I

Milestone 1 – GDD
Game Studio:
Game Title: Bags Please

Credits–
Sounder Designer: Sergey Maydanyuk
Composer: Sergey Maydanyuk
Game Designer: Datorien Anderson
2D Artist, UX (User experience): Philipp Brunsteiner

Table of Contents

Bags Please
GAME DESIGN DOCUMENT

Overview

Bags Please–is a casual top-down game about a smuggler who wants to smuggle contraband at an airport. You must figure out how to sneak illegal stuff through the airport security, scanners and successfully fly out from the airport.

This game design document is constantly updated and will have changes and highlights to reflect that. Feel free to add commits and such on the document, for more asynchronous communication!

Theme

The game's theme is: Inventory puzzle management and smuggling!

Platforms: PC
Game Engine: Unreal Engine
Input: Keyboard and Mouse
Gameplay References:

- Papers, Please
- This Is The Police

Scope: 3 - 6 Months
Split %: Revenue-Sharing

Main Restriction:
The game must be accessible by a casual audience and the fun factor will have to be thoroughly tested to ensure it's fun, and has humor. However, the game should not feel too happy at all.

Links:
High Level Overview
Audio-Visual Brief
Competitor Analysis
Marketing Plan

GRATIS

Main Goal

- Pay off your enormous debt.

Goals

The goals are:

- Pass the airport security and onboard the airplane.
- Optimize the packing of the suitcase.
- Avoid getting caught by security.
- Obtain new items to stash.

Rules

Win:

- Successfully flew out of the airport.

Lose:

- Caught by airport security.
- Loss of all contraband.

Actions

The player will be able to:

-

Transitions

How the game transitions from one state to another

- When the player progresses to the end of a level, the game will transition to another.

Items

Everything the player can interact with

- Suitcase
- Illegal Items
- Legal Items

Setup

The initial state of the game/level(s):

- For the initial play, the player starts on the screen where they must conceal the contraband; however, this is a set bag that they must deliver to the airport.

Game loop Time: 10 - 25 Mins

5 Screens:

- Route-Planning / Inventory Screen / 1st Packing Timer
- Duty-Free shop with the legal items
    - A minimum of two different shops to go through for stashing items.
    - Then, Packing Timer again(?)
- Airport Screen
- Scanner Screen / Inspection Process
- Screen with Airplane Departure

Feedback Loop(s) Ideas
Negative:

- Increased Security
- Increased Debt to Income
- Quota of items to smuggle increases. (but there is a maxQuota)

Positive:

- Scoring System
- Reward System

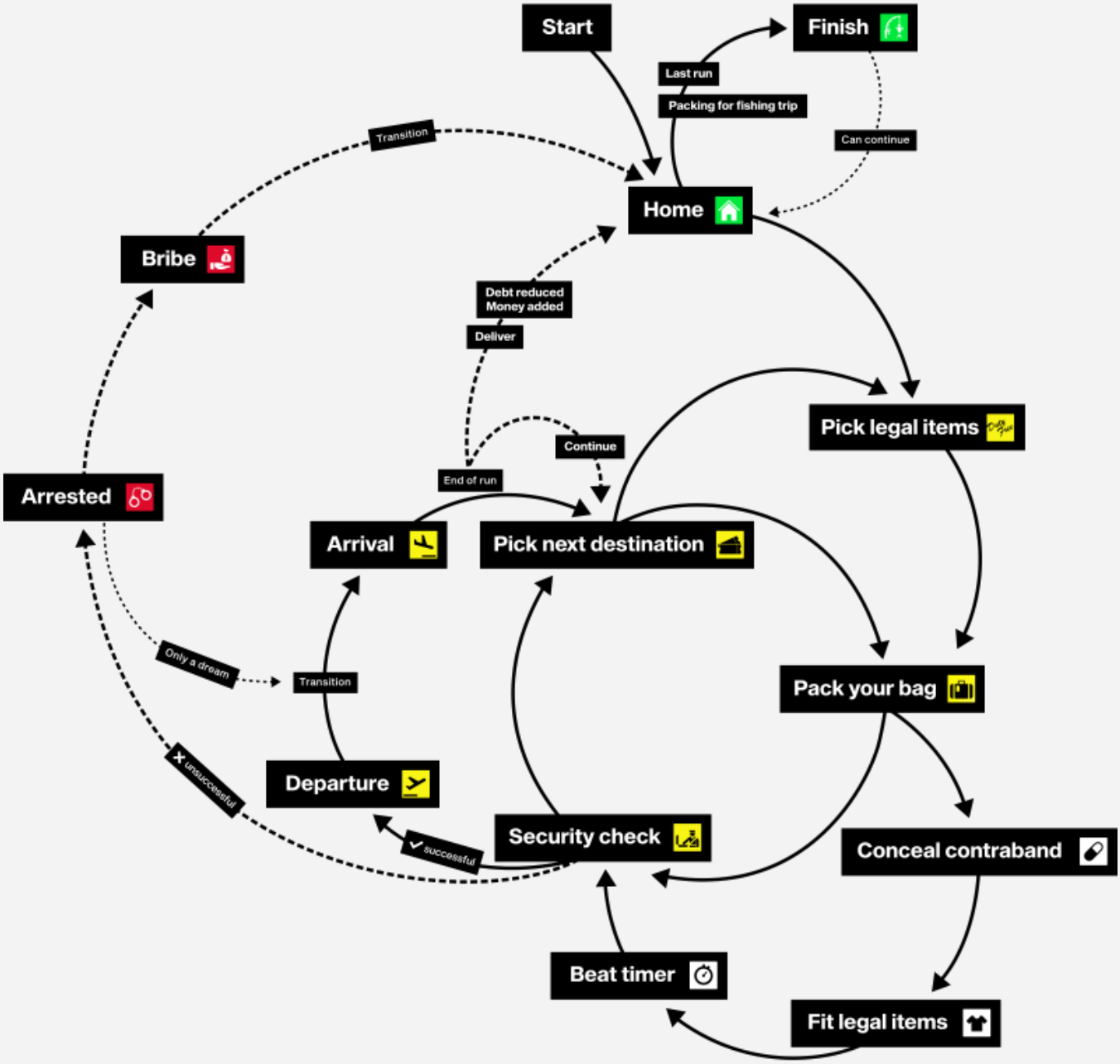
Game Antagonists:

- Time
- Airport Security
- Debt

Diagrams

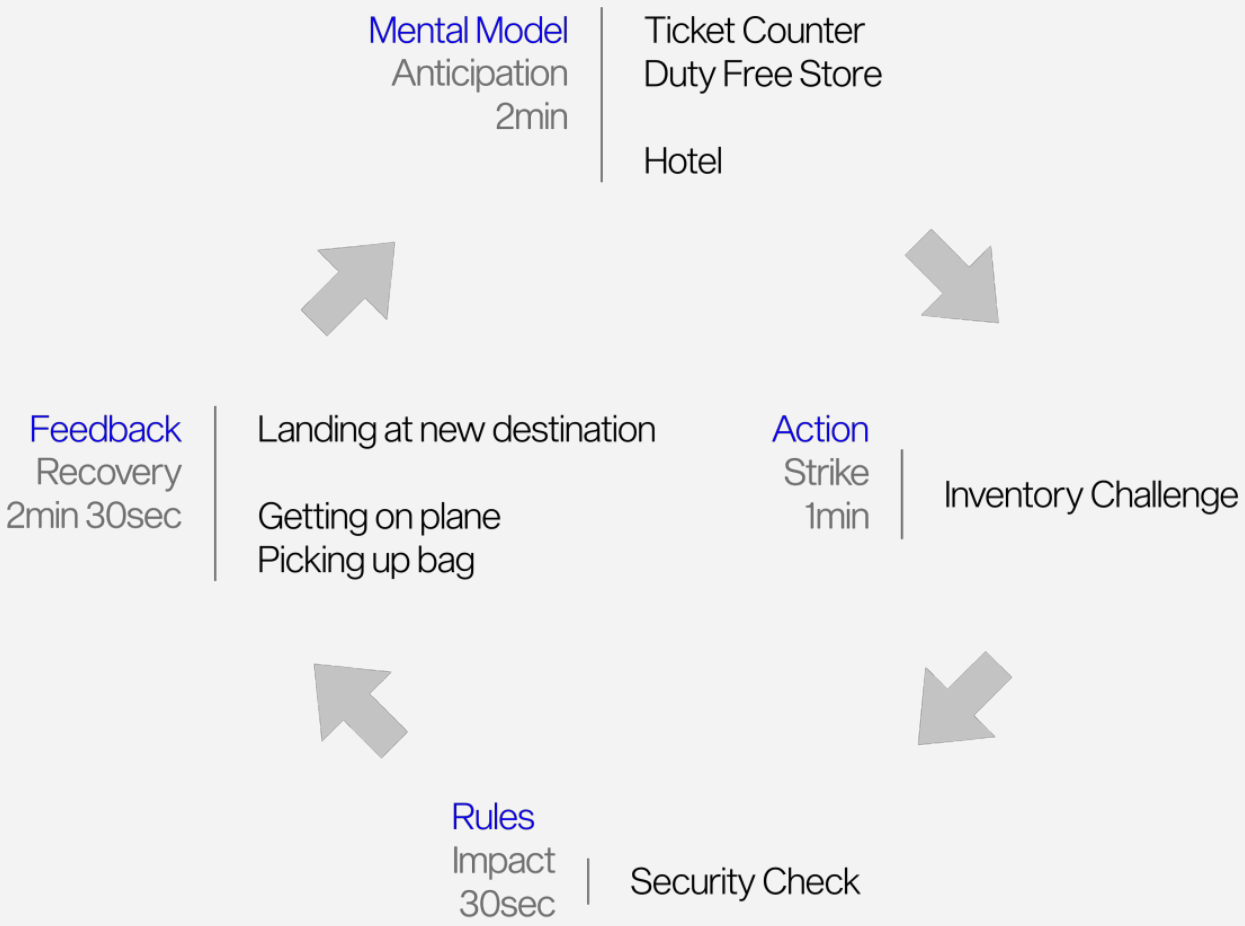
Below are diagrams that will be updated and omitted during the process of creating this -- it will be iterated in the same faucet as the game. This would also be the development log of sorts as well.

Core Loop



Represented are all actions that the player will likely take and visit over the course of a run.
Red – Fail, Yellow – Transition, Green – Safe, White – Active.

(Time per stage approximate goal)



Game Mechanics

Tetris-Style Inventory System:

- Tangible
- Explanation of Intended Effect:
  - Allows the player to store items, as well as hide items that they can smuggle.
- Core Action
  - The player will place, organize and rotate items in the inventory– such as in Resident Evil 4 and the gameplay of Tetris (in rotation).
- Technical Rules
  - It starts as a 7x5.
  - It can be increased to a medium (10x7) or large (14x10)
  - If an item can be hidden it will highlight green.
  - If an item can't be hidden it will highlight red.
- Potential Effects on Gameplay
  - Increased sizes of that inventory space will likely add more obstruction to the inventory. See the following: BagsPls_CanYouHideIt, check Inventory sheet.
- Potential Risks
  - The bigger risk is in the complexity of this mechanic–if using Unreal, I've found an asset that might mitigate some risk. Mitigation: Here
  - Creating this mechanic without on the go will need to be devised to be modular.

Game References for Mechanic:

1. Save Room, Organizational Puzzle
2. Backpack Hero

1 

2 

Inventory & Upgrades regarding Mechanic

Suitcase Upgrades:

1. What will happen when the suitcase is upgraded?

Upgrade (will create a minimum of five upgrades)	Possible Cons
Increased storage	There might be some blockers, obstructing specific grid sizes.
Protection against the scanner	It randomly protects one item that gets scanned, that would have otherwise been highlighted red.
Scent Blocker	Has a higher chance of negating weed from being discovered.

Scanner System:

- Tangible
- Explanation of Intended Effect:
    - An animation shows the player's luggage, if an item is not hidden it will be confiscated. The bag is scanned and the item shows up via an x-ray like vision to reveal poorly hidden items.
- Core Action
    - The player will press a button to scan the item, when the item scans, there will be a success/failure
- Technical Rules
    - Will likely use a type of hit raycast to check if it's a ‘hidden’, ‘unhidden’ object.
- Effects on Gameplay
    - Adds tension and a risk/failure to scanning.
- Potential Risks
    - Can easily become a little too complicated, vice versa, could be too simple of a mechanic.
    - Mitigation: Can be a shader effect, swap material event.
- Notes
    - Maybe also, an indicator to see if something is successfully hidden will be an x in the middle of the item if not and a check mark if it's successfully hidden. Or a type of crosshatch or diagonal lines, this will account for color blindness in players.
    -

Game References for Mechanic:

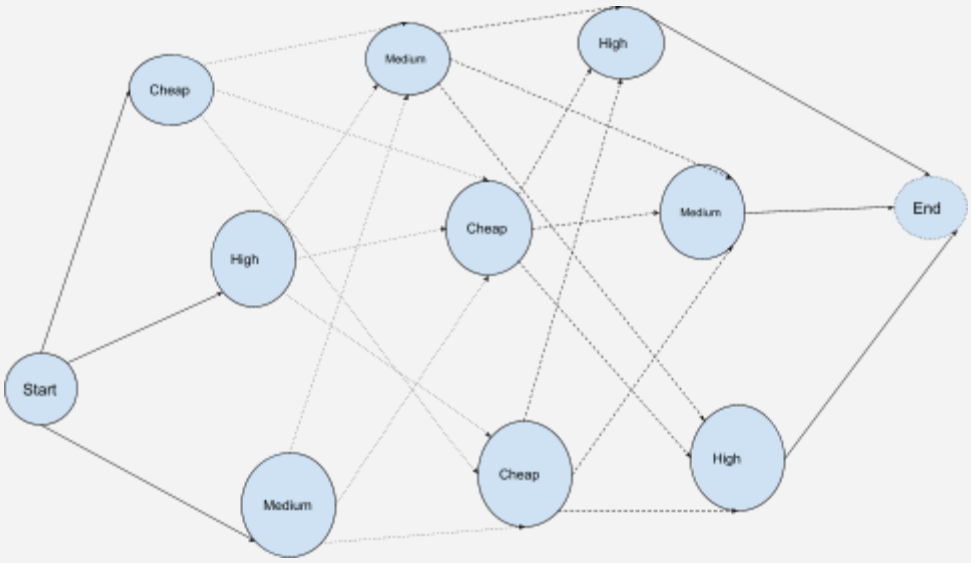
1. Operation X-Ray Match Up
2. Airport Scanner 2

Route-Traversal System:

- Tangible
- Explanation of Intended Effect:
    - The player will be able to determine what routes that they want to go through.
- Core Action
    - The player will select a series of ‘destinations’ to indicate a route which will be ‘x’ nodes.
- Technical Rules
    - Will use a specific graph-traversal algorithm–specific ones described below–this will be used to go through the destination nodes.
- Effects on Gameplay
    - Adds variability in gameplay destinations.
- Potential Risks
    - Figuring out how exactly to organize this– as a node– inside of the intended engine.

There are a few ways that navigation between destinations can work, the ‘best’ way to do that that will account for a loop will using a graph traversal algorithm:

- Breadth-First Search (BFS) algorithm will likely work the best as we aren’t calculating for how far away a destination is–as that can be determined purely by how much the next place is.
- Diiksta’s algorithm–if we wanted the shortest path to always be the cheapest one–in this and the next case it definitely will calculate how far away the next one is.
- A* (A star) algorithm, if we want an easier way to guide the user towards the ‘main’ destination. In this case, depending on how we want the game to be: that could be delivering all cash on hand to the debtors.



With the node graph traversal:
Start < cheap, high, medium < medium, cheap, cheap < high, medium, high < End

As like one route planned would be:
Start - Cheap
Cheap -> Medium
Medium -> High
High -> End

Example Route:

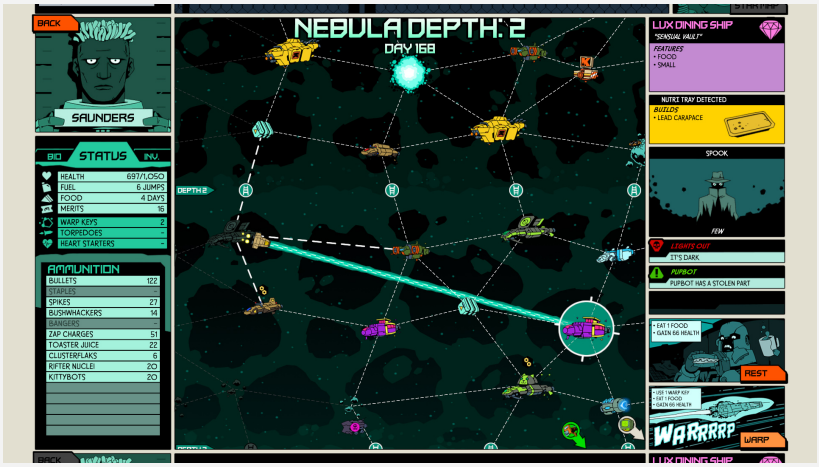
1. Start in Colombia. (Start)
2. Fly to Caracas.
3. Fly to the Santo Domingo Airport.
4. Finally fly to Miami. (End)

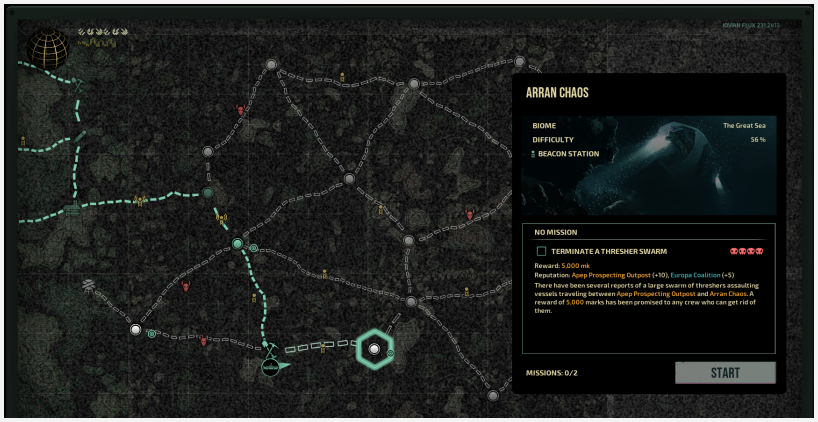
The player will then traverse a sequence of nodes to complete a successful smuggling operation. So, then– “it’s okay, one more smuggling run.”

Route-Planning for the loop– streamlines the ‘destination traversal’ process and maybe we can add a tidbit, that if some contraband is found it will add a delay to the time. So, the player has ‘x’ amount of days to reach the final destination after they’ve selected their route. However, in-game, specific other decisions that the player makes can add or detract from that time. As in, if contraband is found, they might be anywhere from a few hours to a day more added.

Game References:

- Slay the Spire
- Void Bastards
- Barotrauma


BACK
SAUNDERS
BIO
STATUS
INV
HEALTH 697/1,050
FUEL 6 JUMPS
FOOD 4 DAYS
MERITS 16
WARP KEYS 2
TORPEDOES -
HEART STARTERS -
AMMUNITION
BULLETS 122
STAPLES -
SPIKES 27
BUSHWHACKERS 14
SHRAPNEL -
ZAP CHARGES 51
TOASTER JUICE 22
CLUSTERFLAKS 6
RIFTER NUCLEI 20
KITTYBOTS 20
NEBULA DEPTH: 2
DAY 168
DEPTH 2
LUX DINING SHIP
"SENSUAL VALUE!"
FEATURES
• FOOD
• SMALL
NUTRI TRAY DETECTED
BUILDS
• LEAD CARAPACE
SPOOK
FEW
LIGHTS OUT
IT'S DARK
PUPBOT
PUPBOT HAS A STOLEN PART
• EAT 1 FOOD
• GAIN 50 HEALTH
REST
USES 1 WARP KEY
• EAT 1 FOOD
• GAIN 50 HEALTH
WARRRRP
WARP
LUX DINING SHIP


ARRAN CHAOS
BIOME
The Great Sea
DIFFICULTY
56 %
BEACON STATION
NO MISSION
TERMINATE A THRESHER SWARM
Reward: 5,000 mk
Reputation: Apep Prospecting Outpost (+10), Europa Coalition (+5)
There have been several reports of a large swarm of threshers assaulting vessels traveling between Apep Prospecting Outpost and Arran Chaos. A reward of 5,000 marks has been promised to any crew who can get rid of them.
MISSIONS: 0/2
START

Outfit System:

- Tangible
- Explanation of Intended Effect:
    - Outfits that the player equips will either bonuses or debuffs dependent on location and scanner passing.
- Core Action
    - The player will equip an outfit.
- Technical Rules
    - Outfits should take up a part of the bag space.
    - Outfits give bonuses++ and debuffs–; dependent on location.
        - Outfits are categorized based on location as: Good, Neutral, Bad.
    - Previously used outfits will give major debugs in the same destination; this should refresh every ‘x’ run.
- Potential Effects on Gameplay
    - What the player has equipped as an outfit allows for additional strategic choices to be made into the game.
    - If the player forgets, it will add some major debuffs.
- Potential Risks
    - Will need another art designer for the outfit concepts.

Phone System:

- Tangible
- Explanation of Intended Effect:
    - The player will be able to switch o
- Core Action
    - The player will place, organize and rotate items in the inventory– such as in Resident Evil 4 and the gameplay of Tetris (in rotation).
- Technical Rules
    - It starts as a 7x5.
- Potential Effects on Gameplay
    - Increased sizes of that inventory space will likely add more obstruction to the inventory. See the following: BagsPls_CanYouHideIt, check Inventory sheet.
- Potential Risks
    - The bigger risk is in the complexity of this mechanic–if using Unreal, I've found an asset that might mitigate some risk. Mitigation: Here

Player Onboarding

ONBOARDING EXPERIENCE

Onboarding Goal: Allow the player to learn basic controls before sending them out, a good onboarding doesn't need an in-depth tutorial. This will help with the casualness. This can be the pictured 'Miami Connection' smuggling run above.

1. The player will start with a bag given to them that they must smuggle–this will be the first instance where the inventory management and also the first step–a good first impression is needed.
2. After the player conceals the given items, they will be put on the next screen, where a predetermined destination will be chosen.
3. Then the player will transition to the bag-scanning, where the animation will be shown for the first time, red highlights meaning item found and green highlights meaning item successfully smuggled.

    a. Red highlighted objects are removed and the player is fined.
    b. Green highlighted objects have been successfully smuggled.
4. The player has successfully smuggled an item for the tutorial, and then will need to determine their route. The player will plot their route; they will then have access to the other menus.
5. Once the player finishes their route–the player onboarding is technically done.

Art Production

[]

Art Brief ← Feel Free to add things here: moodboards, color templates and etc.,

Audio Visual Brief (Miro)

Sound Design

Audio Benchmarks:

· **Games:**

o Disco Elysium;

o Neighbors From Hell;

o This is the Police

· **Films:**

o Who Framed Roger Rabbit;

Audio Pillars

Core gameplay built around a short game loop which divided on few steps:

1. Choice destination and your outfit;
2. Choice legal items;
3. Packaging;
4. Departure and upgrade your bag or arrest and bribe giving.

This game loop can be united by the adaptive music system and it will be a part which fully unites one game loop pass (about music form you may read in the full version of SDD).

Main mood of the game will be to reference us to a noir crime story with a lot of clichés during the story line. This decision will be supported by the music: noir jazz will be very atmospheric in this setting.

The action of the game take place on late 80's – early 90's. It means that except noir jazz music can use popular genres of this time like synthwave and synthpop.

Character of the music genres will be connected with the system of player's outfit (more accurately you may read this in the mechanics chapter). In the other way, the adaptive system will be connected just only with different parts of the game loop.

As was said, the main story in sound will be told by music. All other sound stuff will be divided on few not so large systems:

1. UI/UX;
2. Ambience;
3. Voice Over;
4. Foley.

All types of sounds should be realistic and accurately describe what the player sees on his screen: items in a bag, sounds in locations, scanner in an airport, sounds of different types of security.

Sounds in different locations should have a unique character of sound and a bit different general mix for a contrast and for better player immersion.

Technology

Engine: Unreal Engine 5.2.1;

Middleware: FMOD;

File Format: .wav, 44100 Hz, 24 Bit PCM. As close to -23dB LUFS;

Platform: PC (Windows);

Data budget for audio: no limits (on moment of 07.2023);

Multiplayer: none;

Spatialization needs: randomly spatialized additional layers of the ambience.

Audio Workflow

The audio team will have a different workflow for the work of composers and sound designers.

If all necessary mechanics are approved by the Lead Game Designer, composers may start to work with their tracks accordingly with an approved adaptive music system (check the Sound Identity/Music/Technology Information chapter).

Sound designers start to work after creating a prototype. If we'll have only one general audio department first task will be to create necessary tree of the sound system inside FMOD that include:

1. System of UI/UX:
    - Main Menu UI;
        - Hovers;
        - Press;
    - Settings Menu UI;
        - Hovers;
        - Press;
    - Pause Menu UI;
        - Duck effect on the UI bus for the all in-game sounds;
        - Hovers;

- Press;

- In-game UI/UX:
  - Outfit selection:
    - § List of the different outfits;
    - § Click;
    - § Back;
    - § Confirmation stinger;
  - Legal Items;
  - Illegal Items;
    - § List of the illegal items combined with a negative tonal tone;
  - Destinations;
  - Departures;
  - Clicks in shop;
    - § Hover;
    - § Cancel;
    - § Confirmation;
  - Drag’n’drop system in a bag;
    - § Selection;
    - § Confirmation;
    - § Cancel;
    - § Click;
  - Bug’s upgrade;
    - § Click;
    - § Hover;
    - § Confirmation;
    - § Cancel;
  - Bribe scene;

        - § Press;
        - § Confirmation;
        - § Cancel;

2. Ambiences:
    - Bag packaging;
        - Optional: the hotel sounds;
    - Shop;
        - With additional randomly sequenced layers;
    - Ticket corner;
        - With additional randomly sequenced layers;
    - Hotel;
        - With additional randomly sequenced layers;
        - With additional random voice overs;
    - Departure screen;
    - Jail screen.
3. Foley (Can’t be written now because we haven’t a prototype);
4. Music:
    - General:
        - Duck effect on the music bus for the all music during the open pause menu;
    - Main Menu:
        - Main music theme;
        - Crossfade to the music in ticket corner;
    - Ticket corner;
        - Music theme of the ticket corner (random selection);
        - Transition to the next part of the track;
    - Hotel:
        - Outro of the music theme of the ticket corner;

- Transition to the next part of the track;

- Going to the shop till the scanner scene:
  - End of the music theme of the ticket corner;
  - Selection of the next track accordingly with player's outfit:
    - List of outfits connected with the game switch of "radio stations" (random selection);
    - Transition to the extensive part of the track connected with the game parameter of the time on the timer;
    - Transition to the scanner scene;

- Scanner scene:
  - Build up with transition to the next scene;

- Final of the game loop:
  - Final of the "outfit" track:
    - Outro for the departure screen;
    - Outro for the jail screen.
  - Transitional to the random container with the music theme of the ticket corner.

More accurate information about necessary systems and accurate audio assets you may find at Sound List. Without a created sound system the sound creation is **forbidden!**

If we'll have separation on technical and audio sound designers technical sound designers should to got an approve from their team leader that all mechanics and supposed sound systems are ready for implementation. After this they can start to work with creation of the sound system. Audio Designer may to work with already created and approved sound systems by their Team Leader!

First of all, audio designers should get a short gameplay video for the audio prototype creation. Rules of designing of audio prototype:

1. All sounds should to exist in particular space, ambience in your videos is essential;
2. One video should to contains one done system or one done sound elements with or without variations (it depends from type of sound);
3. Created video should be sent to your Team Leader;
4. After approval of your work your sounds will be transferred to a particular technical sound designer.

Narrative Design

Ideas:

1. Wrapped around how much money has been generated via runs.
2. Wrapped around the number of runs successfully completed.

Based on these two, it can have triggers based on either how many runs have been completed, failed runs, net cash smuggled, and etc.

How should the narrative be baked into the game?

- Visual novel-based
    - Possibly only for a short start scene, middle scene and end scene.
- Phone communication-based?
    - Phone based goes with the theme of smuggling items.
    - Possible Interactions:
        - Phone can have a countdown-timer until needed to go to the security check for departure.
        - Phones can be used to pick the next smuggling route.
            - The main screen will show the destination map with possible routes but the phone can be used to actually select it.
        - Contact list for narrative–as in the phone will buzz or have a contrasting highlight if there is a message.
            - Priority Request– Smuggling a specific item / accept or decline
    - Phone can likely fit the overall aesthetic in a complimenting way without too heavy of a divergence of the general–2D theme.
    - Phone can also be an intuitive way of how many ‘lives’ the player has until they have to start over?
        - Green – Pristine phone
        - Yellow – Scuffed phone
        - Red – Damaged phone
        - Danger – Phone screen doesn’t work??

Beginning–

Middle—

End—

Current Questions & Possible Rules

Possible rules of the [Analog] Gameplay:

1. Baggage that the player will have as their inventory space will use tetris-styled inventory.
2. Items have a specific grid-size but that's not indicative of how much the item is valued, in fact, item payout will not correlate with item size.
3. Each Illegal item has an associated Legal Item that it can be hidden with.
4. Specific baggage cases can also have their own restriction types.
5. The starting base luggage size is: 7x5
6. The destination (chosen or random) can be tracked, allowing for specific special rules for destinations.

1st Batch of Questions–Game Design Related:

Questions to ponder will be answered as concept and development progresses. Would suggest making a quick analog (board and card game, paper) prototype to test the rules.

1. What determines how difficult it is to get through the security check.
2. How would items be obtained, are they randomly generated?
    a. Duty-Free Shop
    b. Randomly-Filled but real world logic
3. How will the players get the legal items?
4. How do you measure what makes a run successful?
    a. What will determine what makes a run harder?
5. Will items have an attached monetary value?
    a. Will these fluctuate? Pros/Cons. Yes/No.
6. What are the pros and cons to having a suitcase upgraded, what can be a negative feedback loop.
7. What happens when an item is found–is it more than one?
8. There is a cocaine bag, yes, but what if there is a bigger cocaine bag?
    a. How much would that change the grid-size by?
    b. How much would that increase the value by?
9. What can we do to limit how much the player can earn such as:
    a. Decreasing the end-value of the total cash earned upon successfully smuggling?
10. What can we do to have a risk / reward trade for specific systems?
11. What happens when you upgrade the suitcase and what possible rewards–can you lose the suitcase, and go back to the 7x5?
    a. What are the different types of baggage that can be obtained to clean?
12. Due to its 2D nature some dynamic randomness with events can be afforded, would this be a feasible want?
13. Is there a set amount of lives before it is completely over–or is it a one-shot thing each time?
14. What defines being able to hide something within' an object.
15. What if the player's character has their own comfort level–and maybe there is a menu for them to 'purchase items' while on flight, or the character just randomly does that.
    a. The trade-off would be that the player will start the HQ with less money overall.

16. Items may have an assigned weight that when it goes over that specific amount a fee will be introduced.
    a. What's in an item?
        i. Name
        ii. Description
        iii. Value
        iv. GridSize
        v. Weight?
        vi. LegalItems have a usage count?

17. For this, I'll use Tarkov's inventory system as an example; using a screenshot from my game. Let's say the following is true:
    - duffle bag–is a legal item
    - the augmentin is the pill-bottle
    - Vaseline is a normal legal item.
    - grizzly is a bag of cocaine
    - lcd will be the laptop

If I wanted to store the ecstasy inside of the pill bottle, I would drag and drop it into the augmentin. It will highlight green. As it's a valid combination.

I'd have nowhere to stash the cocaine as there is no powder object. Every object will highlight red. As none of the combinations are valid.

If I purchase or find a pistol, I can drop that into the LCD and hide it in that.

If the duffle bag is a lunchbox: inside of it, I can store both the heart and the organ box.

*Note: that the hand drill can be vertical or horizontal, but either way, it's taking up half of the inventory space.

18. So far, there are three gameplay systems/mechanics: inventory puzzle, monetary system and traversal to destination.

Technical

Coding Principles and Technical Writing

Variable Rules → Coding conventions that will be followed:

- Ex. UPPER_CASE = Number Variables

File Structures → Determine how files will be organized, for example:

- Root Dir:
    - Core Assets
        - Gameplay
            - Blueprints
    - Art Assets
        - Terrain
            - TerrainCity_Road
        - Props
            - MainProp_Terminal_Escalator
    - Imports

Technical Management

Version Naming → 1.X.XX.

1. This will be the major build.
2. X is the major system addition, change or rework.
3. XX is small patches, bug-fixes and general.

ArtAsset_Naming → MainTypeSubType_General

As in what will this be:

TerrainCity_Road?

MainProp_Terminal_Escalator?

System Requirements

Part	Minimum	Recommended
Processor		
Memory		
Graphics		
Storage		

Vertical-Slice Checklist

What is needed for a concept-checking version of the game. Feel free to add more.

- ☐ Route-Traversal System
- ☐ Scanning – X-Ray System
- ☐ Inventory Puzzle Management System
- ☐ Outfit System (Maybe– might be an offshoot of the Inventory System)
- ☐ Defined UI
- ☐ Core-Loop

Milestones

Milestone 1:

- Core Features are implemented with basic placeholders.
- Screens are concepted, determined and implemented.

Milestone 2:

- Placeholder.
- Placeholder.

Milestone 3:

- Placeholder.

- Placeholder.

Milestone 1 Playtest Report

Individual Playtests

Playtest # 1

Name:
Age:
Gender:
Gaming Experience: N/Y
Observations:

-

Length of Play:
Got Stuck?:
Overall Difficulty:
Overall Enjoyment:
One thing they liked?: N/A
One thing they would change?: N/A

Action Plan

1. Based on conclusions, what are you going to do about it?
    -
2. Should be SPECIFIC **MECHANICAL** changes not generalizations
    -
3. How do you expect this to change the player experience?
    -
4. What will you add to/change in the playtest process (form) to make sure?
    -