



Turf Wars

Defend the Nest!

Roles	Name	Role
Project owner	Sterling Henderson	Project Sponsor
Team leader	Matthew Pflieger	Developer
Team members	Chase Holt	Documentation
	Andrew Burgos	Developer
	Hezekiah Holloman	Documentation
	John McDowell	Developer
Advisor / Instructor	Sharon Perry	Facilitate project progress; advise on project planning and management.

Table of Contents

1. Revision History.....	2
2. Game Overview.....	2
3. Gameplay.....	3
3.1 Objectives.....	3
3.2 Game Progression and Play Flow.....	3
3.3 Mission Challenge or Puzzle Structure.....	3
4. Mechanics.....	3
4.1 Turf Game Mode Mechanics.....	3
4.2 Combat Mechanics.....	4
4.3 Team/Spawning Mechanics.....	4
5. Game World.....	4
6. Characters and Opponents.....	5
7. Levels.....	5
8. User Interface.....	5
9. Content Additions.....	6
10. Feedback for the Player.....	6
11. Deployment.....	6
12. Development.....	7

1. Revision History

– only needed if you are working on multiple versions of the document or making major changes to the game. Probably not needed for most games in this class. However if I review a version that then significantly changes, include the history.

2. Game Overview

Turf Wars is a fast paced, team based, player versus player experience that combines elements of a traditional “tug-of-war” game with Minecraft’s building and combat mechanics. Gold Team and Black Team face off in an arena where the ground they stand on represents their score in the game. The objective is simple: expand your team’s territory by eliminating enemy opponents.

Kennesaw State University Esports needed a fresh and engaging Minecraft gamemode for campus events. This game will satisfy all of these requirements helping drive more players to the KSU Minecraft server. This gamemode was purposefully designed with simplicity in mind so experienced players and new players will be able to enjoy the game together.

3. Gameplay

3.1 Objectives

The objective of the game is to claim the enemy territory and eventually claim the entire map. This is done by eliminating players on the enemy team.

3.2 Game Progression and Play Flow

The player will start in a build phase where they can build structures to act as cover. After the build phase ends there will be a capture phase where the teams will now eliminate players of the opposite team in order to claim a section of the arena. This phase will continue until a team is able to eliminate enough players that allows for the capture of the whole arena.

3.3 Mission Challenge or Puzzle Structure

there will be two phases; a build and capture phase. Once the build phase is complete the capture phase will start and will end when all the territory is captured or the timer runs out.

4. Mechanics

4.1 Turf Game Mode Mechanics

- **“Turf” Tracking:** The system must track the divide line between the two teams.
- **Kill-Based Advancement:** When a player kills an opponent, the divide line will move X blocks into the enemy territory.

- **Build Restrictions:** Players must be restricted to placing blocks only in their teams “turf,” and not being able to place blocks on the enemy team’s “turf.”
- **Win Condition:** The game ends when one of the team’s territory reaches 100% of the arena area, or whichever team has the most area controlled when the timer runs out.

4.2 Combat Mechanics

- **One-Shot Kills:** The system should register any arrow hit as an instant kill and begin respawning the player.
- **Respawn Mechanics:** The respawn mechanics will include a dynamic timer that is less for smaller games (2v2, 4v4, etc.) and greater for larger games (8v8, 16v16, etc.).
- **Materials and Ammo:** Each player should spawn in with 16 blocks and 16 arrows. During a predetermined interval of time (e.g. every 15 seconds) each player will receive an additional block and arrow.
- **Arrow Retrieval:** The system should automatically clear arrows from the ground to prevent lag.

4.3 Team/Spawning Mechanics

- **Auto-Balancing:** The system must distribute players into two teams Gold Team and Black Team ensuring even numbers at the start of the match.
- **Friend Queuing:** The system should prioritize two or more players queuing together being placed on the same team.
- **Spawn Protection:** Players should be invincible for 2-3 seconds after respawning to prevent “spawn camping.”

5. Game World

The Game World design really just consists of the waiting lobby and the gameplay map itself. The waiting lobby will be a medium sized space where players can move around freely, given the boundaries provided. In addition there will be some small obstacles/interactive objects that a player can utilize while they pass time. For user insight, information like player lobby size, time elapsed, and game mode selection will be displayed.

The gameplay map is designed to have two spawn locations for the opposing teams. This will be a room closed off from all directions facing away from the opposing team. Once the allotted time for game start has depleted, players will be able to move into the main area of the map. The main area resembles a relatively large rectangular field that connects the two spawn rooms. This field is where all gameplay takes place. The flooring will visibly be split in half by the two team colors, this will also be the visual cue for which team has the upper-hand. Other than about one or two climbable objects on each team's side, this area will mostly be empty. This allows for the players to have space to build blocks for coverage when given time to do so. This gameplay map is reset and used again after each playable round.

6. Characters and Opponents

- **Player Avatar:** Players will be able to use the skins/accessories owned by their microsoft account.
- **Characters in the Game:** Fully user based game, meaning the only characters in the game are the users on each opposing team.
- **Opponents:** The opposing team of users.

7. Levels

- One playable level/map is provided.
- Setup in a two team, Teamdeath match style of play
- All game mechanics will be provided and understood before game start
- Player progress is assessed as a team, as they get eliminations on the opposing team they will gain a color row corresponding to team color.

8. User Interface

- What is the visual system? Specifically,
 - If there is a HUD, what is on it, how is it displayed and how is it accessed?
 - What menus are included and how are they navigated?
 - What is the camera model?

- Control System – How does the game player control the game?
- What are the specific commands?
- Audio, music, sound effects Game Art – intended style
- Help System Settings – what can the user change?

9. Content Additions

– How can the administrator add new content to the game? The information on the Functional Specification and Software Design Document pages may be useful in thinking about this non-game part of our project.

- Content additions can include things like tweaks to gameplay, ex: arrows are given on 5 sec intervals vs 15.
- Content additions can also include adjusting timers, kill rewards, respawn timers, etc.
- The additions can be done in the source code and rebuild for the gamemode.

10. Feedback for the Player

How can the player tell if they are doing well? Is there mechanisms to tell them what they are doing wrong?

- When a player successfully hits another player with an arrow a ding sound effect will be played client side.
- When holding tab players will be able to track their kills, and thus the amount of turf they've acquired for their team. Deaths will also be tracked.
- In addition to seeing their turf grow, a counter will display how much area the player's team controls.

11. Deployment

- The gamemode is playable on minecraft on the official KSU minecraft server.
- Minimum requirements for running Minecraft java edition according to minecraft.net are: 64-bit Windows 10/11 OS, Intel Core i3-3210/AMD A8-7600 CPU, 4GB RAM, and Intel HD Graphics 4000 or NVIDIA GeForce 400 series GPU.

- Since this is online multiplayer, an ethernet connection is preferred. If this is not possible, strong Wi-Fi will work as well.
- To get started players must connect to the server and navigate the player to the portal labeled Turf Wars, there they will be taken to the lobby to wait for more players to start.
- After the lobby phase is over (sufficient players join, or host starts), all players in the lobby will be assigned to teams (chosen or random) and teleported into the map where the game will begin.

12. Development

– What does a developer need to know in order to continue development on the game? The information on the Software Design Document pages may be useful in thinking about what needs to be captured.

- Minecraft/Paper API 1.21.11
- The code for our project will be in our team's public github repository.
- WorldEdit API