasing surplus of acorns such that the Prince could have enough to feed his kingdom during Winter. Additionally, the player needed to decide which squirrel-mate to upgrade. If they upgraded only one or two of the squirrel-mates, it could prove disastrous if they died. Yet, equally upgrading every squirrel-mate one had would most likely be a poor decision, because the player would be wasting resources if most squirrel-mates aren't used.

Questing

When a player-controlled squirrel finds a new squirrel, they are given information on the stats of the new squirrel and what its quest is. With this information, the player makes a decision on whether or not they should accept the quest. Maybe a player likes how the squirrel looks, and wants to add him/her to their collection. Or maybe the player thinks the squirrel can be a valuable team member. Or maybe the player wants a backup squirrel-mate. There could be many reasons that a player would want the new squirrel to join his/her group.

The quests that a new squirrel could give would vary. Some quests would be gathering related -- bring the squirrel some amount of acorns. Some quests could be to fight off the monsters surrounding the new squirrel. These quests could be difficult for players to complete if they didn't have the right team. For this reason, a balanced team could be a good idea in the beginning, until the player had a sufficient amount of squirrel-mates to concentrate on a specific type of team.

Player Punishment

When a day is over, the player had to meet a minimal acorn requirement in order to fill up hunger and store up for the upcoming Winter. If you failed to meet the goal, the game would punish the player by decreasing each squirrel's stats, such as health or speed.

Playable Winter

Once Winter starts, resources become scarce. Players need to donate a given amount of nuts to the squirrel kingdom every day, akin to a tribute. If the Prince manages to survive Winter, the player wins the game. However, it is a harsh season that can test the player's ability to manage resources and assess risks.

Changes to Original Plan

Limiting Player-Controlled Squirrels to Two

We decided that having one person control 4 squirrel is quite difficult. Just controlling two squirrels is difficult enough, especially with how close the camera was to the characters. Thus, we limited the player-controlled squirrels to two.

By limiting the controllable squirrels to two, we stopped the player from having a chaotic mess in his/her hands. There wouldn't have been enough time between managing all four squirrels, to even think. The player wouldn't have time to make strategic or informed decisions. With two squirrels, the player still had enough to juggle, but won't be overwhelmed by the amount of information they had to process.

We also decided to simplify the gameplay by only having two squirrels. We thought that the player would spend more time on the "end day" screen, than actually playing. Thus, we removed having a variable amount of squirrels.

Start With Two Squirrels

To compensate that there would only be two squirrels, the player is given two squirrels at the beginning. One would be the Prince, and the other is his best squirrel-mate who wouldn't let the Prince go out into the wilderness alone. Each starts with the same stats, so that the player can decide what roles each of them will fit in.

No Quests

Because there will only be a maximum of two squirrels, the quest system had to be sacked. We were thinking about keeping it and maybe the squirrels gave out some sort of reward instead. Maybe a random squirrel or animal in distress could ask for a favor and reward the player with acorns or free stats. However, the idea never got fleshed out.

No Combat

The combat system felt a little inadequate even from the original plan. With only two squirrels, attacking seemed less desirable. We felt that the time it took for the second squirrel to come over and help would negate the time loss.

We were given feedback as well, that combat should be simplified. Instead of attacking monsters,

Incapacitation Instead of Death

Since the only controllable squirrels were the Prince and his best squirrel-mate, letting any squirrel die after being attacked seemed unreasonable. Now, each squirrel gets incapacitated for the duration of the day instead of outright dying. This balanced the game a bit and didn't result in a quick game over.

Project Outcomes and Deliverables

Many parts of the project were dropped in favor of simplifying gameplay and also because of time constraints. However, we both learned a lot about how Unity worked. We learned about animation and switching between them and we learned a bit about what goes on in the backend of Unity. Overall, this senior project allowed us to understand a game engine by creating our own game with it. We can transfer this knowledge of Unity's game engine to another game engine and learn it much quicker than our initial attempt with Unity.

It is a great asset to have learned about Unity's game engine. With the growing amount of game engines coming into the market, we feel that learning about them is beneficial in learning about the technology of today.

Functional Specification

Criteria

Operating Constraints

System Requirement: Android 2.3.1 'Gingerbread' and up

Security

There is no security measures in place since it is a single player game.

Deployment

The game will be deployed on the Play Store for any android phones that run the specified operating systems.

Legal

Unity 4 is free to use and with published titles. All sprites are custom made.

Costs

There is cost to this game.

Known Bugs

Gathering at Day's End

A small bug occurs when a squirrel is still collecting acorns from a tree when the day ends. When the day ends, the squirrel's gather time does not get reset and the tree disappears. When the day restarts, the squirrel tries to continue collecting the acorns from the (now supposedly deleted) tree. The squirrel finishes gathering the acorns, but both squirrels are now unable to gather from trees.

Due to time restraints, we were unable to fix the bug. However, we believe it is a simple fix. We're not quite sure why the squirrels are unable to gather from trees afterwards. We believe that it is because the gathering squirrel is still holding onto old tree data when the day ends. Then the new trees are added on top of the old tree data. When we check if any trees are near a squirrel, Unity throws an error (because the old trees don't exist) and doesn't check the entire list, resulting in the squirrels being unable to gather the trees.

Button Press Propagation

UI Buttons (such as the "Gather" button or the squirrel icons that switch control between the two squirrels) work. However, the squirrel sometimes moves in the direction that the player pressed when the button is pressed. This is especially apparent when the player presses the button and moves his/her finger around while the button is still pressed.

We suspect this is largely because of how Unity works with touch devices. When a finger presses down on the button, but doesn't release it, Unity doesn't notify the appropriate script that the button is pressed. This makes sense, as computers and phones react similarly. Nevertheless, the problem arises when the finger moves even just a bit. The touch input is regularly updated every frame in the game, in order to move the squirrel to the desired spot. However, because of this, Unity believes that if the player didn't press (and release) the button, it means that they should move to that spot.

This bug caused a bit of trouble because it led to the primary squirrel moving when the player wanted to switch to the secondary squirrel. It resulted in unfair deaths and player annoyance because they did not intend for the squirrel to move.

We've remedied this bug a little bit by stopping the movement of the squirrel after a button is pressed. The squirrel still faces the direction of the button and nudges itself in the direction, but it doesn't actually move to the point. The squirrel's position is updated before Unity checks the button press, so there wasn't much we could do fix it. We also decided to check whether the touch input is the same as before. If it is, we don't update the position. Essentially, the player has to retouch the screen in order for the squirrel to move. This might result in mild annoyance because of the repeated presses, but it does diminish this bug's occurrence.

Another decision we were considering was to put all the buttons to one side of the screen. We then could fence off that area as a "no touch-movement" zone. So when the player pressed any part of that screen's section, the squirrel would not move.

Collision Detection

Collision detection between two objects work. However, there is a bug that stems through partially our design choice and partially how Unity works. We didn't want the squirrel to move via velocity, which might incur awkward sliding or speeding up of movement. We wanted the squirrels to stop immediately when it reached the desired spot and move a set amount whenever it was moving. Thus, our decision was to move the squirrel some unit amount every frame. This proved to be a problem later on.

This problem was caused partially by the same problem as the Button Press Propagation bug. Touch input is checked every frame to always update where the squirrel wants to go. However, we believe that Unity checks physics collision after any updates (on-time or late), which led to this bug. It didn't help that we were moving by units instead of velocity.

Every frame, we updated that the squirrel move some unit amount. Unity thus moves the squirrel appropriately. Unfortunately, it didn't check if the squirrel had collided with an object until afterwards. It caused the squirrel to move just a tiny bit into the collision box of another object before it realized that it collided. If the player repeatedly pressed in the direction of the object, the squirrel would nudge itself bit by bit through the collision box of the object. Eventually, the squirrel can finally pop out on the other side of the object, effectively ignoring the collision detection.

This bug caused a huge problem in the beginning. While a player was pressing the screen, the movement would be updated every frame. Because the movement was updated every frame, the "nudges" of squirrel movement through a collision box were more pronounced, to the point of just phasing through the object. At that point, there was no point in having a collision box because it just looked like the squirrel went right through the object.

We've remedied this bug a little bit by checking the touch input. When the player initially touches the screen, the squirrel is moved to the point. Until the player touches the screen again, the squirrel won't update the point to move to. As stated before in the Button Press Propagation bug, this might result in mild annoyance, but it greatly diminishes these two problems.

The problem still exists, but is minimized. The player has to repeatedly press in the direction of the object to go through it. Nonetheless, it is much better than just phasing through the object like there wasn't a collision box. Also, if a player decided to abuse this, it would take longer to go through the object than it was beneficial. The only use of doing this would probably be to escape monsters, but to enter inside another object fully would possibly take too much time. The player also can't see any collision boxes, so it would be difficult to determine if they were safe inside another object.

Another problem regarding the collision detection sprouted after the previous remedy was implemented. The squirrel was entering so far into the collision box of another object that leaving in the same direction that it entered in was a hassle. The player had to press 3 or 4 times in order to leave the collision box at the very least.

We scavenged the internet for a solution. Eventually we found that Unity has a variable called "Fixed Timestep", which changes the Physics update rate. After inserting a smaller number for this variable, the physics engine checked for collisions more often. The squirrel could collide with an object and not be embedded into it by the time the physics engine checked for collisions.

Future Development

Better Gameplay

There are a few gameplay issues that should be addressed in the future.

When the snake is close enough to attack, it should do an animation before hitting. This allows the player to quickly move the squirrel away from the snake, avoiding "death". The effect would be less unfair deaths and more control of the game.

The squirrels should check for monsters nearby and indicate if there are any in proximity. It enables players to make more informed gameplay decisions like whether or not they should gather trees in this area.

Hunger could be a more prominent concern for players. Maybe the more the squirrel moves, the more hungry it is at the end of the day. Hunger could then be a limiting factor on how much the player is willing to move around the map. Additionally, the Carry Capacity stat may become more attractive to upgrade. If the squirrel can carry more, it would decrease the amount of times the player had to go back to deposit the acorns, saving time and slowing hunger.

Changing Gameplay with the Seasons

Nutty Adventures has some changing gameplay as time passes, yet we can improve on it. As days pass, the amount of acorns gatherable on the ground or in the trees diminishes. This attempts to show that Winter is coming and resources are becoming scarce.

Another way we can change gameplay is to vary the amount of acorns gatherable from each resource depending on the season. Maybe there are more acorns on the ground than in the trees during Fall, because they are dropping to the ground. The player might decide that roaming around collecting acorns from the ground is faster than from the trees. But during Spring, there are more acorns in the trees than on the ground. The result is the player constantly changing their strategy as seasons pass.

We also could have different monsters come out in different seasons. These monsters might have a different attacking pattern than others, hence switching up the player's strategies. Also, the new monsters would indicate to the player that a new season has begun.

As Winter approaches, the amount of day time should change. During summer, the time limit for a day could be at the maximum. However, during Fall, the time limit could decrease. The player has less time to scavenge and less resources available, emphasizing the seasonal change and indicate that they should gather the most in another season. If we combine this with the aforementioned hunger suggestion, it would provide players with more strategies. Players could decide to not move their squirrels in order to minimize hunger and to just wait out the day if they think they have enough acorns for the kingdom and to survive until Winter starts. Their decision can also be based on the fact that they do not want to risk going out when new monsters are around. Maybe they would rather be safer and not find out what the new monsters' attacking patterns are.

More Resources

Providing players with more resources can vastly increase the strategies open to them.

Different resources can conceivably be in the form of objects such as berry bushes or mounds of dirt. "Acorns collected" would change to "Resources collected". Acorns could be worth more than berries in terms of resources, but it takes less time to harvest berries from berry bushes. Mounds of dirt can be dug up by the squirrel, but the payout is a gamble. The squirrel can hit the jackpot, but it can also get so little acorns that it wasn't worth it.

Combining these new resources with seasonal changes can notably alter the dynamics of the game as it progresses. Each season can have better ways to collect resources than others. It would produce more strategies, which improves gameplay and fun.

More Monsters

Currently, we have a snake monster that roams around a specific area and chases a squirrel if it detects it. However, we want to expand our monster list.

An idea we had was to create a “bird” monster. A bird shadow would appear from off screen and circle around the squirrel. The player has to run away from that area before the bird swooped in.

We also wanted to bring back the randomly roaming monster, but maybe tweak some characteristics. Instead of just outright killing a squirrel, the monster could slow down the squirrel or steal a few nuts. If the squirrel isn’t active, it can indicate that the monster is in close proximity with its icon so that the player can take appropriate action.

Animation & Graphics

Animations should be smoothed out to not look as choppy and more graphics and animations should be created. These assets will greatly help the art direction of the game and improve gameplay by giving the player more variety. Many of the desired graphics were not created in the project duration largely because of time constraints.

The snake needs an attacking animation so that the player can avoid unfair deaths. There needs to be an icon indicating that a monster is in the proximity of a squirrel. An already harvested tree should have a different look indicating that the player can not collect acorns from it anymore.

A stretch goal would be to have different seasonal looks for many objects. It would enhance the gameplay by indicating that Winter is fast approaching and providing the player with a sense of urgency. For example, when Winter is about to begin, a bit of light snowfall might occur, and patches of snow on the ground are apparent. Some trees are bare, so players are unable to collect acorns from them.

User Interface

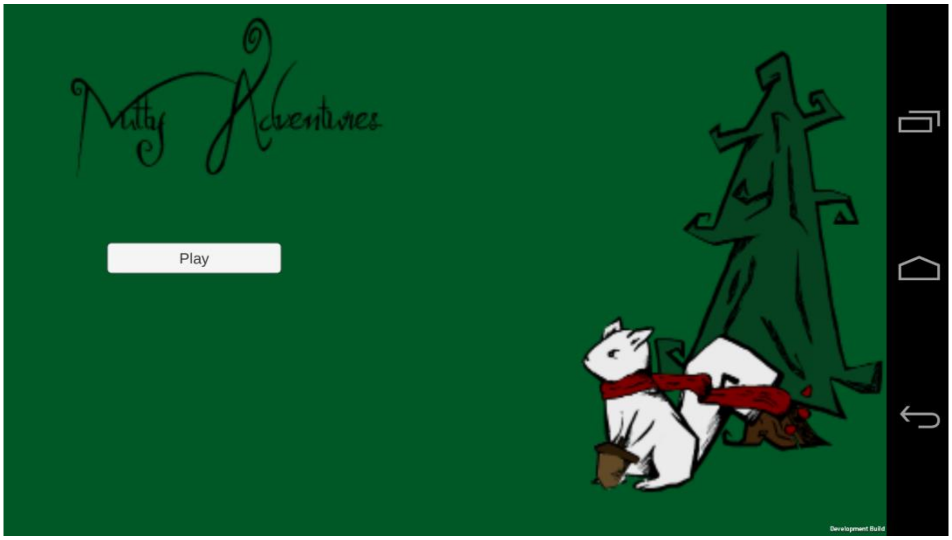
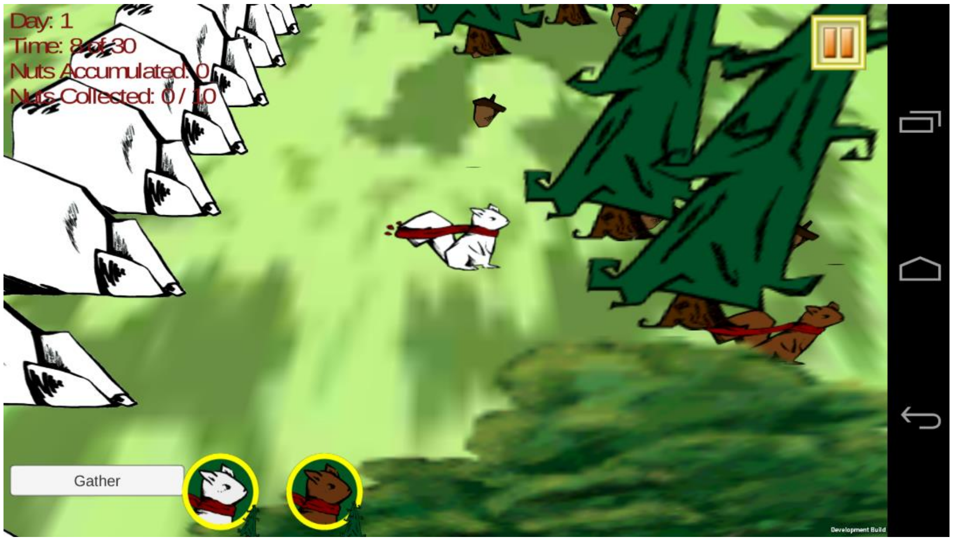


Figure 1 Menu Screen



Day: 1
Time: 8 of 30
Nuts Accumulated: 0
Nuts Collected: 0 / 10

Figure 2 Start of a Game and squirrels are next to gatherable trees

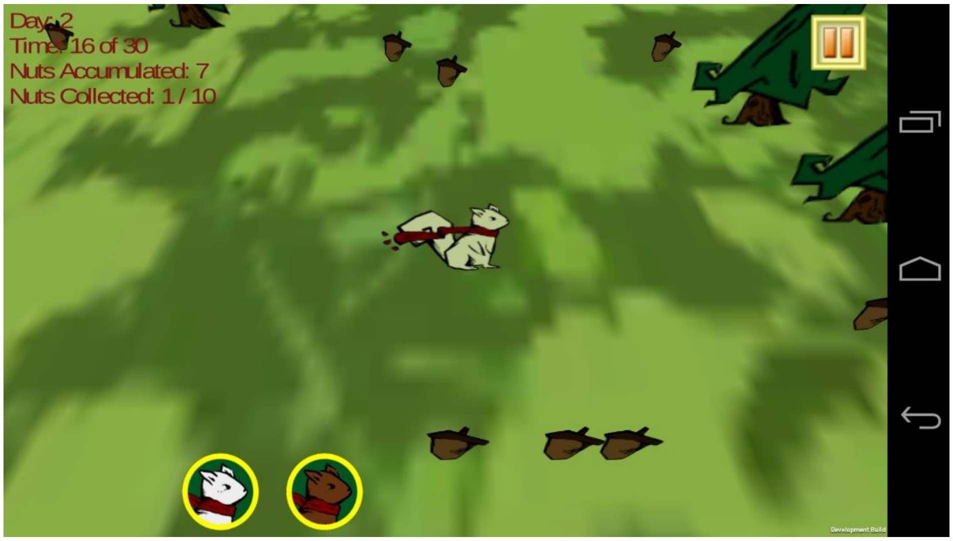


Figure 3 Lights Changed to Afternoon

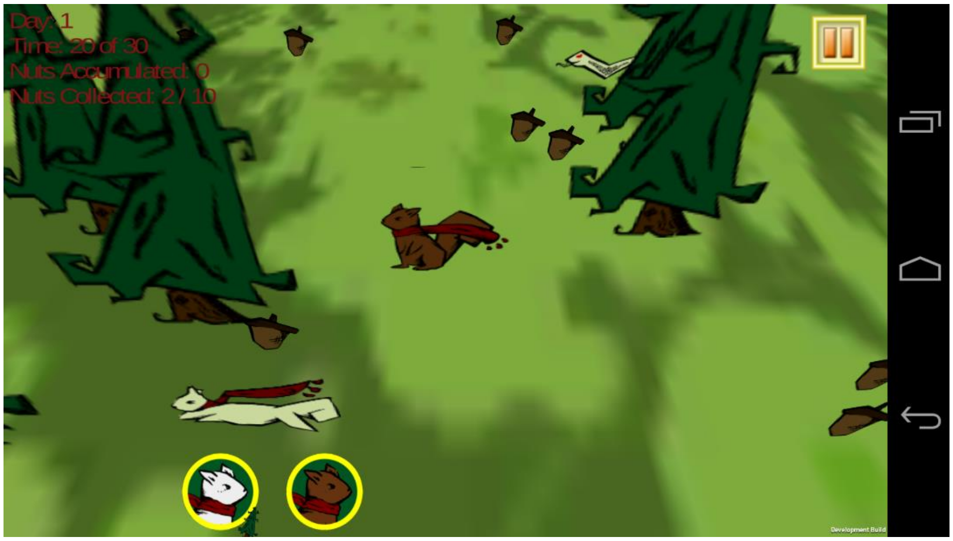
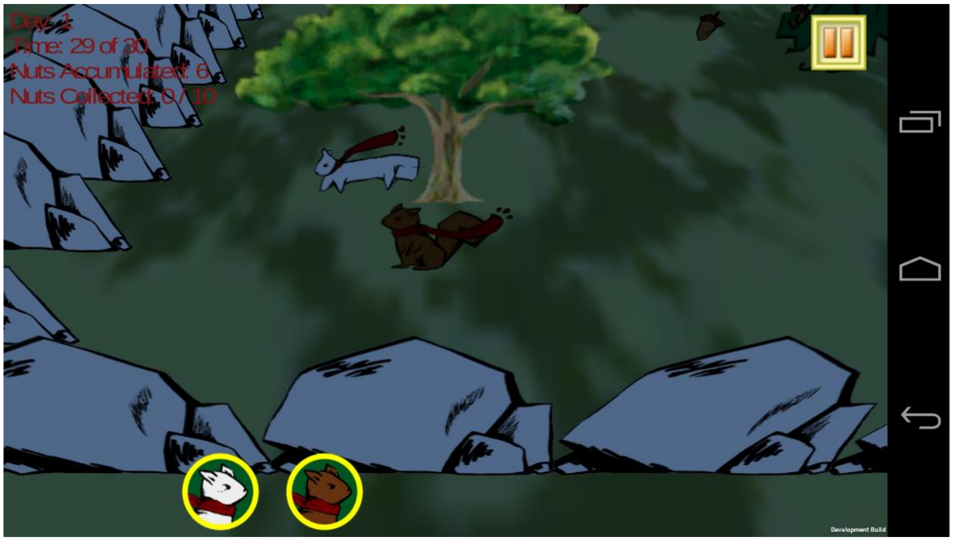
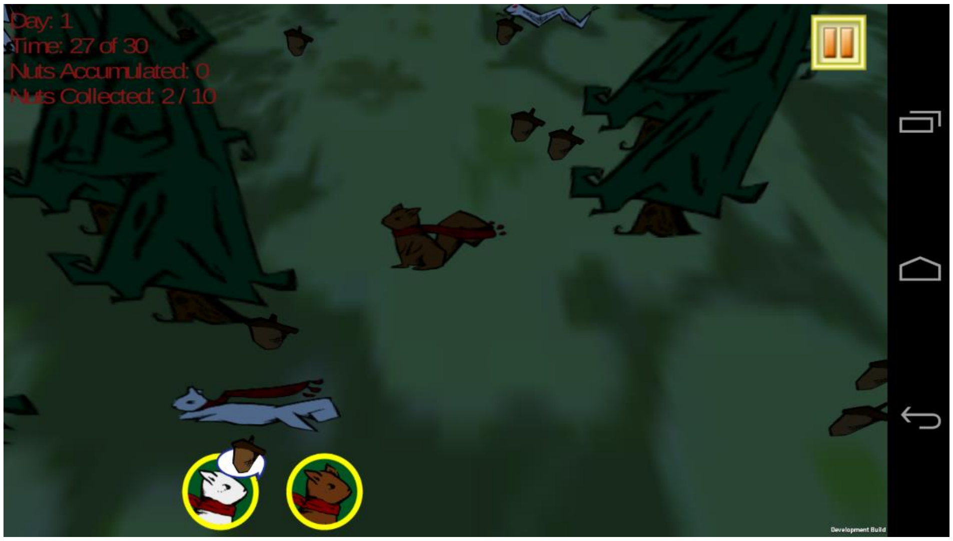


Figure 4 Enemy Spotted - Snake



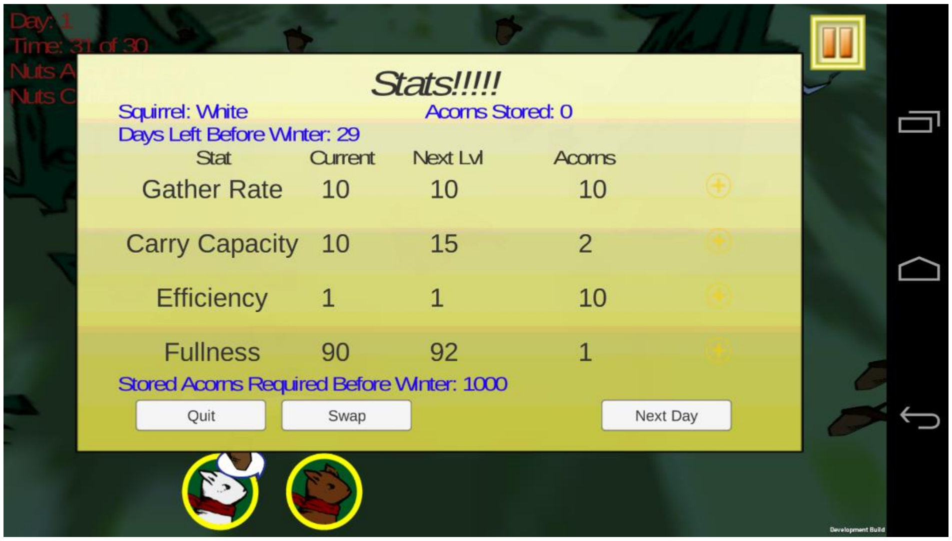
Day: 1
Time: 29 of 30
Nuts Accumulated: 6
Nuts Collected: 0 / 10

Figure 5 Lights Changed to Night



Day: 1
Time: 27 of 30
Nuts Accumulated: 0
Nuts Collected: 2 / 10

Figure 6 Still avoiding the snake at night and white squirrel has finished gathering from trees



Day: 1
Time: 31 of 30
Nuts A
Nuts C

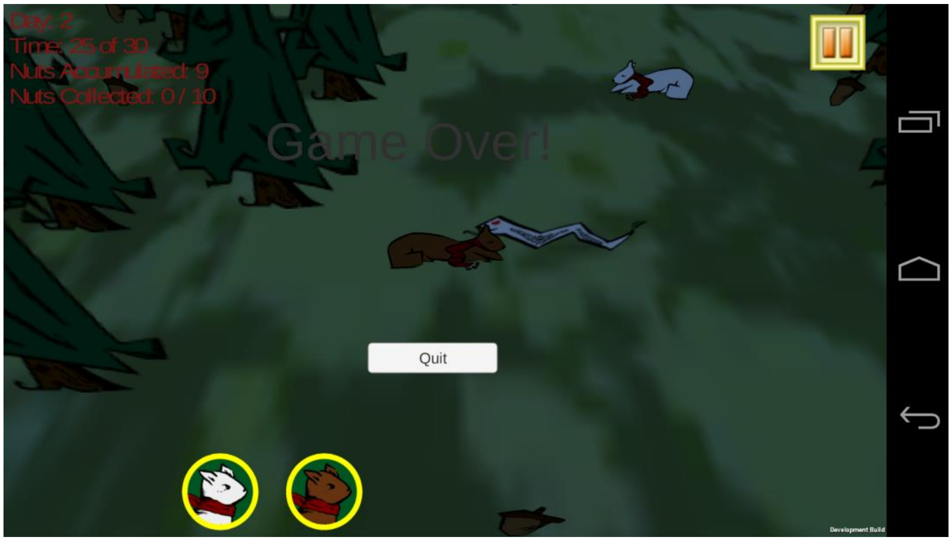
Stats!!!!!

Squirrel: White
Acorns Stored: 0
Days Left Before Winter: 29

Stat	Current	Next Lvl	Acorns
Gather Rate	10	10	10
Carry Capacity	10	15	2
Efficiency	1	1	10
Fullness	90	92	1

Stored Acorns Required Before Winter: 1000

Figure 7 Stat Score Screen



Day: 2
Time: 25 of 30
Nuts Accumulated: 9
Nuts Collected: 0 / 10

Game Over!

Figure 8 Game Over from snake killing both squirrels

Appendix

Credits

Unity3D (Version 4.X) - The game engine used to create Nutty Adventures

https://unity3d.com/

Unity Tutorials - For learning the basics about Unity

https://unity3d.com/learn/tutorials/modules

Unity Forums - For answering many questions

http://forum.unity3d.com/

StackOverflow - For also answering many questions

http://stackoverflow.com/

Michael Cummings - For a great tutorial about creating 2D animated sprites for Unity

http://michaelcummings.net/mathoms/creating-2d-animated-sprites-using-unity-4.3