Galactic Enforcer Design Doc v2.0

CSU Chico, APCG, Fall 2012

9/13/2012

Game Overview

- Galactic Enforcer is primarily a puzzle game and secondarily a first-person shooter.
- It is designed with mobile devices (specifically Android and iPhone) in mind.
- Arcade style gameplay designed to compliment the pick-up-and-play nature of mobile games.
- Galactic Enforcer is built to be distributed in modules to allow for add-on content and digital distribution of new levels and patches.
- The primary purpose behind the development of Galactic Enforcer is to create an entertaining and impressive game that will reflect the skills and abilities possessed by APCG students.
- The secondary purpose is to develop a game that can be digitally distributed in an attempt to generate interest and raise funds for the department.

Proposed Feature Set Final Build

3D Characters/Environments	Movement
Arcade Style Gameplay	Multiple Areas
Various Objectives	Three Unique Alien Races
Three Weapon Types	Shooting Mini-Game
Downloadable Worlds	Add-On Weapons

Proposed Feature Set Beta Build

3D Characters/Environments	Movement
Arcade Style Gameplay	Single Area
Various Objectives	Three Unique Alien Races
Single Weapon Type	Shooting Mini-Game

Proposed Feature Set Alpha Build

3D Characters/Environments	Stationary Player
Proof of Concept	Single Area
Single Objective	Single Alien Race
Single Weapon Type	Prototype Shooting

The World of Galactic Enforcer

The Story

The game takes place on a Pax Ursae, a massive space station orbiting a dead planet in a distant galaxy. The various intelligent lifeforms who inhabit this galaxy have put their differences aside and formed a truce in the name of shared prosperity. This pact led to the creation of a shared nexus point that serves as a hub between the three planets. Nearly a century has passed since the construction of the this massive space station. Generations of aliens have been born, lived, and died on this massive floating metropolis. However, even in these times of peace, it is not surprising that corruption and backstabbing runs rampant. This influx of shady business has led certain individuals to pursue a career in the elimination of wrongdoers. Thus, this group of bounty hunters, referring to themselves as "Galactic Enforcers," has made a lucrative business out of punishing those who would do harm to others.

The Physical World

Overview:

Galactic Enforcer is broken into three content packs. All three packs are made up of different areas throughout the larger world within Pax Ursae. The world consists of three different districts: The Slums, The Business District, and the Noble District. For the purpose of the Alpha and Beta build of the game, we are only concerned with the Slums and the contained sub-districts.

Key Locations:

The Hideout: This small building is hidden from the public view. The Galactic Enforcers gather here to purchase weapons, obtain mission on targets, and to escape the watchful eye of the authorities.

The Slums: The most massive and populated of these areas. Within the slums, there are several different sub-districts that are controlled by warring gangs.

The Business District: This area contains more prolific buildings of grand stature. There are many office buildings, factories, and the space port.

The Noble District: Within this district there are a handful of mansions belonging to the wealthy as well as the capitol and the opera house.

Travel:

Pax Ursae is equipped with numerous teleporting facilities that make long distance travel simple and quick. One need only step into a teleporter and they can be whisked away to any of the districts throughout the space station. However, any other travel within the the districts themselves must be done on foot.

In gameplay terms, this allows the player to jump to different areas throughout the city quickly by

entering a teleporter and selecting a destination on the map. They will be transported to a corresponding teleporter in the district of choice and from there they must set out on foot to accomplish their mission. Once the mission is completed, the player can use a teleporter to return to the hideout and re-stock on weapons or receive new leads on potential bounties.

Scale:

The inhabitants of Pax Ursae range in size and shape, and as such, the buildings and passages must accommodate for the largest of alien races. Doorways and passages must be at least tall enough to allow for an 8 feet tall character to pass through. In comparison to the the player, who is assumed to be 6 feet tall, most areas will seem a bit too tall.

Each playable district spans several blocks but will contain multiple objectives and give the player multiple locations to explore and paths to navigate. This will lead to levels being designed around having one larger “frame level” in which certain assets will be re-arranged to restrict access to locations and mix-up the way the player interacts with and traverses the world.

The Characters

Alien Races:

Drestil: The most numerous of all races. They are prolific builders. Drestil are more of a jack-of-all trades race: They lack the technological brilliance of the Kiroli or the brute force of the Maa Taal but are good at taking advantage of a situation. On their own they developed limited technology that allowed them to fly to nearby planets after exhausting their own planet of resources. They analyze and improve upon designs and as such have recently made great technological strides. Their inherently mercenary nature precludes any racial prejudice; any Drestil is your best friend if you've got something it wants. In appearance, Drestil are of average stature with thin and lanky extremities. They come in numerous vibrant skin tones and tend to wear gaudy clothing and armor. They walk with a relaxed posture and never seem to be in a hurry.

Kiroli: An intelligent species, they quietly developed advanced technology on their small home planet until they set out to map the galaxy in search of other intelligent life. Their culture is made up of a complex caste system based on skin color and the patterns of their markings. Many Kiroli regard the other races as inferior but utilize the mercenary nature of the Drestil and the strength of the Maa Taal to accomplish their goals. Considerably smaller than the other two races, they have pastel colored skin adorned with colorful markings. The Kiroli have large expressive eyes and tall pointy ears. Their small size and short limbs force them to walk in a hasty bouncing gait.

Maa Taal: The natives of a resource-rich planet, the Maa Taal have found themselves dealing with traveling Drestil and Kiroli in recent centuries. A massive hulking race, their society is built upon physical strength, allowing for more social mobility within their culture than the Kiroli. Despite their relatively recent exposure to technology, the Maa Taal have adjusted quite well to life on Pax Ursae and can be found in all rungs of society. Their massive size and imposing appearance commands respect and allows them to intimidate the other races and achieve their personal goals. Maa Taal have tough leathery skin that is primarily blue or green. They have tall muscular bodies with exceedingly broad shoulders and always walk with purpose.

Key Characters:

Jackal: The main character of Galactic Enforcer. Jackal is silent and calculating but isn't given much of a personality. The player is never shown directly what Jackal looks like, but it can be inferred that Jackal is a mercenary Drestil.

The Handler: This helpful individual serves as Jackal's main point of contact on missions. The handler mainly deals in assisting Jackal by offering details and clues related to the current target. The handler is likely a resourceful Kiroli working alongside the Galactic Enforcers.

Art Direction

Color Palette:

Due to the age of the space station, many of the surfaces are faded and thus have a low saturation. In most locations, make heavy use of faded earthy tones or shades of gray and white. This can be offset by using bright neon lights to highlight areas and objects of importance. Using these color themes, the player will be able to follow distinct visual cues when exploring the world.

Characters should be light and colorful to help them stand out on small screens and give a strong contrast between the surrounding areas. Drestil should be have high saturation colors in the middle value range. Kiroli are small, but will be more visible to the player due to their light value ranged colors and bright clothing. Maa Taal are a bit larger, but will be slightly more difficult to spot due to their low saturation earthy-toned clothing. These color choices will ensure that all computer-controlled races are somewhat equally able to be spotted.

The large variety in hardware in android devices will cause us exercise a bit of restraint with textures. Incredibly complex and realistic textures will look grainy on smaller screens. Where possible, the textures for objects and characters should utilize flat colors as a way of combating the disparity in display resolutions. This does not mean the textures cannot be detailed, just that they will be less photo-realistic and stylistically be more similar to a comic book or cartoon.

World Design:

Assuming that Pax Ursae is a century old space station should inspire some idea of what the world looks like. Consider many aspects of the environment to be somewhat worn down and decrepit while other smaller objects or buildings may be newer additions or repairs. Some areas of the space station may be better maintained than others. The business district, for example, would likely have a janitorial staff and a repair crew while the slums are left to fall into a state of disrepair.

Most environments the player interacts with will be indoors. This will allow for modular pieces that can be re-arranged with minimal effort. Draw inspiration from the architecture used in airports or museums and make great use of multiple tiered chambers and various sizes of hallways. Bright neon signs or vibrant lights can be used to differentiate areas or buildings. In the case of slums, posters and

graffiti can help break apart an otherwise very repetitive environment.

Object Design:

The technologically driven science-fiction setting of Galactic Enforcer calls for distinctly high-tech objects and locations. However, this setting is assumed to have had this technology for quite some time. Consider most objects to be aged, dirty, or otherwise in a state of only partial functionality. This can be done through texture art or from the core design of the object itself. Pull inspiration from popular science-fiction settings to easily convey the purpose of objects to the player without the need for explanation.

Keep in mind we have a fairly limited polygon budget for Galactic Enforcer. Objects should be designed with the intention of getting the best results from as few surfaces as possible. Opt for flat edges and distinctly polygonal shapes rather than curves or spheres. If an object is to have depth and feel organic, it will likely need to be done with clever texturing rather than the inherent shape of the model.

Character Design:

All alien races should be bipedal for simplicity of rigging and animation. They will need to be fairly low poly in execution and should be designed with this goal in mind. Ideally, the proportions of the alien races will be somewhat exaggerated to allow them to be easily differentiated from one-another.

Characters will need clothing that is either tight (and can conveyed through texture art), thick and armor-like (so it can be handled with few polygons), or starchy (anything cloth will likely move in a very rigid way). The less polygons that are spent trying to create detailed clothing the more we will be able to allot to body joints for optimal rigging and animation expression. Despite these efforts, animations will likely be fairly simple and not especially expressive due to polygonal limitations. On the bright side, this will allow for more characters and objects to be displayed on screen and create more of a feeling of a living, breathing city.

Lighting and Effects:

Currently, the plan (subject to change) is to bake in lighting to the textures. This will allow for more consistent and realistic lighting than the Shiva engine will typically allow. Additionally, we will be able to achieve this look with less reliance on graphics processing. The downside to baking in textures is that it will require more space allocated to textures, as each part of the world will need its own texture. This is one of the reasons for using a frame-level that will be utilized for multiple missions in numerous ways; a model that will allow for maximum variety in gameplay without a large reliance on multiple levels.

Particle effects will need to use as few particles as possible while retaining a crisp and readable appearance. The setting of Galactic Enforcer calls for a number of glowing and bright effects. Lasers, rocket engines, neon signs, and teleporter glows and flashes rank among the more common visuals that

will be found in the world. Many of these effects will be vibrant and colorful which will make them stand out quite a bit from the more drab surrounding areas. It is these eye-catching visuals that will really pull the science-fiction setting together in a convincing way.

Sound:

Music can utilize some common instrumentation that is used in electronic music. There will be two main themes in Galactic Enforcer: A searching theme and a hunting theme. The searching theme should be something atmospheric and slow. It will need to be something that can loop well since it is possible the player will be searching for most of their play session. The hunting theme will play when the player is actively hunting the target, specifically when they enter aiming mode at the correct location. This piece will need to start quickly with very little build-up. The hunting theme music could play for only a few moments or potentially closer to a minute. As such, it is important that this piece is mid or fast paced, fairly long, and has the potential to be looped without being too noticeable.

Sound effects should be easily identifiable by the player. Draw inspiration from popular science-fiction films but personalize and tweak these sounds to reflect Galactic Enforcer's setting. Work in tandem with the Tech Art team to design sound effects that match visual effects.

The game will also need several pieces of alien dialogue that amounts to nothing more than gibberish in various tones and pitches. These sounds will play when you pass by conversing aliens and will serve to flesh out the world slightly and make it feel more authentic. Have fun with these voices and be sure to cover a wide range of emotions. We'll likely only need one voice actor for each alien race, but having more to choose from is always nice.

Game Design

Gameplay Features:

Camera Perspective: Galactic Enforcer is set in a 3D world in a first person perspective. The camera should be positioned assuming the player is six feet tall. This camera will navigate assuming it is the players first-person perspective.

Movement: The player can control Jackal through use of a virtual analog stick. The vertical axis will move the player in the corresponding direction: either forward in the direction that the player is facing or back away in the opposite direction. The horizontal axis will turn the player, causing their orientation to rotate in the direction that the stick is pressed.

Area Entrances: If the player enters a teleporter or walks through a specified entrance/exit, a map of selectable areas will be displayed. Tapping on one of the areas will take the player to the corresponding entrance to that area.

Mission Selection: When the player enters the hideout, they will be given a mission list that shows all

available objectives. A small description of the mission is given when the mission is tapped and the player must either accept or decline the mission. Once a mission is accepted, the player will be pushed back to the world map where they select a destination to begin searching for their target.

Shop: As a stretch goal, it would be beneficial to implement a shop in the hideout to give the player incentive to save up money to purchase weapons or equipment.

Interact: The player will be able to interact with objects by either passing through them or standing in a key location. This will allow for slightly more complex level design based around picking up keys that match corresponding doors or interacting with switches and objective points. Ideally, we would handle this aspect by tapping on the object itself, but a simple pickup/pass through system will suffice for early builds of the game.

Aiming: The player enters aiming mode by tapping the gun on the bottom portion of the screen. This causes aiming and shooting mode to be initiated. During aiming mode, the player is stationary. The player can tap the gun (bottom right) again to exit aiming mode.

Shooting: Shooting is done by tapping on a target when in aiming mode. The gun will begin to glow as it charges and a corresponding gauge will be shown in the HUD. The glow will cycle through green, yellow, and red and back to green. Tapping again will release the shot. At this point, the power of the shot should be compared to the distance between the player and the target. If the shot is within the appropriate bounds, the target will be vaporized. If the shot has been charged for too long, the target will explode and cause an alert. If the shot has not been charged enough, the shot misses the target and they are alerted to the location of the player. This gives the player a second chance to shoot the target. If the player once again misses the shot on the second attempt, the target flees and an alert phase is triggered. However, in this situation, no mission award will be granted. If the target is successfully vaporized, they will be fed another mission and seamlessly continue the game.

Escaping: If an alert phase is triggered, the player is given a limited amount of time to navigate towards the exit of the level. During this time, aggressive enemies will chase the player. This can be done with either spawning aggressive enemies or simply having aggressive enemies follow the player if there are enemies in the current level. Capture results in a mission failure.

Arcade Mode: As a stretch goal, we should consider adding an arcade mode. Similar to the alpha build of the game, the player will be positioned in a stationary location and be given a series of targets. A timer will begin counting down. Groups of enemies will walk across the screen and the player must identify and eliminate the target to proceed. Successful completion of a mark will award the player with points and a time extension. Shooting the wrong target will result in a time deduction. The goal is to eliminate as many targets as quickly as possible before the timer runs out. At the end, the player is awarded a score on a local scoreboard. This will give the player a quick, arcade-style game to play if they don't feel like completing missions.

Main Menu: This menu will need four main buttons. “Mission” will take the player to the game proper where they are given missions and track down marks. “Arcade” will take the player to a fast paced mini-game that can be played in short bursts. “Instructions” will take the player to a short document detailing the controls and goal of the game. “Exit Game” will close the current process of the game and return the player to their desktop. “Exit Game” opens a prompt asking if they really want to quit.

HUD:

The following buttons and utilities will be needed in the gameplay HUD:

One joystick, placed on the left center of the screen, to control the player movement and orientation.

One timer at the top left of the screen that can either count down or count up, depending on mission.

An indicator of current score on the top right of the screen.

Tapping on the bottom right of the screen will enter aiming mode.

A button in the bottom left to open up a mission description.

Mission Description screen will be closed by tapping the mission button a second time.

An abandon mission button on the bottom right of the mission description screen.

Tapping abandon mission must open a prompt asking if they really want to quit.

Upon tapping “Yes,” the game will return to the mission select screen.

Tapping “No” will close the prompt.

Level Design

Frame-Level: Each area will be designed around one large “Frame-Level” that is roughly the size of several city blocks. These levels will have multiple paths and areas that can be restricted for the player depending on the mission through the use of assets such as locked doors or road blocks. This one frame level will be used for a multitude of missions of various types.

We will be building many mazes that connect together into a shape that is ultimately circular. The circle is a basic tenant of FPS game design. It keeps the player moving. It opens up opportunities for exploration. It minimizes back tracking through familiar areas, connecting interesting places to a hub seamlessly. Every level designer will be building mazes, puzzles, and stealth sections that connect to one, if not several circles. This ties into us having one theme with one primary set of assets for all of the missions in one setting.

Mission Types:

Assassination: This is a hunting type mission. The player is given a dossier of information regarding the target's appearance, last known whereabouts, habits, and/or interests. Using this information, the player must navigate the area in search of their target, which they must eliminate with as few witnesses as possible. These missions may or may not involve sneaking depending on the rank and profile of the target.

Score Modifiers:

Time: Time limits can be added to any of these missions to increase variation or add difficulty. Otherwise, time will factor into score.

Collateral Damage: A minimum or maximum amount of collateral damage can be specified on a mission by mission basis. Data Retrieval and Assassination will typically have a maximum amount of collateral damage. Sabotage could possibly include either maximum or minimum collateral specifications.