Game Design Document

YAC

YET ANOTHER COMMANDO

Written by Nicolas Lebedenco

Version: 1.0

August 04, 2016

Conteúdo

[Changelog 4](#_Toc458277488)

[Concept 5](#_Toc458277489)

[Overview 5](#_Toc458277490)

[Premise 5](#_Toc458277491)

[Objective 5](#_Toc458277492)

[Structure 5](#_Toc458277493)

[Unique traits 5](#_Toc458277494)

[Platform 5](#_Toc458277495)

[Genre 5](#_Toc458277496)

[Rating 5](#_Toc458277497)

[Target audience 6](#_Toc458277498)

[Competitive analysis 6](#_Toc458277499)

[Financial analysis 6](#_Toc458277500)

[Technology 6](#_Toc458277501)

[Team 6](#_Toc458277502)

[Producers 6](#_Toc458277503)

[Developers 6](#_Toc458277504)

[Game Designers 6](#_Toc458277505)

[Contributors 7](#_Toc458277506)

[Visual Assets 7](#_Toc458277507)

[Audio Assets 7](#_Toc458277508)

[Features 7](#_Toc458277509)

[General 7](#_Toc458277510)

[Game play 7](#_Toc458277511)

[Elements 7](#_Toc458277512)

[Player movement 7](#_Toc458277513)

[Player actions 8](#_Toc458277514)

[Combat 9](#_Toc458277515)

[Weapons 9](#_Toc458277516)

[Power-Ups 9](#_Toc458277517)

[Camera 9](#_Toc458277518)

[Perspective 9](#_Toc458277519)

[Control 9](#_Toc458277520)

[User interface 9](#_Toc458277521)

[GUI 9](#_Toc458277522)

[HUD 9](#_Toc458277523)

[Multiplayer 9](#_Toc458277524)

[Game world 10](#_Toc458277525)

[Overview 10](#_Toc458277526)

[Travel 10](#_Toc458277527)

[Scale 10](#_Toc458277528)

[Time 10](#_Toc458277529)

[Scenarios 10](#_Toc458277530)

[Static Elements of the Scenario 10](#_Toc458277531)

[Dynamic Elements of the Scenario 10](#_Toc458277532)

[Player 10](#_Toc458277533)

[Enemies 10](#_Toc458277534)

[NPCs 10](#_Toc458277535)

[Level design 10](#_Toc458277536)

[Skills 10](#_Toc458277537)

[Obstacles 10](#_Toc458277538)

[Enemies 10](#_Toc458277539)

[Non-Enemies 10](#_Toc458277540)

[Level progression 10](#_Toc458277541)

[Game engine 10](#_Toc458277542)

[Render 10](#_Toc458277543)

[Textures 10](#_Toc458277544)

[Lights 10](#_Toc458277545)

[Shadows 10](#_Toc458277546)

[Audio 10](#_Toc458277547)

[Terrain 11](#_Toc458277548)

[Water 11](#_Toc458277549)

[Particles 11](#_Toc458277550)

[Physics 11](#_Toc458277551)

[Input controls 11](#_Toc458277552)

[Music and Sound FX 11](#_Toc458277553)

[Visual Style 11](#_Toc458277554)

[References 11](#_Toc458277555)

[Rambo: First Blood Part II / Ashura (阿修羅) / Secret Command - Sega Master System 11](#_Toc458277556)

[Assault Commando 11](#_Toc458277557)

[Assault Commando 2 11](#_Toc458277558)

[Minigore 2 11](#_Toc458277559)

[Appendix A: GUI Layout 11](#_Toc458277560)

[Appendix B: HUD Layout 11](#_Toc458277561)

[Appendix C: Character Animation 11](#_Toc458277562)

[Appendix D: Story 11](#_Toc458277563)

# Changelog

**Version 1.0**

* Initial concept, design and description of basic game mechanics and reference titles

# Concept

## Overview

Fast paced 3D vertical scroller shoot ‘em up resembling old NES games such as *Commando* and *Ikari Warriors*. With very simple controls suitable for mobile platforms, the game sets a pace for play, and the player must react quickly to the changing environment.

## Premise

A highly trained military asset is the last resort in a large scale conflict. In order to accomplish his mission, he must pass through enemy territory destroying as many opposing units as possible.

## Objective

The main objective of the game is to keep your character alive by destroying as many enemy units as possible until you find the exit point in each scenario. The exit point can be a door, a button/switch, a checkpoint or an element to be destroyed. Some levels may have secondary objectives such as to unlock passages, find secrets, escort prisoners, kill certain foes or free friendly units.

## Structure

Game is divided in chapters with each chapter later subdivided in levels. Each level can be played independently although they are expected to give the player a sense of progressive difficulty. No inventory system or complicated character customizations.

## Unique traits

A rule of thumb is any new game concept should have at least three unique selling points (USPs). These are elements that customers would see on the back of the box to entice them to buy the game (marketing bulletpoints).

## Platform

Mobile Devices: iOS 8+, Android 5.0+ (API21), Windows Phone 8.1

Minimum operating system versions are imposed by the game engine (Unity 5.x)

## Genre

Run and gun shooter: a shoot 'em up in which the protagonist fights on foot, sometimes with the ability to jump. Run and gun games may use side scrolling, vertical scrolling or isometric viewpoints and may feature multidirectional movement.

## Rating

Describe the ESRB rating you are aiming for with your game. Try to include the content descriptors that you expect to receive. Ratings and content descriptors can be found here: http://www.esrb.org/ratings/ratings\_guide.jsp

## Target audience

Every game needs to have a target customer in mind. The most basic demographic include age, gender. Ideally, you want to cover psychographics and positioning here.

## Competitive analysis

Describe the games that will likely attract the same target customers as the proposed game. It should include a list of relative strengths for each competitor and a side-by-side comparison across all competitors and the proposed game.

## Financial analysis

What is the argument for making the game? The components of the analysis are budget, projected sales, and return on investment.

## Technology

* Development:
  + Unity 5.x
  + Android SDK
  + iOS SDK
  + Windows Phone SDK
* Distribution:
  + Apple Store
  + Google Play
  + Windows Mobile Store

# Team

## Producers

Nicolas Lebedenco  
Thiago Borda

## Developers

Nicolas Lebedenco  
Thiago Borda

## Game Designers

Nicolas Lebedenco  
Thiago Borda

## Contributors

### Visual Assets

Add names of companies/individuals that contributed with assets used in the game

### Audio Assets

Add names of companies/individuals that contributed with assets used in the game

# Features

## General

1. Create Profile: a player must create a profile to store records of progress and personal game settings. Recommended number of profile slots: 6
2. List Missions with best score/stats: the player can list all game missions but only those missions previously accomplished and the immediate next unaccomplished mission can be played. List must show best score for accomplished missions.
3. Play Mission: a player can select a mission from the List of Missions and mission will be loaded to play.
4. Automatically save progress in the end of each accomplished mission. A mission **cannot** be saved in the middle.
5. View score/stats after a Mission is accomplished
6. Automatically save best score/stats for each accomplished mission in the player’s profile.
7. View/Modify Game Settings:
   1. Music [on|off]
   2. Sound FX [on|off]
   3. Language Selection [English|Spanish|French|Pt-br]
8. Abandon/Restart Mission
9. View Credits
10. Pause Game

## Game play

The basic game mechanic is that of a shoot ‘em up with vertical scrolling where the player can control direction of movement and aim independently.

The player moves towards the top of the screen while enemies come in the opposing direction (top to bottom).

Missions take place in static scenarios (previously generated) with an adjustable degree of random obstacles.

Power-ups help replenishing player stats and bring a variety of effects to weapons.

### Elements

1. **Mission** (aka Level): the association of a scenario with obstacles, including enemies, and one or more objectives. The game is divided in several missions that must completed in sequence by the player.
2. **Scenario**: the physical setting where the game action takes place. Scenarios are comprised of terrain, vegetation, water, static structures, weather, light (and shadow) sources, sound effects and music. Notice that because of the camera perspective there is no immediate use for skyboxes or fog.
3. **Static** **Structure**: a game object that is considered part of the scenario and does not move.
4. **Unit**: a game element that moves. Can be a Player, Enemy or NPC (non-player character)
5. **Player**: unit controlled by the user.
6. **Enemy**: units controlled by AI which pose an immediate threat to the player.
7. **NPC**: units controlled by AI that are neutral or allied to the player.
8. **Bullet**: general term for a projectile fired by a unit analogous to a real weapon’s bullet (but not necessarily holding the exact same attributes and constraints as in the real world).
9. **Power-up**: imaginary item that grants the player an advantage in game that may be either permanent or temporary. E.g.: increased movement speed, additional ammo, piercing shots, extended shoot range, etc.
10. **Destructible**: any game element that is subject to damage. This includes the Player, Enemies and less obviously some Static Structures.

### Player movement

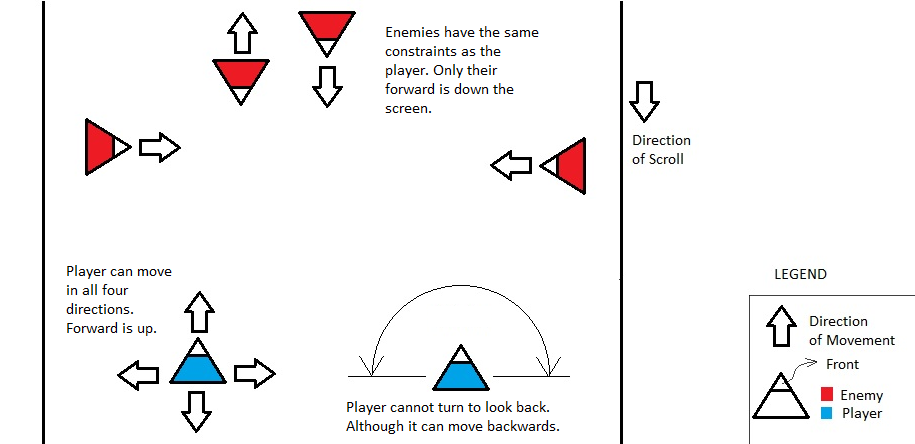
Player moves in constant speed. There is no distinction between walk and run.

Player can move in all four directions and aim at a different direction than that of movement. Controls for movement and aim are independent. Aim is constrained to the forward 180o arc.

Because of the scrolling mechanic (detailed in [reference section detailing scrolling]), movement backwards is limited and the player’s retreat is blocked by the bottom of the screen. The same does not apply to enemies which can run across the screen in any direction, most of the times top to bottom. Any other constraint such as maximum angle of aim, however, does apply for both types of units.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Up | Down | Left | Right |
| Aim | X |  | X | X |
| Move | X | X | X | X |

Table - Possible player movement and aim directions relative to the screen

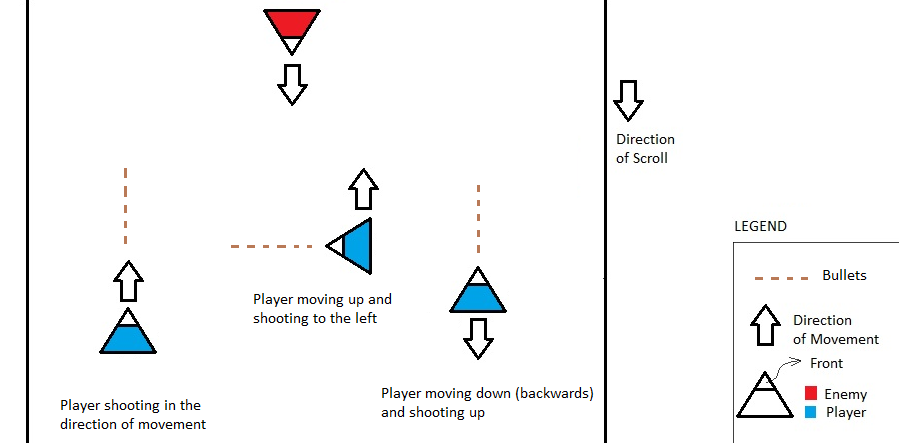


Picture - Player comes from the bottom of the screen; Enemies come from the top. Both Player and Enemies can aim at a maximum ±90o forward even though they can move in a different direction than that of aiming. Contrary to the Player, Enemies can move freely across the upper and lower bounds of the screen.

### Player actions

A Player can perform actions and move at the same time.

1. **Shoot:**
   * Player fires a bullet (in abstract terms unrelated to the actual game implementation which may or may not simulate a projectile in fact).
   * Because of the movement constraints a player **cannot** shoot backwards
   * Player can shoot with one of two weapons (see [add reference to weapons section]):
     1. Primary: activated and directed by the right touch pad [see add reference input controls]. Can be blocked by barricades [see add reference to static structures]
     2. Secondary: activated and directed by touching any point in the scenario. Do not have repeated fire so fire rate does not apply. Can shoot over barricades [see add reference to static structures]



Picture - Player shooting and moving

1. **Collect Power-Ups:** 
   * Player can collect power ups by walking over them.
   * No other unit can collect power-ups.

### Combat

1. **Health**
   * All destructible elements spawn with a certain amount of health points.
   * Damage taken deducts health points.
   * If the value reaches zero or below the game element is destroyed/killed.
   * Health points can be restored by:
     1. Collecting power ups
2. **Ammo:**
   * Only Secondary weapon uses ammo (Primary weapon has infinite ammo)
   * Every shoot consumes 1 unit if ammo
   * Ammo can be replenished by:
     1. Collecting power-ups
3. **Damage**
   * Bullet hits and explosions cause a certain amount of damage which is a value deducted from the health of affected destructible elements.
   * No friendly fire.
     1. It is naturally impossible for the player to get hit by his own bullets (no ricochet) but a player **can** take damage from his own explosives.
     2. A player can take damage from explosive structures.
     3. A player cannot damage or take damage from NPCs.
     4. Enemy units do not damage each other. As a consequence enemy bullets pass through other enemy units. [is this a good idea?]

### Weapons

Attributes and their possible values are:

1. Ammo: positive integer
2. Range: close | mid | far
3. Rate of Fire: low | high
4. Damage: low | med | high
5. Type of Damage: simple | piercing | splash

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Role | Ammo | Range | Fire Rate | Damage | Dmg Type |
| Machine Gun | Primary | - | Close | Low | Low | Simple |
| Flame Thrower | Primary | - | Mid | High | Low | Simple |
| Assault Rifle | Primary | - | Close | Low | Med | Piercing |
| Chain gun | Primary | - | Mid | High | Med | Piercing |
| Grenade | Secondary | 4 | Close | - | Low | Splash |
| Explosive Arrow | Secondary | 6 | Mid | - | Med | Splash |
| Rocket Launcher | Secondary | 4 | Far | - | Med | Splash |
| Air Strike | Secondary | 3 | Far | - | High | Splash |

### Power-Ups

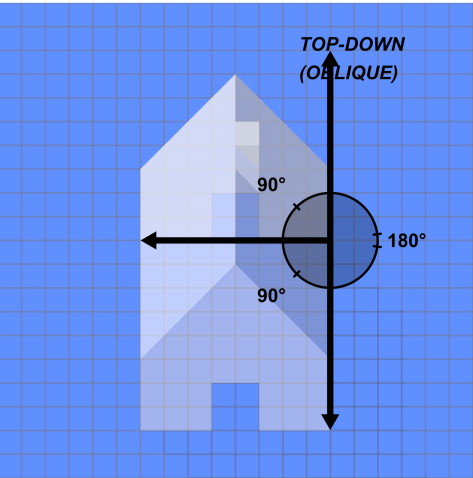
Power-ups are dropped by prisoners that the player helps to escape.

1. Restore Health
2. Restore Ammo
3. Weapon

## Camera

### Perspective

Top down camera with very small perspective distortion and a small angle relative to the ground normal (something around 20o) to resemble a 2.5D game using a ¾ (oblique) perspective.



Picture - Camera Perspective

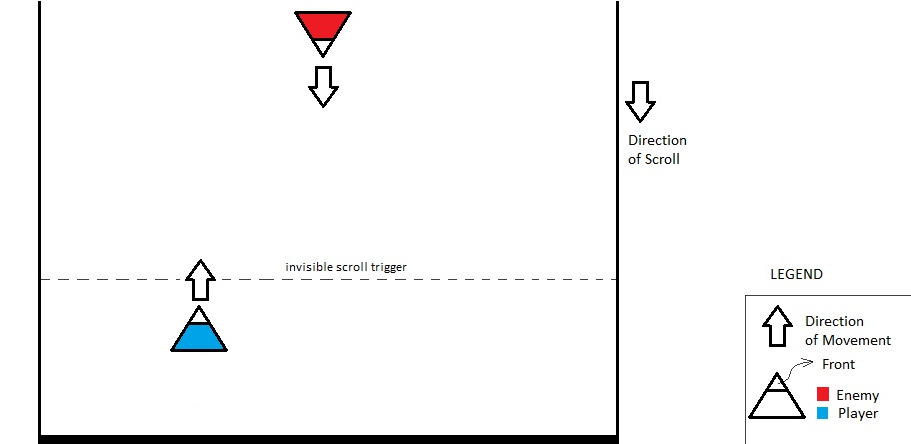
### Control

The scenario is as a static and whole piece for which the camera can only display a portion. [see add reference to scenario section]

The camera only moves up (relative to screen coordinates).

Imagine a line that cuts the viewport horizontally at a certain height contained in the lower half of the screen. This line accompanies the viewport. Let us call it an invisible scroll trigger.

Every time the player moves above and beyond this line, the camera must adjust its position by moving up as well (with a certain delay and speed) until the player gets back inside the delimited lower portion of the screen.



Picture - View port is defined by the thick black lines. In the picture, the invisble scroll trigger is shown and the player is about to cross it.

.

## User interface

### GUI

1. Splash Screen
2. Main Menu
   1. Start
   2. Options
   3. Credits
3. Create Profile
4. Select Profile
5. Confirm Delete Profile
6. Select Mission
7. Pause Menu
   1. Abandon
   2. Options
   3. Resume
8. Mission Outcome (score/stats)
9. Loading Screen

[Describe elements (types of widgets and there function)]

[Refer to section with layouts]

### HUD

1. Health
2. Ammo (Secondary Weapon)
3. Pause Button
4. Left/Right directional touch pads
5. Damage indication (flash red overlay)

[Describe elements]

[Refer to section with layouts]

## Multiplayer

Not supported.

# Level Design

## Overview

## Travel

Player moves from bottom to top of the screen and as a consequence the screen scrolls vertically to keep the player in scene all the time. Enemies come from the top of the screen and walk down, in opposition to the player.

Scenario, static structures and units are 3D but the game is essentially 2D since all units move in the same plane.

There is no collision between units, only with static structures.

## Scale

[to be defined]

## Time

Game works in real time.

Time stops when the game is paused.

## Scenarios

### Prison Camp

[describe scenario]

### Military Base

[describe scenario]

### Desert

[describe scenario]

### Island

[describe scenario]

### Swamp

[describe scenario]

### Research Facility

[describe scenario]

### Polar Station

[describe scenario]

## Static Elements

### Barricade

A barricade is a static obstacle that cannot be transposed by units which must walk around. It can offer cover by blocking primary weapons’ bullets. Beyond ordinary piles of rock and sand sacks, barricades may assume other forms such as fences, walls, recked vehicles, barrels, crates, etc…

### Rivers

A river is a static obstacle that can neither be transposed nor bypassed by units and **do not** offer fire cover. Rivers can only be crossed by taking a bridge. They may assume other forms such as that of cliffs and craters.

### Bridges

A bridge is a path that connects two sides of a river.

### Ponds

A pond is an area covered by shallow water so units can walk over it but they get slowed down in the process. Ponds may also assume the form of mud and ice.

### Trees

A tree is a static structure that can offer cover from bullets and explosions. Trees can also hide enemies that will run/jump towards the player from behind the trunks when it gets close enough. Tree crowns may also partially block the user’s top down vision of the game contributing to increase the level of difficulty of a mission.

### Bushes

A bush is a small chunk of vegetation that cannot be used as fire cover but can partially hide enemies will run/jump towards the player from behind the trunks when it gets close enough.

### Tall grass

When Tall grass covers an area, enemies will bend over and walk in disguise (showing only a small visual clue) to sneak closer to the player. They still need to stand in order to shoot in which case a larger portion of the unit becomes visible. Tall grass has no effect on the player (besides any possible visual implications)

### Huts

Huts are destructible structures that may contain prisoners. A player can destroy a hut only by using the secondary weapon. When a prisoner escapes, it drops a power up. Huts can assume other forms such as that of cages, cells, tents and trucks.

## Enemies

Enemies can be distinguished depending on:

* Whether they are stationary or can move
* Whether they can move and shoot at the same time
* How they move: only forward, only backward, only sideways or freely
* Whether they can aim at the player or will blindly shoot forward
* Range
* Rate of Fire
* Type of Damage (simple, piercing, splash)
* Whether they have a secondary weapon (i.e. fire bullets at close range and throw grenades at mid range)

Enemies live under the following constraints:

* NEVER shoot backward
* Try to avoid shooting each other (despite that there is no friendly fire damage in the game) to avoid awkward situations.
* Try neither to collide nor pass through each other, although there is no collision between units.

### Trooper

### Mortar

### Knife Thrower

### Bazooka Soldier

### Rifleman

### Sniper

### Flame-thrower

### Rocket Launcher

### Vehicle Patrol

### Tank

## NPCs

### Released prisoner

[describe]

## Skills

## Missions

# Game engine

## Input controls

Input control is in the form of two touch areas simulating directional pads on the lower corners of the screen. The left one is used to control the player’s movement and the right one is used to control aim direction and fire of the primary weapon both with a single gesture.

Secondary weapon can be activated by clicking directly over the desired target point.



Picture - Emulated directional pads in the touch screen. Left one controls movement, right one controls aim and fire.

## Textures

[what are the texture limitations imposed by the platform]

## Lighting

[what are the lighting limitations imposed by the platform?]

## Shadows

[what are the lighting limitations imposed by the platform?]

## Audio

**Do not** use 3D sound (spatial) sound sources.

**Do not** use Stereo tracks. Use Mono only.

Only import WAV files and let the Game Engine convert and compress the files in the build when needed to avoid re-compression.

Compressed audio (or modules) are best for long files like background music or dialog, while Native is better for short sound effects but having the incorrect compression settings can still mess up your memory usage or cause CPU spikes while playing. The iPhone, for example, can only decompress one audio file at a time using the hardware decoder so there might be CPU spikes if more than one compressed file is used. This can be solved by using the “Decompress On Load” setting. Performance increases, but it causes the memory footprint to skyrocket. So only use this when it’s really needed.

Rules of thumb for optimal performance when it comes to different types of audio:

* Short Clips – Native
* Longer (or looping) clips – compressed in memory
* Music – Stream from disc
* Files which consistently cause CPU spikes – Decompress on load

# Music and Sound FX

# Visual Style

# References

## Rambo: First Blood Part II / Ashura (阿修羅) / Secret Command - Sega Master System



## Assault Commando 1

## review-assault-commando-2.jpg (480×320)

## Assault Commando 2



## Minigore 2



# Appendix A: GUI Layout

# Appendix B: HUD Layout

# Appendix C: Character Animation

# Appendix D: Story