

Iron Snout (Game Deconstruction)

By Avneet Dhaka

Game Name - Iron Snout

Developer Studio - SnoutUp

Publishing Studio - SnoutUp (on Steam and PC) and Ratalaika Games S.L. (on consoles)

Platforms -

PC (Windows, macOS, Linux via Steam / itch.io)

Mobile - Android, IOS

Consoles - PS4, PS Vita, Xbox One/Series, Nintendo Switch

Genre - 2D Action / Endless Survivor

Core Gameplay -

Fast and Reflex based combat where a pig fights waves of foxes and enemies using punch, kick, jump, dodge and weapon catch mechanics.

1. Core Gameplay Loop -

> 1.1 Breakdown the primary gameplay loop -

Start Game -> Enemies Spawn (Left/Right and Above) -> Player Reacts (Direction Based Input) -> Enemy Defeated -> Score Increases -> Score Increases -> Difficulty Increases (Fast Spawns / Weapons / Mixed Patterns) -> player alive? -

If Yes -

Loop Goes on

If No -

Game Over -> Show Score -> Restart

- Input

1. Left / Right Attacks
2. Up Attacks and Down Dodge
3. Combos based on All 4 - Up, Down, Left and Right

- Combat

1. If input matches enemy direction -> successful hit
2. If wrong -> player gets damage
3. Combos give variations in combats
4. Using down input helps in dodging that works against enemy's thrown objects if timed perfectly

- Rewards

1. Score Raise
2. Visual Feedback (Blood and Gore)
3. Combo Potential (based on key-combinations)

- Escalation Layer (Difficulty Curve)

1. Spawn Rate increased
2. Enemy Variety Increased
3. Weapon frequency Increased
4. Slight Enemy Load increased

> 1.2 Visual Representation of the core gameplay loop -

[Here is a flow chart i have made using figjam](#)

2. Game Mechanics and Systems -

> 2.1 Core Mechanics -

A. Directional Combat System -

- Player attacks left or right based on enemy direction
- Jump or aerial kick for airborne threats
- No free movement - character remains centred

Design intent - *Removes movement complexity & focuses on pure reaction speed & directional accuracy.*

B. Timing Based Counter System -

- Attacks must be timed correctly
- Early input results in 0 player attack and late input results in player damage.
- Certain enemies require defined directional combos to attack.

Design intent - *Makes the game easy to learn but hard to master.*

C. Weapon Interaction Mechanic -

- Enemies throw weapons (axes, rockets, etc)
- Players can deflect or reuse these weapons.
- When directional combos are used perfectly on time, the weapons can hit other enemies thrown by their own types

Design intent - *Adds variations and increases depth beyond basic punch/kick combat.*

D. Health & Failure System -

- The player has limited health.
- Taking damage reduces HP
- Game ends upon 0 HP

Design intent - *Creates Tension.*

E. Endless System -

- No Level Completion
- Difficulty curve increases based on player playtime
- Score defines playtime, playtime defines difficulty.

Design intent - *Adds Competition nature if made for multiplayer games, encourages replayability.*

> 2.2 Progression -

A. Skill Based Progression -

- Player improves by their own playtime
- No Character Levels
- No Stat Upgrades
- No achievable / unlockable abilities

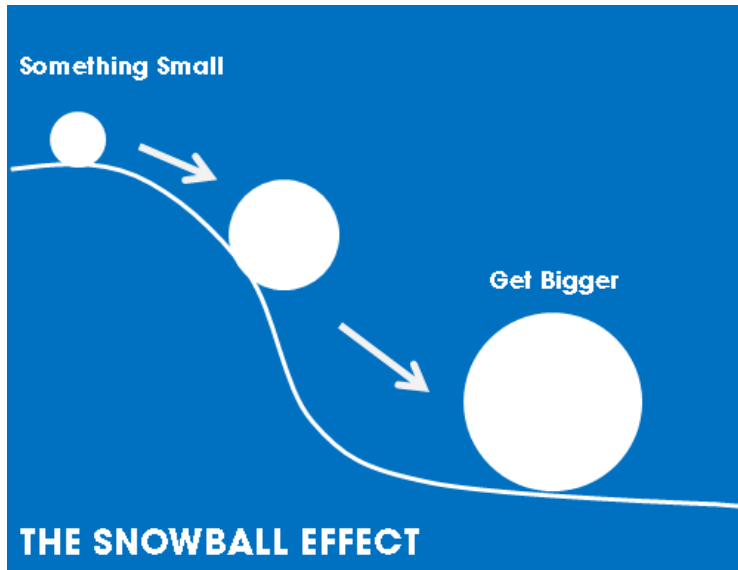
- Progression happens through -

- Multiple Play tests
- Pattern Recognition
- Improved Game/Direction-Sense

> 2.3 Balance Observations -

A. Difficulty Curve -

- The game is based on endless system (it is a hypercasual game)
- The balance curve is a “no win” difficulty curve that is designed for score-attack, fast paced 2D fighting games.



- The game follows more of a snowballing difficulty system (like snowball effect)
- Figure above explains how a snowball effect works

B. Weapon Balance -

- Some weapons are easier to deflect
- Weapons like rockets are hard to deflect
- Some weapons require dodging in specific directional combos.

> 2.4 Player Controller -

The player character remains centered, no horizontal movement allowed, combat occurs within a fixed combat space. No camera control, it is a static side-view camera with no zoom or panning, all threats remain within a readable field of view.

A. Keyboard and Mouse (PC)

- Arrow Keys - Used for directional attacks
- W - Jump and upward attack
- A - Left Attack
- S - Crouch
- D - Right Attack

B. Mobile Controls

- Swiping - Move Up/Down to dodge, jump or switch faces
- Tapping - Punch or Kick Enemies
- Projectiles - Tap/Swipe to catch or kick back items thrown by enemies

3. Level Design and Difficulty Curve -

> 3.1 How levels or challenges are structured -

A. Iron Snout doesn't have defined levels but have One Infinite Level -

- There is one single level which is endless
- The single level follows a simple Linear progression.
- The arena remains fixed throughout the session
- However, player can change the background / setting of the game level.

B. Challenge is structured as -

- Endless wave of enemy is spawned
- Intensity increases overtime
- Survival Based performance measurement

> 3.2 How difficulty increases -

A. Difficulty Scales dynamically during a session through -

- Increased spawn frequency
- Introduction of weapon throwing enemies
- Faster enemy movement.
- Complex enemy combinations.

> 3.3 Pacing -

A. Snowballing Flow / Tension Curve -

- In initial phase, the player stress is low
- Gradual acceleration
- Sudden spike moments when multiple threats overlap
- Eventually reaches a chaotic state.

> 3.4 Enemy / Obstacle Patterns -

A. Enemies Attack from -

- Left
- Right
- Above

B. Pattern Includes -

- Single-Direction threats.
- Weapon projectile + melee combination attacks
- Alternating left-right patterns.

C. Weapons add secondary obstacle layers -

- Rockets
- Axes
- Other throwable objects

4. My Personal Integration -

1. Player Vs Player

- Two players are matched in real-time where both are placed in a split screen arena in the initial phase, later changes to single screen once the game starts.
- The player who lasts till the very end wins one round
- The game progresses through 3 Levels
- Whosoever wins the first two levels, becomes the winner.

2. Ranked Based Gameplay

- MMR-based matchmaking (Matchmaking Rate)
- Game has different leagues like -> Iron, Silver, Gold, Plat, Diamond, etc
- Each league gives higher risk-to-reward ratio
- To climb a league, player need RR (Rank Rating)
- Players Fight for Rank RR
- Upon winning/loosing Rank Match, the player gets/looses RR

3. Tournament Mode

- Fixed Leagues, where player applies for the Mode
- Requires a higher league ranking and in-game desired currency
- Tournament is monthly based and each month, players can win some special rewards based on ranks achieved

Additional Things that can be done -

A. Customisable Avatars

- Profile Layouts
- Player Cards
- Player Character Skins
- Special-Edition Backgrounds

B. Stats Upgrades like -

- Each level up unlocks player abilities
- These abilities can have their own stats and can be upgraded based on time

C. Unique Characters based on abilities -

- There can be numerous characters
- Each character can have their own abilities
- These abilities define their work

D. Based on my experience with battle verse -

- Playing with Bots / Enemy AI felt too overwhelming
- No tutorial felt like a poor integration
- Practice Arena should have felt like -
 1. One place for practicing aim
 2. For playing against bots