Wilderness Survival: Multiplayer Survival Game Using Unity Game Engine (C#)

Todd Bertorelli, Justin Fujikawa, and Brady Thomas

California Polytechnic State University, San Luis Obispo

College of Engineering, Computer Science Department

June 2015

Abstract

The goal of this senior project was to develop a standalone real-time multiplayer survival game based on elements from other similar games and mods from the computer game *Warcraft III*. The Unity game engine was used because it has a lot of the basic game development frameworks so more time could be done designing the game rather than implementing the frameworks. Even with the Unity game engine, there was still a large amount of implementation required to provide a foundation for designing the game, and thus more emphasis was put into framework implementation. This report discusses the original vision of the game, Wilderness Survival, the incorporated elements from similar games, our design decisions, and implementation overview. The final product is a playable game with core game features.

Introduction

Wilderness Survival is a multiplayer action real time strategy (ARTS) survival game. The basic premise is two teams spawn on separate locations on an island and must scavenge their surroundings in order to establish shelter, create tools to survive, and advance their technologies. The goal of the game is to either eliminate the other team or to complete a major objective before the other team. Wilderness Survival is based on *Island Troll Tribes,* a custom modification (mod) of the real-time strategy game *Warcraft III*.

There are currently several similar multiplayer survival games, but Wilderness Survival offers a new kind of non-persistent gameplay. Rather than having a persistent, sandbox-type game like other survival games, Wilderness Survival is played in matches. Each match is designed to only last 30-40 minutes. Like the *Warcraft III* mod, after the match is over, the next match will start with no progress saved from the previous matches. Having a game where every match starts on an even playing field is a major reason for creating Wilderness Survival.

Background

A lot of thought went into choosing the correct engine for the project. The first decision was to determine if developing a custom game engine or using an existing solution fit the project.

A custom game engine seemed impractical for a variety of reasons. Developing a custom game engine would create a large development burden that wasn't directly related to the project. Features like client side prediction, interpolation, vertex skinning, navigation meshes, controls, sound, and rendering meshes are solved problems in other game engines. However, in a custom game engine all of the previously mentioned features would need to be implemented. This would result in little time left over to actually develop the game.

After researching game engines on many different platforms, only two engine options remained that seemed viable and had relevance to our java skillsets. The first option was the JMonkey3 engine, a Java-based game engine. The second option was a newer engine, Unity, in which games are developed in C#.

JMonkey3 engine had a lack of features. When inspecting the technologies that JMonkey3 supported, it lagged far behind other engines such as Unreal, Unity, and CryEngine. Almost no professional solutions use JMonkey3 as their engine, and large Java game projects such as *Minecraft* chose to develop their own engine instead. JMonkey3 did provide one notable advantage however: Java would be our project's platform, so we could benefit from our experience with Java. However, the benefit of developing in Java was not a strong enough reason to use JMonkey3.

The Unity engine has many attributes that encourage its use in new projects. First, Unity is currently being used in many professional game projects, such as *Hearthstone: Heroes of Warcraft* and *Rust*. This indicated that Unity would be suitable for collaborative development, as the companies behind these projects have sizable development teams. Second, Unity's website provides a detailed manual, code documentation, and rich set of tutorials. Using Unity would mean that the game's implementation had to be in C#, but given that the language is very similar to Java, it seemed that it would be a natural transition. After exploring the many resources available for getting started with Unity on their website, Unity presented itself as the best choice for the project's game engine.

Technologies

After choosing the Unity engine, we searched for tools that could assist in our development of the project.

Unity Game Engine

As mentioned previously, the Unity game engine provides many solutions such as model, mesh, material, and animation importing, input handling. These solutions allow us to develop at a much faster rate. We can go straight into implementing core game components on the back end without having to deal with most of the front end graphics rendering. Its vast documentation and tutorials allow for quick problem solving and its Asset Store allows us to bypass most of the artistic work that goes into making a game.

Photon Unity Networking Library

The Photon Unity Networking Library (PUN) is a networking solution implemented in Unity that supports both the peer-to-peer and authoritative server multiplayer design patterns. PUN provides a number of advantages over the base networking library shipped by Unity. First, PUN offers a free cloud service that is capable of organizing lobbies for connecting players and automatically designating them to a server. This severely simplifies the workflow for getting a multiplayer game session started, and does not require configuring a connection to the Unity Master server for a NAT punchthrough. Second, an authoritative server design is much easier to implement using PUN than the standard Unity solution.

Design

This section will go over the way that Wilderness Survival was designed to be played. It will be broken into different subsections with the design of that specific section below. The sections that game design will be broken up into are as follows: “Game Philosophy,” “Player Stats,” “Controls,” “Art,” “Combat,” “Items, Construction, and Crafting,” and “World Design.”

Game Philosophy

- Player Roles
    - We designed three essential roles that gameplay in Wilderness Survival will follow. Each has a general philosophy that was followed in the design.
    - Gathering
        - Safely acquire resources essential to survival.
    - Construction
        - Increase efficiency and survivability of the team
        - Provide items which allow access to previously inaccessible resources and objectives
    - Exploration
        - Find strategic objectives (threats/opportunities) and navigate the team to these objectives
- Combat
    - Combat is a core part of the game, but not necessarily something players will need to specialize in. Player skill, equipped items, and strategy should determine the outcome of a battle.
    - Combat events should supplement the Player Roles and make their choices more interesting.
- Survival Stats
    - Survival stats are built into each player and must be managed in order to efficiently progress through the game. Some stats are lethal when they are depleted, while some hinder the actions of the player. It is important to keep the maintenance of these stats an interesting choice for the player.
- Group and Social Structure
    - Players are organized into specific teams before the game starts in a lobby state.
    - Players may communicate via chat with their own team, or locally with other players nearby.
    - Teams are fixed until the game ends.
- Game Progression
    - Teams spawn at different locations on the island.
    - Players make choices consistent with the Player Roles design philosophy to gain an advantage over other teams.

    - Teams engage each other in combat, or a team attempts the PvE victory objective.
    - The last team standing, or first to complete the PvE victory objective wins.
    - The target game duration is 30 to 40 minutes.

Player Stats

Player states are an essential part of every survival game and the way the player manages their stats is crucial to their survival. In this section each player stat will be discussed as well as how a player gains and loses each of those stats. The four main player stats are health, stamina, hunger, and warmth. Having an excess or deficit in any of these stats can affect the regeneration or degeneration of another stat.

- Health
    - This is the basic player health, it will be decremented each time a player is damaged by an enemy or when a player is very low on hunger and/or warmth. Player health will also regenerate when the player has a high warmth stat and a high hunger stat. Once a player's health reaches zero, the player dies.
- Stamina
    - This stat represents the amount of time a player is able to sprint for, and the amount of times a player can perform a dodge or roll. This stat is decremented when a player is sprinting, jumping, or rolling. This stat is constantly increasing when it is not at full value and when the player is not using it.
- Hunger
    - This stat represents how the player is managing their food intake. If a player is well fed, the hunger stat will be a higher value. When a player has a high hunger stat they will be able to regenerate health more quickly. The hunger stat is always decreasing unless a player finds food to eat, when this is the case the players hunger stat will increase based on the food they eat. If a player's hunger stat becomes too low, the player will suffer consequences such as health loss until the stat is increased or until the player is dead.
- Warmth
    - This stat represents how warm the player is. The lower the stat, the colder the player, and the higher the stat, the closer the player is to optimal temperature. This stat will be affected by the temperature in the current area of the map the player is on. If the player is in high elevation it will be colder, also if it is night time, the player will be colder. In order to keep a warmth stat high, a player must create clothing to wear so they are able to survive at extreme temperatures. If a player has a very low warmth stat, the player's health will be affected by constantly draining until the player warms up, or the player has died. On the other hand, if a player has a high warmth stat, the player will be able to generate health until the health value is at the max.

Controls

For controls, we wanted to make them familiar to other games that share the same player view like most MMOs. We tried to minimize the number of core controls so players can easily learn them. Gameplay and efficiency were kept in mind when designing a control scheme so we changed some things that would otherwise seem normal in other games. One example is strafing with ‘a’ and ‘d’ instead of rotating the player. The rotation is controlled by dragging the mouse across the screen. It allows more efficient movements from the player since the mouse is heavily used during the game. Another example is having both “Attack” and “Interact” bound to left-click but “Attack” requires right-click (camera control) being held down, otherwise “Interact.” This setup forces the player to make use of the camera controls while in combat to make aiming more efficient.

I. Movement
    - w - Forward
    - s - Backpedal
    - a - Strafe Left
    - d - Strafe Right
    - Double tap direction - Roll
    - Right Click - Camera/Turn
    - Hold shift - Sprint

II. Actions
    - Left Click - Interact
    - Left Click w/ Right click down - Attack

Art

Since we don’t have artists on our team, most assets from were obtained from Unity’s provided standard assets and the Asset Store, where the Unity community can distribute their art. All of the art used in Wilderness Survival is freely available. When available, assets that fit the setting of the island were favored.

Items, Construction, and Crafting

The mechanic of crafting and construction is a staple part of Wilderness Survival, it is what allows the player to progress and evolve their character into the role that they plan on playing for their team. This section will go into detail the design decisions that were made when creating the different items available to the player and the means the player must go through to craft those items. This section will also discuss the buildings and place-able objects that are available for the player to craft.

The player inventory is the main way the player will manage the items that they have acquired. The inventory will have twenty-four slots with some items being stackable and some not. The items that will be stackable will mainly be the raw materials the player can gather to craft other more important items such as weapons, armor, and improved tools. The items that will not be stackable will be the items that the player will be able to use and equip. Included in the inventory, the player will also have access to an “equipped” panel. This panel will manage the gear that the player is currently wearing. The player will be able to equip one weapon or one primary “tool.” The player will also be able to equip three different pieces of armor: a head piece, a body piece, and a leg piece. The player will be able to manage their equipped items and inventory through a drag and drop user interface in-game.

As well as having an inventory panel dedicated to the player's inventory and equipped items, there will also be a UI panel available to the player that is used for crafting. To ease the crafting process, the crafting system queries the player’s inventory to check what items can be crafted based on the items currently in their inventory. This is to keep the game moving at a quick pace and to save the player from trying to figure out what combination of items will result in craftable recipe. The player will have a list of the items that they are able to craft in the crafting screen and a list of the items required to craft them. When the player clicks on an item, the resources required for it will be removed from their inventory and the item that the recipe results in will be placed in their inventory. This removes the hassle of inventory management from the player.

When designing the items and the progression of the items for Wilderness Survival, the item set should not be too complicated for a player to figure out and should be able to progress through within a 30 to 40 minute game. Following these heuristics, only one tool for each resource exists. For example, instead of there being multiple tiers of pickaxes, there is only one pickaxe used for mining metals and ores. Unlike the one-for-all method used for tools in Wilderness Survival, the weapons system will have a more diverse progression. For example, instead of being only one melee weapon, the players will be able to craft a few different melee weapons, such as a wooden club, a sword or an axe. The weapon recipe components directly correlate to their effectiveness in combat. For example crafting a sword will require resources that are much harder to obtain than materials for a wooden club. Every item that a player crafts will have some impact on the player’s progress to the end game objective. It will be up to the players to decide which items to craft to best match their strategy for beating the other team.

The last component in our crafting system is buildings and construction. As well as crafting different items, players will be able to use the resources they gather to craft buildings. The buildings that a player creates will beneficial the player’s team’s ability to survive on the island. Players will be able to craft fires: this is essential for players to keep their warmth stat at a healthy level. Crafting a storage building can keep extra resources and items for later. Walls will also be craftable and their primary use will be to protect a team’s base from the other team. For example, having walls around a base will keep players safe while they are trying to heal themselves and keep resources safe when the team is away from their base.

Combat

Combat adds another dimension to the game by allowing players to choose different paths to focus on for winning. Instead of the game being a race to see which team can complete objectives faster, combat allows teams to fight the enemy team, hindering the enemy team’s progress, or even stealing their belongings to gain advantage over them.

Combat is not a separate part of the game, but rather a mechanic that is in play at all times. At any time, players can attack each other. This allows many different strategies to utilize combat to try and gain an advantage at any point in the game. An example of utilizing combat is to rush the opposing team with simple weapons in hopes of gaining an early advantage by catching the opposing team unprepared.

Like every other element in the game, combat also has trade-offs. Gaining advantage while putting the enemy team at a disadvantage may sound too powerful but those that search for a fight must spend time to find the opponents. This time spent by the attacker is being used by the opponents to establish a base and advancing their construction tech so if the attack fails, the opponents gain even more of an advantage and the attackers will lose their time, resources, and even their lives if they are counterattacked.

The game will have different tiers of combat equipment. Low tier equipment are made of simple materials like sticks and rocks but after gathering more valuable resources, players will be able to craft higher tier equipment such as bows, swords, and metal armor. This gives incentive for both teams to spend resources to gain higher tier equipment so that they do not have to fight at a disadvantage.

World Design

World design efforts focused on creating a world that would be challenging and interesting to survive in, without losing a connection to real world locations. The island setting was placed up north, around Oregon’s and Washington’s latitude, in order for temperature to be considered a core survival stat. A northern location allows colder temperatures in comparison to other island locations, such as the tropical areas. The foliage for the inner parts of our island would also follow those found in these areas.

We also drove our world design by creating zones where players are likely to be according to the game time. Basic resources like sticks and stones were placed close to the team spawn points, and higher value resources were placed towards the center. Metals are valuable resources and are found in mountainous regions, thus mountainous regions were placed near the center of the island. Large forests also play an important role in creating a realistic environment by providing placement for animals, fruit-bearing plants, and other food sources.

Implementation

Our implementation process for the project followed a pattern of strict incremental development. We began by creating many development tasks that were categorized and didn’t depend on any other tasks. We kept these tasks organized in a spreadsheet and identified them by their category and a short description of what was required to complete it. We then spent time at our meetings expanding our task list to include more tasks that related to each of these categories. For example, we had categories for “Environment” and “Structure” related tasks. Each task was numbered based on if it required the prior task: the first task in the “Environment” category was named “Environment 1”, while the next task that depended on it in the category was named “Environment 2”. Once a week, we assigned tasks to the three of us. The categories of tasks were particularly helpful as it allowed specific members to specialize their knowledge, reducing the large burden of research that we had.

Due to time constraints, we realized that there was not enough time to implement everything. Task priorities changed to complete the core game components before working on tasks for less important components like sound effects or implementing swimming movement while underwater. Focus was placed on completing gathering, crafting, construction, and combat to proceed with play testing.

Analysis

This section will cover the analysis of our product thus far and how we can improve it in the future.

Play Testing

In order to assess gameplay and player immersion, once Wilderness Survival was in a playable state, play testing was conducted to get feedback on the mechanics of the game. Since Wilderness Survival is a multiplayer game, there were at least 6 people playing simultaneously during each play test to get the best feedback.

After players played Wilderness Survival, each was asked to fill out a short survey to help us better develop the game in the future. Questions on the survey included questions related to mechanics of the game such as: player control, combat, resource gathering, crafting, and building. More questions asked about the environment such as: the ease of navigation on the island, the placement of critical resources on the island, and the placement of teams on the island. Questions about how the user interface and what players would do to improve upon it were also asked.

Most players came back with similar feedback as one another, and the feedback that was received was very valuable to the future development of the game. Most players agreed that the player stats drained too quickly and it was hard to survive upon starting the game. Another issue that was brought up was being able to tell which teams each player was on because every player had the same player model. Players found it very difficult to tell who they should be attacking, and who they should be helping. It was also discovered that players had a hard time figuring out which resources were needed to craft certain items. Many players expressed interest in having a table that showed the craftable items in game.

Most players agreed that controlling the character was natural and useable, but that an indication that they made contact with another player when attacking would be beneficial. Navigating the inventory UI, the crafting UI, and the building UI all seemed to be intuitive for users as most expressed they had no issues with figuring out those mechanics. Play-testers also felt that the player stats UI were in a good location and easy to read, but need labels as well as colors to distinguish which bar refers to what.

Future Work

After receiving the feedback from playtesting, there are many aspects in Wilderness Survival that can be improved upon. It is clear that much of the feedback is related to the UI in the game, so this is the next logical step for improvement. The crafting UI should be expanded to include recipes for all items. Additionally, the stat bars at the bottom left of the screen should be labeled with what they represent. Gameplay could be improved by providing a better means to sustain hunger, by adding more efficient hunger recovery items to the game. Members of the same team should also have indications above their heads that display they are teamed, such as a player name tag. Finally, particle effects should be rendered to the player when they have successfully attacked another player.

Future development will also be considered to complete all of the tasks created when the game was originally designed. This includes a more complete crafting system, additional items and buildings, and improved lobby system.

Related Work

This section covers two similar games that inspired us to make Wilderness Survival. We have incorporated features from both games and will cover them below.

Minecraft

A similar game to Wilderness Survival is a well-known game called *Minecraft*. The early stages of *Minecraft* are similar to Wilderness Survival. Initial resource gathering and the crafting of basic resources and items are parts of *Minecraft* that were drawn from and put into the game. Similar to *Minecraft*, in Wilderness Survival the players are dropped onto an island without any starting resources or items. The initial goal is to then gather resources as fast as possible to survive. Like *Minecraft*, Wilderness Survival has a resource collection system that depends on the player going and finding specific resources and using a specific tool to gather it and put it into their inventory. The difference between the two being that unlike *Minecraft*, in Wilderness Survival there is only one tier of items used for collection. For example, in *Minecraft*, a player can craft a stone pickaxe, then an iron pickaxe, then a diamond pickaxe, and so on. In Wilderness Survival, there will be one pickaxe and that pickaxe will be used to mine all of the resources pertaining to mining.

Another similarity between the two games is the way a player uses the resources gathered to create new items to aid survival. If a player wants to craft a sword in Wilderness Survival they will need to find the necessary materials to create that sword and then place them together to create the new item. Players are also able to craft buildings and place them down later. The main difference with buildings in *Minecraft* and Wilderness Survival is that *Minecraft* has a lot of extra buildings for aesthetic purposes whereas Wilderness Survival focuses on how buildings can aid in progressing towards a team’s goal.

Island Troll Tribes

The *Warcraft III* custom mod, Island Troll Tribes, plays a large role in guiding design decisions for Wilderness Survival. This modification pits three teams in a free-for-all survival setting where each player selects a different class to play as. The players begin with nothing besides a limited skillset depending on what class they choose. Each team works to gather resources and food, construct protective structures, forge weapons, and ultimately, eliminate the other teams. Because it was just a mod, the game had its limitations, such as the inventory maxing out at 6 spaces, requiring owning *Warcraft III* to play it, and a bulky crafting process

Regardless of its crafting limitations, Island Troll Tribes has an extensive crafting system that we based our construction system on. The functionality of craftable items in Island Troll Tribes is the driving factor behind many of the items we include in Wilderness Survival. Such items include axes, swords, traps, and the bonfire. However, there are items that exist in Island Troll Tribes that we did not want to include in Wilderness Survival such as incantation scrolls and magic wands because of their otherworldliness.

Island Troll Tribes has different classes that are specialized to do specific tasks such as scouting, hunting, resource gathering, and fighting. We ultimately decided against such a system in favor of having everyone start on a level playing field and allow them to choose their own path of progression. Some of the classes in Island Troll Tribes are also capable of using magic to fulfill their roles like healing their allies, teleporting, and using spells to disable their opponents. Like the magical items, we decided to stay away from magical abilities because we want Wilderness Survival to be mostly realistic.

Conclusion

Wilderness Survival is a standalone multiplayer survival game created using the Unity game engine. The Unity game engine was the most appropriate development tool for Wilderness Survival because Unity provides most of the solutions and frameworks needed to create a full and complete game. Wilderness Survival incorporates ideas from similar games such as *Minecraft* and the *Warcraft III* custom map, Island Troll Tribes.

Wilderness Survival is designed to be a non-persistent game where two teams compete against each other in matches that last about 30 to 40 minutes. To reach this target time, tools were simplified to have a single tier, the crafting system lists all available items to craft, and the environment was designed so players will be most powerful around the 30 minute mark. Despite the seemingly short game time, there are many different strategies available from the moment the game begins.

Due to time constraints, only the core game components were implemented. However, many of the play-testers expressed enjoyment from the game and provided useful feedback for improving the game. Future development will be guided by the feedback and full completion of the game is possible.