**Wroughtten**

**GDD**

*Video Demo: https://youtu.be/B5c-pfe-Ro4*

By Oisín Ivory

**Contents**

**Content…………………………………………………………………………………………………………… Page**

1. Introduction……………………………………………………………………………………………………… 3
2. Specification……………………………………………………………………………………………………… 4
   1. Concept………………………………………………………………………………………………. 4
   2. Story…………………………………………………………………………………………………… 4
   3. Game Structure………………………………………………………………………………… 4
   4. Objective……………………………………………………………………………………………. 5
   5. Graphics……………………………………………………………………………………………… 5

2.5.1 Technical Overview………………………………………………………………. 5

2.5.2 Visual Theme………………………………………………………………………… 5

2.5.3 Lighting………………………………………………………………………………… 5

1. Gameplay…………………………………………………………………………………………………………. 5

3.1 Resources…………………………………………………………………………………………… 5

3.1.1 Health……………………………………………………………………………………. 5

3.1.2 Stamina…………………………………………………………………………………. 6

3.2 Movement…………………………………………………………………………………………… 6

3.3 Item Interaction…………………………………………………………………………………… 6

3.4 Hands Mechanic………………………………………………………………………………….. 6

3.5 Weapon Manipulation………………………………………………………………………… 6

3.5.1 Fire Selector………………………………………………………………………….... 6

3.5.2 Bolt Manipulation……………………………………………………………………. 6

3.5.2.1 Bolt Action………………………………………………………………… 7

3.5.2.2 Closed Bolt………………………………………………………………… 7

3.5.2.3 Open Bolt…………………………………………………………………. 7 3.5.3 Reloading & Reloading Methods…………………………………………….. 7

3.5.3.1 Magazines………………………………………………………………… 8

3.5.3.2 Stripper Clips…………………………………………………………… 8

3.5.3.3 Manual Loading……………………………………………………….. 8

3.6 Ammo and Loaders…………………………………………………………………………….. 8

3.7 Inventory…………………………………………………………………………………………….. 8

3.7.2 Chest Rig………………………………………………………………………………. 8

3.8 Enemies………………………………………………………………………………………………. 9

3.8.1 Raiders…………………………………………………………………………………. 9

3.8.2 Paramilitary………………………………………………………………………….. 10

3.8.3 Military…………………………………………………………………………………. 10

1. Artificial Intelligence.…………………………………………………………………………………………. 10
2. Controls…………………..…………………………………………………………………………………………. 11
3. Damage Model……………………………………………………………………………………………………. 11
4. Progression………………………………………………………………………………………………………….. 12
5. Technical……………………………………………………………………………………………………………… 12
6. Project Management………………………………………………………………………………………….. 12

9.1 Project Methodology……………………………………………………………………………….. 12

9.2 Schedule …………………………………………………………………………………………………. 12

9.3 Version Control………………………………………………………………………………………… 12

9.4 Scrum Reports…………………………………………………………………………………………. 12

SCRUM 1…………………………………………………………………………………………… 12

SCRUM 2…………………………………………………………………………………………… 13

SCRUM 3…………………………………………………………………………………………… 13

SCRUM 4…………………………………………………………………………………………… 13

SCRUM 5…………………………………………………………………………………………… 13

SCRUM 6…………………………………………………………………………………………… 14

SCRUM 7…………………………………………………………………………………………… 14

**1. Introduction**

This document is the design for the game Wroughtten.

Wroughtten’s Gameplay

At the height of the cold war, the largest threat comes from within.

Wroughtten is a first-person level-based shooter game.

Players play as a Soviet secret service agent (KGB) and are tasked with stopping an illegal paramilitary organisation that are on the verge of creating nuclear weapons to use against the Soviet Union. Players are tasked with infiltrating the enemies base and stopping the development by any means necessary.

Players are given a weapon at the start; however, their ammo isn’t infinite and must scavenge what they can from the corpses of their enemies to continue. This can be various weapons and ammo.

During levels player can scavenge weapons and ammo from fallen foes. This game is set in the 1970s meaning there will be many different weapons and weapon systems within the game. Players must learn how to manipulate different types of weapons in order to use them effectively.

The main feature that sets this game apart is the unique weapon and inventory systems, by which the player must manually interact with the weapons in the game to use them. Players also make use of a chest rig, where they can store quick access items, such as ammo and magazines.

Wroughtten draws inspiration from games such as;

S.T.A.L.K.E.R.: Shadow of Chernobyl – (GSC Game World)

Receiver – (Wolfire Games)

Escape from Tarkov – (Battlestate Games)

Hotdogs, Horseshoes & Handgrenades – (Rust.Ltd)

Hrot – (Spytihněv)

**2. Specification**

2.1 Concept

The aim of Wroughtten is for players to have a medium-paced shooting game, with unique weapon and inventory systems. Players should be kept on the edge of their seats at the risk of dying, but also looking forward to seeing what new enemies and weapons await them in the next level.

2.2 Story

Wroughtten takes play in Russia during 1970, where the Cold War is escalating. The player is a KGB agent, tasked with disposing of an illegal paramilitary which are attempting to create nuclear weaponry.

This setting is chosen as it allows the use of a variety of different weapon systems, such as bolt action, semi-automatic and automatic weaponry.

2.3 Game Structure

The game will be split into 3 different parts, each having their own visual theme. The first area will be based in urban Russia, where the player is tasked with finding documents that will lead them to the paramilitaries base before finally making it to the final factory where the biggest threats lie. Throughout levels players can search for special weapons lying around which would be tucked away somewhere. This is to incentivise replaying missions to find them.

The first area will act as an introduction to the game, the enemies will not be heavily armed and will consist of lesser trained thugs. This should give the player a better chance of survival while they learn the mechanics. To complete the mission the player must ascend a housing complex to reach the apartment where the documents are, they will be met with resistance from the thugs trying to protect the documents.

Once the player acquires the documents they will then go onto the next level. This will take place on the outskirts of the main base. This will be a more rural area up until the player makes it to the entrance of the base. This act will contain better equipped and tougher enemies. The base will be a military like complex with housing for soldiers. The player must fight their way through and reach the factory.

The factory will be the final level in the game, where players are against the toughest enemies. They must fight their way through a dilapidated factory which is semi-refurbished by the paramilitary. Throughout the level the player should be able to find American weapons, suggesting that they are funded by the American government. The game then ends with the player reaching the development area.

2.4 Objective

The main objective of this game is to survive levels and reach the end. All the levels will have a relatively linear layout, and deviations will most likely consist of secret areas where the players can find some extra and special items which can help them trough the level. Players will also have to scavenge from enemies in order to keep ammo supplies and have weapons.

2.5 Graphics

*2.5.1 Technical Overview*

The aesthetics for Wroughtten will be an attempt at re-creating older style graphics such as those from the PlayStation 1 / GameCube era. The polycount on models will be kept to a minimum, textures will be limited to 128x128 pixels maximum for objects, and the render resolution will be around 640×240. Character models will also be made in sections, the torso, arms, and legs will be separate meshes, this is to keep it consistent with older graphics and help make animations. The render resolution can be modified in settings for those that may not want it.

*2.5.2 Visual Theme*

The first area of the game will be a urban environment, where the player is tasked with finding the location of the visuals will all be based on overgrown, dilapidated, rural, soviet architecture. The soviet theme is going to be used as it allows for a variety of weapons, ranging from new to old. All assets will be made in house.

*2.5.3 Lighting*

Expeditions will be taken place during the evening. It should be dark, but there should still be a high level of visibility to allow the player to see enemies and loot.

The bar area will be underground, and will make use of candles and lights, giving the area a warm yellow/orange lighting to it.

**3. Gameplay**

3.1 Resources

The resources that are vital to the player are health and stamina.

*3.1.1 Health*

For the player they will be able to see their health on screen. In the event of being damage the players health will decrease. Enemies will also have health, however any damage delt to an enemy will be localized. In the event of an enemy being shot in the legs, they will move at half speed. In the event they get hit in the arms, they will be less effective with weapons, most likely a decrease in accuracy. Headshots will be an instant kill on any enemy. And shots to the chest will deal the more damage than hitting the limbs. Each part will have a health relay with an attached modifier, making it so that when those areas are hit, the damage will be multiplied before being deducted from the total health.

*3.1.2 Stamina*

The player must manage their stamina, the lower the stamina the greater their weapon will sway when holding. Stamina is drained when running and jumping. Stamina will regen when the player is not moving or if they are walking.

3.2 Movement

In Wroughtten, the player can, walk, run, crouch, and preform a small jump to help get over obstacles. Running can be used to traverse the map faster at the cost of stamina. If the player runs out of stamina, they can no longer run, only walk.

3.3 Item Interaction

For the player to use weapons and scavenge, they must be able to interact with items. Items can be picked up by looking at them and pressing one of the interact keys (One for the left hand and one for the right hand), this will take the item into the respective hand.

3.4 “Hands” Mechanic

The player has 2 hands, a left and a right. To carry out actions, such as picking up items, the player must have a hand free to do so. Holding a weapon out will always occupy the left hand, leaving the right hand to preform weapon manipulations. To fire a weapon, both hands must be free, as the player must hold the weapon and pull the trigger.

The player can also use their hands to reload weapons and magazines, by holding bullets in one hand and a magazine in the other. The hand movement will make use of transforms and linear interpretation, to make is easy to create movements. Examples of this is that if there is a bullet in one hand and a magazine in the other, then the bullet hand will be brought over to the magazine, loading a round.

3.5 Weapon Manipulation

Weapons are used by the player to attack enemies. The player can aim the weapons by holding the aim button, and fire them using the fire button. Players don’t have to aim to fire, however they are less accurate when they don’t.

*3.5.1 Fire selection*

Weapons in Wroughtten will have fire selections on different weapons, allowing certain weapons to either be in safe where the weapon will not fire, semi-automatic where the weapon will fire 1 bullet with every trigger pull, or automatic where the weapon will continuously fire with a trigger pull. Bolt action rifles are an exception, they will only have a safety.

*3.5.2 Bolt Manipulation*

One of the main features of Wroughtten will be the weapon manipulation. Different guns will operate in different ways and will require the player to manually interact with them. This will be done by having a “bolt manipulation mode”, the player will enter this mode by holding the bolt manipulation mode button, then dragging the mouse to mimic the action. The three main features will be bolt action, close bolt, and open bolt weapons.

*3.5.2.1 Bolt Action*

To operate bolt action weapons, the player must open the bolt, this can be done by ensuring that the safety is off, then entering bolt manipulation mode. Once in this mode the players mouse input will be used to move the bolt.

Bolt actions bolts will make use of 2 stages, the first rotation and the draw back. During the first rotation the mouses x input will be added to the bolts stage progress, the stage progress will be a value of 0 to 1. This will be used as the value for the linear interpretation from the resting position to the rotated position. A speed modifier will be used to control the speed of the bolt.

**stageOneProgress += (Input.GetAxis("Mouse X")\*-1) \* Time.deltaTime \* boltSpeedModifier;**

Once stage one is complete, stage two will take the mouses Y input and use it to draw the bolt back, the mouse Y input will be multiplied by -1, so that it will be positive while drawing back. It will be the same as the first stage, except the transform it is moving to will be at the back. Once the bolt is back, it will eject a cartridge if there was one in the chamber, on the way back it will then catch the next round and chamber it.

To help with controlling ejecting and loading the bolt can return the percentage of it’s course by adding both stage progresses and dividing by 2.

*3.5.2.2 Close Bolt*

To operate closed bolt weapons, it is much easier and more typical. It is as simple as ensuring that the safety is off, and that there is a round in the chamber. If there is not a round in the chamber, the player can enter bolt manipulation mode, and just pull the bolt back, on release of the mode, the bolt should spring back, closing the bolt. It will also make use of linear interpretation to manage the bolt. Some weapons may also lock back the bolt once it has emptied its magazine, the bolt can be release by pulling it back.

*3.5.2.3 Open Bolt*

These are the opposite of closed bolt weapons. These weapons will fire when the bolt is open. To fire and open bolt weapon, the player must ensure that the weapon isn’t on safe, then pull back the bolt. Once back it will stay open and will only close once it is fired without any ammo in the magazine. To begin firing again the player must insert a new magazine and pull back the bolt. It will also make use of linear interpretation to manage the bolt.

*3.5.3 Reloading & Reloading Methods*

Reloading can be done by holding the ammunition in the right hand, then pressing the reload key. The weapons in Wroughtten will make use of collisions to manage reloading. Different weapon systems will interact differently when it comes to loading. There will also be different loading methods, such as magazines, stripper clips, and manual round loading.

The reloading system will make use of reload points on weapons, where the hand holding the ammo will interpolate towards to load the weapon, once a collision is made it will load the magazine.

Magazines can be ejected by hitting the reload key while there is a magazine in the weapon, this acts as the mag release.

*3.5.3.1 Box Magazines*

Box magazines will be simple to reload, simply by pressing the reload key while holding the magazine in the right hand and having the weapon out. The magazine should then travel to the reload point and then attach.

*3.5.3.2 Stripper Clips*

To reload using a stripper clip the player must ensure that the bolt is locked back, once the bolt is back, they must then hold the reload key. Once the stripper clip attaches to the bolt, then then must move the mouse down to add the rounds into the chamber, when the reload key is released the stripper clip is removed, then can then move the bolt back and chamber a round. Weapons that use stripper clips will have an internal magazine that the rounds will feed into.

*3.5.3.3 Manual Loading*

To load manually, the player must pull the bolt back, and while holding bullets in their right hand, press R to load them into the magazine. Once the rounds around loaded, the player can then close the bolt which will chamber the round.

3.6 Ammo & Loaders

The player can purchase boxes of ammo from the barkeep, they will contain many rounds of ammunition. The boxes of ammo can be unpacked allowing the player to pick up rounds. The player can hold multiple rounds of ammo in their hand making it easier to load. Ammo can be used to fill magazines and stripper clips, or manually load it into weapons, simply by holding the ammo container in one hand and ammo in the other, then press the reload key. If the key is head the ammo is held at the magazine until the magazine is full, then it will stop feeding it rounds.

The player also has access to different ammo types sold by the barman namely being hollow point and armour piercing. Hollow point is more effective against unarmoured targets, whereas it’s less effective against armoured targets. Armour piercing costs more and will do slightly less damage to unarmoured targets, however it is very effective against all targets.

3.7 Chest Rig

*3.7 Chest Rig*

The player will also get a chest rig, used to store quick access items, such as ammunition. Players can take objects out and put objects into their chest rig by using the number keys.

3.8 Enemies

There will be 3 different factions in Wroughtten, there are raiders, paramilitary, and military, each one is used in the different stages of the game. The factions symbolize the difficulty of the enemies. Enemy’s difficulties are based of the weapons that they use, what armour that they have and their efficiency with weapons.

*3.8.1 Thugs*

Thugs are the first type of enemies the player will face in the first area. They are mostly just regular criminals. They are often seen wearing civilian clothing and using low powered weapons.

*Raider*

These are the weakest variation of raiders; they are equipped with pistols and have no armour. They’re very inaccurate and their weapons are very weak against armour, however with enough hits they can easily kill the player.

*Bandit*

This is the mid-tier enemies that are also equipped with bolt action rifles; They are much more proficient and are just about more accurate than the raiders. They have homemade armour that covers their chest, which helps against low powered weapons.

*Marauder*

Marauders are the top tier raiders, they are equipped with bolt action rifles, which fire slowly but can do lots of damage. They’re more accurate than bandits and make use of weak armour.

*3.8.2 Paramilitary*

Paramilitary groups are survivors that have trained similarly to military groups. They have low tier equipment but would be more efficient with it. They often wear dark green and mismatched camo and other military equipment.

*Trapper*

These are the lowest tier of paramilitary. They make use of pistols and are fairly accurate with them. They make use of weak armour to help protect their torso.

*Hunter*

Hunters are the mid-tier enemies for the paramilitary. They make use of bolt action rifles and are quite accurate with them at range. These soldiers are often seen with old semi-automatic rifles, they make use of weak armour to help shield themselves.

*Predator*

These are the highest tier of paramilitary soldiers. They can be distinguished from others by their white gas mask. They’re equipped with old submachine guns, which they’re not accurate with, however they can shoot very fast. They wear decent armour that has been scavenged from other areas.

*3.8.2 Military*

The military is that best equipped faction. They patrol areas in search of raiders but are known to attack any scavengers that they see. They’re seen wearing full military gear and often have strong weapons that they’re familiar with.

*Private*

Privates are the weakest of the military faction; they make use of semi-automatic rifles. They’re accurate with their weapons and have good armour.

*Sergeant*

Sergeants are the mid-tier of the military faction. They use automatic rifles which they’re quite accurate with. They wear full military gear and use good armour, like the predators, they too wear white gas masks.

*Captain*

Captains are the highest tier enemy. They use suppressed automatic rifles and are very accurate with them. They’re seen wearing full combat gear and are often seen wearing large helmets. They have very heavy armour which renders hollow point ammo useless.

4 Artificial Intelligence

The enemies will perform different actions to one another. When the player spawns, enemies will not know of their existence, so they will carry out patrols, searching through rubble and resting. Each enemy will have a vision cone, enemies can see the player when they’re inside of their vision cone. Their last seen position will be recorded.

Enemies will make use of a cover system. Cover points will be placed around the map, these will act as anchors that the ai can path find to, they consist of two points, cover and vision. When an enemy is alerted, they will cast a sphere to search for cover points. If there are no cover points located, the enemy will stand and shoot at the player, however if cover points are located, the cover point will raycast towards the player, then the vision point will raycast towards the player. If the vision raycast succeeds and the cover one fails, then that is considered the best case, and that point is selected. The enemy will then path towards that point and will stop searching. Once a player is out of vision, the enemy will path towards their last know location, once the player is spotted again the process will repeat.

When the enemy shoots at the player, their weapon skill and accuracy will be factored in. Enemies’ skill will be determined by their faction and tier within the faction. The skill of an enemy will be how accurate they are to a certain distance. When an enemy fires at the player it will sphere cast at a point on the path from the barrel to the player based on the enemy’s skill. The radius of the sphere will be the weapon accuracy. The enemy will then shoot at the point on the sphere to attempt to hit the player. The code to get that point is as follows;

Vector3 targetHitPoint = barrellPointTowardPlayer + (vector3.forward \* enemyRangeSkill) + (Random.insideUnitSphere\*weaponAccuracy)

5 Controls

W – Move Forward

A – Move Left

S – Move Backwards

D – Move Right

I – Inventory Screen

R – Reload / Repack hold for stripper clips

Alt – Bolt Manipulation Mode

Mouse 4 – Cycle Fire Modes

Q – Pick up Left hand

E – Pick up Right hand / Stow

H – Holster Weapon

Ctrl – Crouch

Mouse 1 – Fire / Move inventory items

Mouse 2 – Aim

6 Damage Model

Wroughtten will make use of localized damaged as explained earlier. All characters will have their health resource. Damage relays will be used to calculate and transfer damage from different areas of the body to the main health pool. Armour will have its own hitbox and absorb rounds stopping them from doubling damage. Each damage relay will have a damage multiplier assigned to it. The damage relay will take in the ammo type that was used and adjust, such as reducing the effectiveness of the armour value, or doing less damage depending on the ammo type.

7 Progression

The progression system will work is that players will start off with a handgun and some ammunition. At the beginning they must reload the handgun and chamber a round, this will give the player a chance to get used to the weapon manipulation. Once they are comfortable with the weapon. As the player progresses with the mission, they will eventually run out of ammo. They will then have to start scavenging rounds from enemies, this is where the player has the chance to experiment with different weapons, an example in the first chapter is that they can choose to use low damage semi-automatic pistols, or swap to a bolt action.

8 Technical

*Target Hardware*

The target hardware for this game will be medium to high end PCs.

9 Project Management

9.1 Project Methodology

This project will make use of the scrum methodology. This is chosen as it allows for developers to choose a mechanic, plan out how they will go about creating it then spend time creating it before moving onto the next set of mechanics.

9.2 Schedule

See external file ‘plan.xls’.

9.2 Version Control

GitHub is going to be used for version control, it will also allow for working across multiple machines.

9.3 SCRUM Reports

**Scrum 1: Weapon Model**

Sprint 1

Created SKS weapon model in Maya.

Sprint 2

Exported UV maps and used those to texture it in photoshop.

Sprint 3

Imported into Unity.

**Scrum 2: Weapon Mechanics**

Sprint 1

Added Bolt movement.

Sprint 2

Created Magazines, Ammo.

Sprint 3

Joined bolt and magazine in a weapon script.

Sprint 4

Added stripper clips

**Scrum 3: Hand Mechanics**

Sprint 1

Added ‘simple’ hands, can now pick up certain objects.

Sprint 2

Added dynamic reloading system that adjusts based on items held.

Sprint 3

Added weapon manipulation on weapon only if it is held.

**Scrum 4: Chest Rig Functionality**

Sprint 1

Pick up objects and place them into a chest rig slot.

Sprint 2

Retrieve objects from chest rig.

**Scrum 5: Bolt Action Weapon Model**

Sprint 1

Created Mosin Nagant weapon model in Maya.

Sprint 2

Exported UV maps and used those to texture it in photoshop.

Sprint 3

Imported into Unity.

**Scrum 6: Bolt Action Weapon Mechanics**

Sprint 1

Added new type of bolt movement.

Sprint 2

Add new bolt to model and attach relevant scripts.

**Scrum 7: Health and firing**

Sprint 1

Added bullet ray casting from weapon when firing live rounds.

Sprint 2

Made bullets become useless once fired.

Sprint 3

Created a simple health system.

Sprint 4

Added targets to scene that the player can fire at.