



GAME DESIGN DOCUMENT 1.2

INDEX

GAME OVERVIEW	3
SUMMARY	3
GENRE	3
PLATFORM	3
TARGET AUDIENCE	3
MONETIZATION	3
LOOK AND FEEL	3
MINDSET	3
REFERENCES	3
PLOT	4
SCREENS	5
BOOT-UP SCREEN	5
TITLE SCREEN	5
LEVEL LOADING SCREEN	6
GAME SCREEN	6
RESULTS SCREEN	7
MENU SCREEN	7
PROGRESS BAR AND PRIZES	8
LEVEL DETAILS	8
CREDITS AND SETTINGS	9
SKIN SELECTION	9
GAMEPLAY	10
CORE LOOP OVERVIEW	10
CHECKPOINTS	10
EXTRACTION POINT	11
TOKENS	11
DEATH	11
INTERSTITIAL ADS	11
MOVEMENT GRID	12
THE PLAYER CHARACTER	12
MOVEMENT	12
ACTIONS	13
HIT A GUARD	13
OPEN A DOOR	13
THE GUARDS (AI AND THE STEALTH SYSTEM)	13
SIGNALS	13
CONE OF VISION	14
AUDIO SIGNALS	14
REACTION TO SIGNALS	14
ALERT MODE	14
GETTING HIT	15
PATROLS	15
TABLE OF STATES	15
BASIC LEVEL OBJECTS	16
OBSTACLES	16
SURVEILLANCE CAMERAS	16
NOISY FLOORS	17



HOLES IN THE FLOOR	17
SPECIAL LEVEL OBJECTS	17
LASER EMITTERS	17
SWITCHES	17
VENTS	17
GAS	17
LEVEL DESIGN	19
LEVEL DESIGN OVERVIEW	19
BASIC STRUCTURE OF THE LEVEL	19
TYPES OF CHUNKS	19
CHUNK DESIGN NOTES	20
PROCEDURAL GENERATION	22
CHUNK POOLS	22
CHUNK ORDER	22
FINAL CHUNKS	22
THEMES	23
SELECTING A THEME	23
TILESETS	23
ADDING NEW CONTENT (UPDATES)	23
ONBOARDING	24
SCREEN FLOW FOR FIRST-TIME USERS	24
BOOT-UP, TITLE SCREEN AND LOADING SCREEN.	24
TUTORIAL LEVEL	24
RESULTS SCREEN	24
MENU	24
FIRST DAY AD EXCEPTION	24
RATE OUR GAME	24



GAME OVERVIEW

SUMMARY

SNEAK OPS is an action-arcade stealth game with a new level every day. You take the role of a special forces soldier who is assigned the task of secretly infiltrating military bases to gather information while avoiding detection by guards and other security systems. When caught, the player will be returned to the latest unlocked checkpoint. The player can collect floppy disks that will be spent on checkpoints.

GENRE

Action / Stealth

PLATFORM

Android and iOS.

TARGET AUDIENCE

Ages 11 to 35, with a console and mobile gaming background.

MONETIZATION

Primarily ad-based monetization. Every 2 to 3 minutes, when the player loses, an interstitial ad will be displayed. Also, the player can watch video ads to unlock the levels from the previous days, or unlock a checkpoint when he doesn't have enough floppy disks. There are also some in-app purchases: users can buy an ad-free version (with free checkpoint saving) by paying £4.49, and customize their character purchasing skins at different price points.

LOOK AND FEEL

Low res, chunky and cartoony pixelart graphics with voxel-based characters that are seamlessly integrated in the pixelart aesthetic. Top-down perspective (classic 8-16bit action-rpg perspective, like Zelda or Metal Gear series).

The sound and music is light-hearted and cartoony as well, while also fitting the stealth theme.

MINDSET

SNEAK OPS is meant to be played anywhere: while on the train to/from work, while waiting for a friend, or at home in the couch, the bathroom, or before going to sleep. The game aims to create the habit of playing 10 to 20 minutes of SNEAK OPS every day. It is a game based on repetition, trial and error, planning, puzzle-solving and overcoming a daily challenge.

REFERENCES

Leap Day, Metal Gear Solid: Ghost Babel, Zelda: Phantom Hourglass.

PLOT

A soldier from the Sneak Operations Force has been dispatched to a new mission!



Avoiding surveillance cameras, guards, and other devices and traps, Lucas and the other SNEAK OPS officers will infiltrate all kinds of secret bases they are thrown in, gather all the information and then sneak out and board the helicopter to come back home.

SCREENS

BOOT-UP SCREEN



The game will start with a static Boot-up screen, in which the system is loading the application. When finished, this will automatically lead to the game's title screen.

TITLE SCREEN



This is a splashscreen with animated elements and a blinking START sign. Tapping anywhere on the screen will lead directly to today's level.

LEVEL LOADING SCREEN



While this screen is displayed, the game will load the assets and generate the level, using the date of the level (YYYY/MM/DD) as the seed for the random number generator. Any saved progress will also be read at this point.

A transition will be played to leave the previous screen, then an animated loop and then another transition to enter the next screen. The minimum loading time will be the time it takes to play both transitions.

A transition will be played to leave the previous screen, then an animated loop and then another transition to enter the next screen. The minimum loading time will be the time it takes to play both transitions..

GAME SCREEN



This is the main screen, where the core gameplay happens. It has a HUD displaying the diskettes to spend in this level, and menu button to exit to the menu screen.

When starting a level, a series of tooltips will be displayed: One that shows how to move by tapping on an empty tile, another to teach how to open doors, and another to show the player he can knock enemies out. Each tooltip will be displayed until the player performs the corresponding action, and only then the next tooltip will be shown.

Tooltips will be displayed the first three days since the player has started playing or has come back to the game after a 30+ day lapse.

RESULTS SCREEN

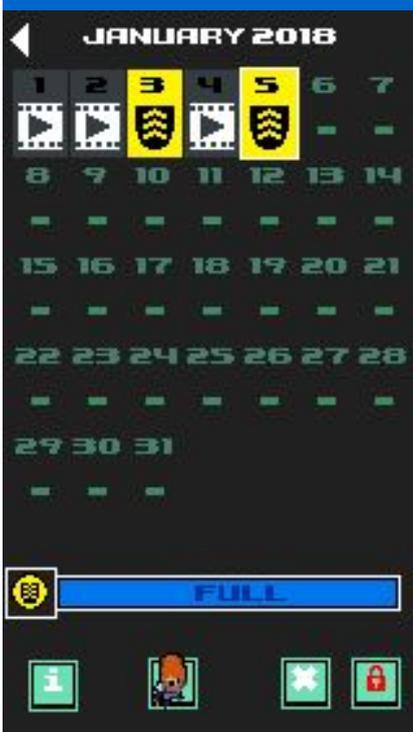


When the player reaches the extraction point, he will see the end level results screen. This screen displays the % of floppy disks the player has collected in this level, a celebration header, a blinking text 'TAP TO CONTINUE' and a picture which is a button to generate a screenshot with the level result for sharing with friends in social media or other apps like IM apps.

In the background, a helicopter will pick up the player character and leave the screen.

Tapping on the screen will switch the game to the menu screen.

MENU SCREEN



On the menu screen, a calendar will be displayed. This will display the levels this month, with different buttons indicating the badges collected on each level.

Below the calendar there is a SNEAKING POINTS progress bar and buttons for Achievements, Platform's Games Services, Skin Selection and Settings+Credits screens. In the top right corner, a button to resume the current game will be shown.

Swiping left or right will display the month after or the month before the current.

PROGRESS BAR AND PRIZES

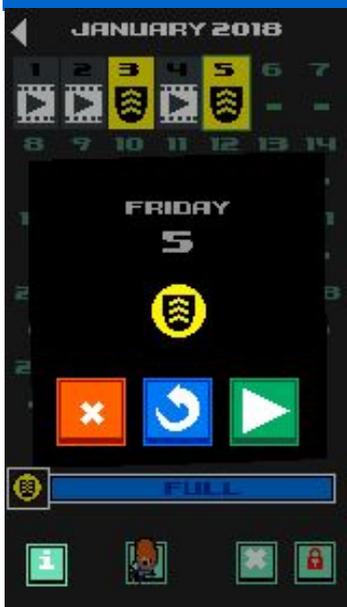


The progress bar indicates how many SNEAKING POINTS have been collected and how many are needed to get the next prize. Every time the player goes back to the menu screen having earned new rewarded badges, this progress bar will be updated: each rewarded badge that hasn't been accounted for will be shown and the bar will grow, one badge at a time.

Every time the player earns 8 SNEAKING POINTS and fills the bar, he will be rewarded with a prize box that contains a new skin, selected randomly between the skins available. When the player already has unlocked that skin, a 'DUPLICATE' sign will be displayed.

Rewarded badges are given only in today's level.

LEVEL DETAILS



Tapping on any of calendar day buttons will show the level details window, in which the player can continue or restart the level. Restarting the level makes all floppy disks respawn, and saving a checkpoint will overwrite the previous saved game for that level but if the player doesn't save, he will keep the previous saved version. Continuing a level means loading the latest saved game for that level.

If the player doesn't play the level any given day, the level for that day will be locked and he will have to watch a video-ad to play it. Levels for days in the future will be unavailable for all players.

CREDITS AND SETTINGS



On this screen, credits are displayed in the top part of the screen.

On the bottom part, buttons to toggle SFX, Music, Log in and out of the platform's game services and Reload any purchases that might be unloaded after reinstalling the game.

SKIN SELECTION



The Skin Selection window has a page for each skin and can be navigated with the page up/down buttons.

If the skin has been unlocked, the player will have the option to select that skin and close the window, and an illustration for that skin will be displayed.

If the skin is locked, a silhouette version of the illustration and a lock will be displayed, and the player will have the option to

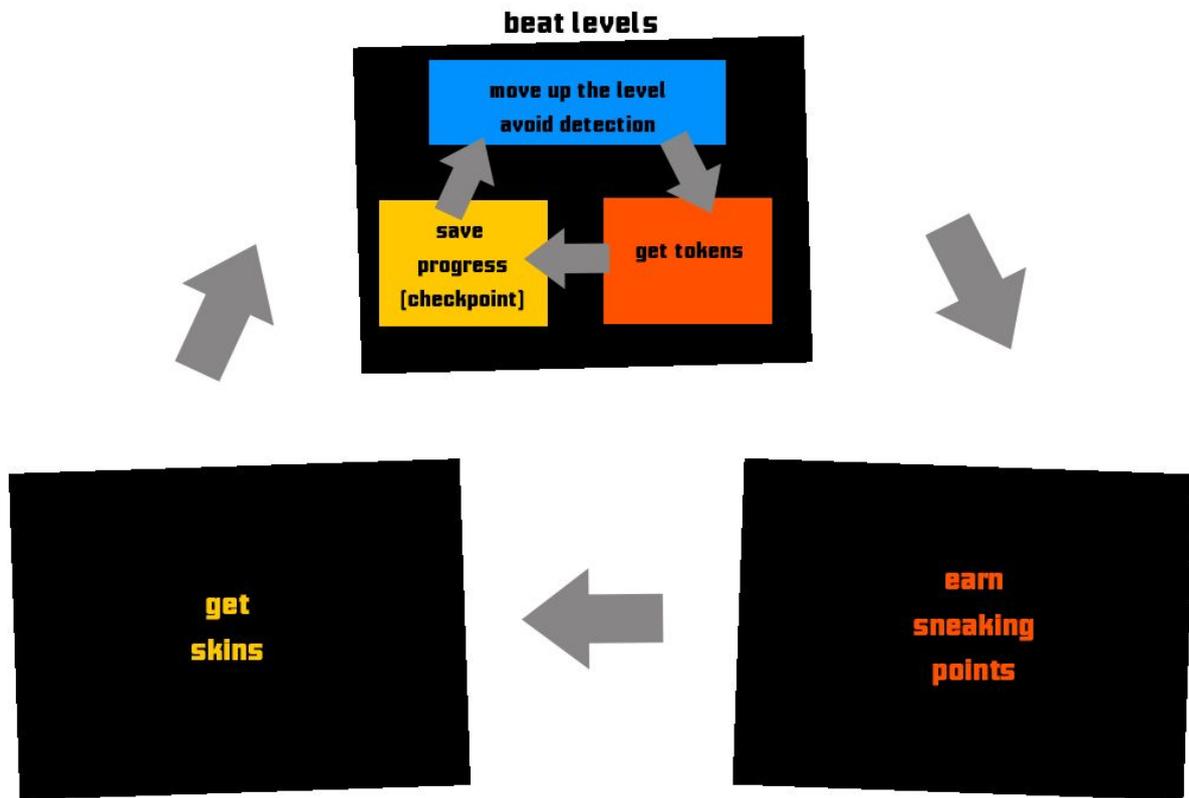
purchase the skin with hard currency.

The name of the skin will be shown in both cases, and the pages will loop (going up the first page will show the last page and the other way around)

GAMEPLAY

CORE LOOP OVERVIEW

In Sneakout, the player has one main goal: reach the extraction point at the end of the level, every day. In order to do so, he must move the character up the screen, avoiding detection by the guards and detection devices like surveillance cameras and lasers.



CHECKPOINTS

Checkpoints are safe zones where the player can save progress. When stepping on a checkpoint save switch, the player will be asked if he wants to save that checkpoint and how to pay to activate it. The player can pay by watching a video ad or spending 20 floppy disks. If the player has paid to remove the ads, saving a checkpoint is free of any charge and the only button displayed will have a single diskette icon. There are 12 checkpoints on each level.

Reaching checkpoints number 4, 8, and clearing the level give the player one sneaking point each and a bronze, silver, and gold trophy in the calendar, respectively. If the level is today's level, the badge will be rewarded with one sneaking point. When completing the level, if the player has collected 100% floppy disks from that level, the player will be given a star badge, rewarded with one sneaking point.

When the game is saved, the checkpoint number, the tokens the player has collected so



far, and the number of tokens still in his possession, are stored. This information is loaded when entering the level.

Badges are saved in the moment they are obtained, not taking into account whether the player has saved the checkpoint or not.

EXTRACTION POINT

The extraction point is a special kind of checkpoint at the end of the level. Stepping on it autosaves the game at checkpoint 1 and activates the results screen.

TOKENS

In the way up, the player will find tokens in the form of floppy disks. These tokens allow the player to save progress when they reach a checkpoint, and if the player beats the level having collected all the tokens, he is rewarded with a star badge, which is worth 2 sneaking points.

Once the player has collected a token, it is saved for that level, and it will not respawn anymore. However, if the player leaves the level, any tokens collected after the last checkpoint saved will be restored to their spawning position when the player enters the level again, and the player will have the same amount of tokens he had after saving for the last time.

DEATH

When a guard points a gun at the player character for long enough, the guard shoots and the player dies. When this happens, the game resets:

- Guards get teleported to their spawn point and reset to their initial conditions.
- All level objects are reset to their initial conditions.
- The player character will be teleported to the last checkpoint saved, or the initial spawn if no checkpoints have been saved.

INTERSTITIAL ADS

When the player dies, sometimes an interstitial ad will be shown, depending on the following rules:

1. At least 1 minute must have passed since the last time the game showed an interstitial ad
2. The player has died at least 4 times since the last interstitial ad
3. If more than 2 minutes have passed since the last interstitial ad, an ad will be shown regardless of rule 2.

MOVEMENT GRID

The game works with two grids of 16x16 pixel tiles, with a width of 11 tiles and a height that depends on the procedural generation of the level. Every movement of the game is calculated using one of these grids, creating paths between two tiles. The game camera is placed with an offset of 8 pixels, so 9 full columns and half of the side columns are shown at all times.

The tiles on these grids can be either free or occupied. This can change during the game, depending on doors and other activable or moving objects.

One of the grids is the Player Grid, while the other is the AI Grid.

THE PLAYER CHARACTER

MOVEMENT

Moving the player character to the extraction point is the main goal of the game. The player always follows his Chase Path, which is calculated using the Player Grid. This is performed by tapping anywhere on the screen, selecting a destination tile:

- a) If the tile is free, it is selected as the destination tile and the chase path is calculated from the current position to the selected destination.
- b) If the tile is occupied but there are any free tiles around the occupied one, and there is a path to any of those tiles from the current position. The tile with the shortest path is selected as the destination tile and the path is chosen as the chase path.
- c) If a path to the tile or any of the surrounding tiles does not exist, check if there's a path to tile 2 rows below the selected one, and follow that path.
- d) If no path was found, the chase path is void and the player character remains in the current position, displaying a forbidden sign on the selected tile.

When the Chase Path is empty, the player will enter the idle state and stand still in the current tile. However, if any of the 8 surrounding tiles is occupied in the player grid, the player will switch to the wallhug state, which is a special case of idle with the player character hugging the wall for better gamefeel.

The player character will follow the chase path with a speed determined by the constant `PLAYER_SPEED`, leaving a trail of particles behind him.

If the Chase Path crosses a door, the player will perform the 'Open Door' action instead.

The current chase path and destination tile will be indicated on the screen while the player is walking. Players can preview the chase path the player character is going to take to a

specific tile by holding their finger on that tile for a few moments.

Movement controls can be disabled so players can't make the player character move under special circumstances like the results screen or the checkpoint save screen.

ACTIONS

The player character can also perform 2 kinds of actions besides moving: hitting guards, and opening doors.

HIT A GUARD

Players can command the player character to hit a guard by simply tapping on a guard. While there is a viable chase path, the player character will follow the guard until it is at a distance shorter than `PLAYER_ATTACK_RANGE`. When the guard is in range, both the player character and the guard stop moving, the player grabbing the guard while performing his punch, with a duration determined by the constant `ATTACK_DURATION`. While this movement is performed, the target enemy is incapacitated (grabbed by the player) in the "immobilized" state. When the attack ends, the player returns to idle state.

If a guard that is immune to damage is hit, it will turn around to look at the player and the text 'IMMUNE' will appear over the guard's head for a few moments.

OPEN A DOOR

When the player approaches a door, it opens. If the door is locked, access will be denied showing a sign to the player, and the door wont open.

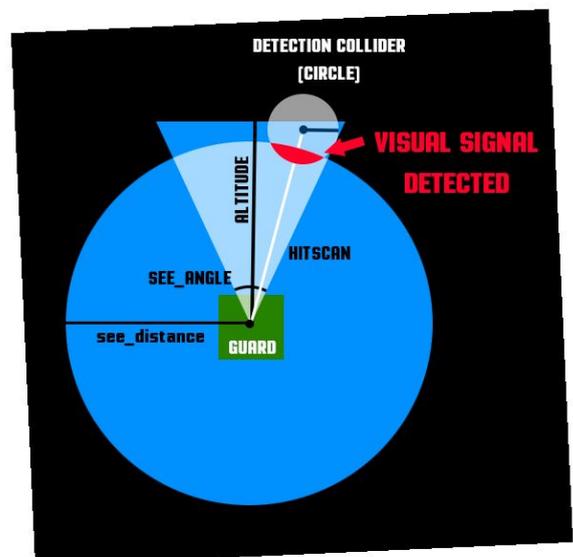
THE GUARDS (AI AND THE STEALTH SYSTEM)

Guards controlled by AI are the main obstacle the player will face when trying to get to the Extraction Point.

SIGNALS

Guards respond to 2 kinds of signals: Visual and Audio signals:

- Visual signal: when the guard's cone of vision covers the detection collider, a circle with radius `DETECTION_COLLIDER_RADIUS` centered in the position (x, y) of a visual signal. They have maximum priority.
- Audio signal: when the guards enter the area of effect (distance) of an active



audio signal. They have medium priority.

CONE OF VISION

The cone of vision of a guard is the intersection of an isosceles triangle with a vertex angle defined by the `SEE_ANGLE` constant, and a circle with a radius defined by the `SEE_DISTANCE` constant. The triangle must have an altitude at least equal to `SEE_DISTANCE`. Also, a hitscan between the AI and the visual signal must be performed to determine if there is any obstacle between the signal and the guard. If both conditions (the detection collider is inside the cone of vision and the hitscan is successful) are met, then the guard is detecting the signal.

AUDIO SIGNALS

Audio signals come from a signal emitter, which can be the player or an object. They differ from visual signals in that audio signals only last for a few game steps before disappearing. Audio signals check for enemies inside their area of effect, with a radius defined by `SOUND_DISTANCE`.

REACTION TO SIGNALS

Guards only react to signals when they are not already in the “React to signal” state or if the new signal is higher priority, or of equal priority and far enough from the previous one, as determined by `NEW_SOUND_DISTANCE`.

When a guard receives a signal, he enters the “React to signal” state, stopping for a moment that lasts as defined by `REACT_TIME`, and displaying a question mark over his head. Then, if there's a viable path between him and the signal, he enters the “Go to signal” state and follows the path to the signal. If there's a direction from which the guard can see the audio signal coordinates, the guard won't move, but turn around to look at those coordinates.

Once the guard has arrived to a signal and there isn't another signal to check, he enters the “Confused” state, displaying a question mark over his head and looking at his right, then at his left, and then back to the front with his cone of vision. After this visual check, the guard enters the “Reset patrol” state and goes back to his spawn point.

ALERT MODE

When a guard (or any other detection object such as a laser or surveillance camera) detects the visual signal of the player character, alert mode is activated. While alert mode is active:

- All guards go chasing the player character's current location.
- While a guard is seeing the player, it counts down from `SURRENDER_TIME` to 0.
- When the countdown reaches 0, the guard shoots, the player dies and the game is



reset to the latest save.

- When the player is not being visually detected, that bar is emptied and the alert mode counts down from `ALERT_MODE_TIME` to 0.

Alert mode ends when the count down reaches 0, or the player hits a checkpoint or the extraction point. When alert mode ends, every guard receives a signal to investigate the position in which the player character was located when alert mode ended.

GETTING HIT

When a guard is attacked by the player character, he enters the “immobilized” state until the attack has finished. Then the guard enters the “flinch” state for a brief moment determined by `ENEMY_FLINCH_DURATION` before entering the “stunned” state for `ENEMY_STUNNED_DURATION` seconds. Then he enters the “get up” state, then the “Confused state” and then “Reset Patrol”. In the states of “flinch”, “stunned” and “get up” a guard can't detect any signals or move.

PATROLS

This is the default state of the guards. The guard walks towards the next point of his patrol path, and when he arrives, enters the “check sides” state before continuing his patrol. After checking the second side, the guard will look at the next point in the path. When reaching the last point of the path, the path loops and the guard patrols to the first point.

If the next point of a path is a `WAIT WAYPOINT` the guard won't check sides and then resume the path. Instead, he will look in the direction of his next waypoint and wait for `WAIT WAYPOINT_TIME` before resuming the patrol.

TABLE OF STATES

STATE	ANIMATION	ACTIONS	PERCEPTIONS
Patrol	Walk	Walk to the next	All signals

		point of the patrol path.	
Reset Patrol	Walk	Walk to the spawn point	All signals
Check Sides	Idle (maybe Pivot?)	Rotate 90 degrees clockwise, then 180 counter-clockwise, then look at the next waypoint	All signals
Go to signal	Walk	Walk to the last signal	All signals of higher priority or equal priority and distance further from previous signal than [NEW_SOUND_DISTANCE]
Confused	Idle (maybe Pivot?)	Create an obj_question, Rotate 90 degrees clockwise, then 180 counter-clockwise, then 90 degrees clockwise.	All signals
React to Signal	Idle (maybe React?)	Create an obj_question, wait a moment	All signals of higher priority or equal priority and distance further from previous signal than [NEW_SOUND_DISTANCE]
Chase	Walk (maybe Run?)	Walk faster to the player position	No signals except seeing the player
Aim	Aim	Wait	No signals except seeing the player
Full Reset	Idle	Teleport to spawn	No signals
Flinch	Flinch	Wait	No signals
Stunned	Stunned	Display sparks that rotate slower as the guard gets ready to get up	No signals
Get Up	Get Up	Wait	No signals

Immobilized	Immobilized	Wait	No signals
-------------	-------------	------	------------

BASIC LEVEL OBJECTS

Levels are full of objects that make the player adapt his strategy to them. They can be hazards that make traversing the level harder or tools that help the player character sneak past the guards to the extraction point.

OBSTACLES

Obstacles are invisible objects that are covered by scenery and occupy all the tiles they cover in the movement grids, permanently.

SURVEILLANCE CAMERAS

Surveillance cameras are suspended in the air, so they don't occupy any tiles on the movement grids. They have the same cone of vision and detect the player in the same way as the guards, except there is a blind spot under them, a circle with a radius determined by `SURVEILLANCE_BLIND_SPOT` where they can't detect the player.

If rotation is set to 0, the camera will remain pointing in the same direction.

If rotation is set to -1 or 1, cameras rotate until the camera is at a -90 or 90 degree angle with the arm. At that point, they reverse their rotation direction until the same condition (but in reverse) is met, so they rotate back and forth. When a camera detects the player character, it stops rotating until alarm mode has ended.

Cameras can also move in a straight path between two camera waypoints, back and forth.

NOISY FLOORS

Noisy floors are objects on the floor that don't occupy any tiles in the movement grids. However, if the player is walking on them, they will create a sound that emits an audio signal nearby guards will receive. When this happens, that noisy floor tile can't be activated again for an interval of `NOISY_FLOOR_COOLDOWN` seconds.

HOLES IN THE FLOOR

There can be holes in the ground, areas that act as obstacles in the movement grids, but don't occlude field of vision of guards or surveillance objects. Holes in the floor are always delimited by a warning line in the floor.

SPECIAL LEVEL OBJECTS

Special level objects are present in special type chunks. Special type chunks can only have one kind of special level object, they are mutually exclusive. Switches are an



exception to this, because they can be in the same chunk as pink lasers, but never with red lasers.

LASER EMITTERS

Lasers emitters send a laser beam from their starting point to the right until the beam meets a laser receiver object. If this laser beam touches the player character, the game enters alert mode. When in alert mode, the player character touching one of these beams will reset the alert mode countdown back to `ALERT_MODE_TIME`. Laser emitters and receivers are also obstacles.

Laser emitters can switch from on and off states, with an off and on duration determined in their creation code. Their indicative pilot blinks faster when they are about to switch states.

Lasers can be red or pink. Pink lasers never shutdown unless deactivated by a switch.

SWITCHES

Pink switches on the floor can activate and deactivate objects they are connected to, indicated by pink power lines on the floor.

The duration of the switch activation can be set in the level editor. Switches can unlock Power-locked doors and shut pink lasers down. When the switch goes back into normal state, any connected objects will be powered up again (lasers will be activated again and doors will become power locked again).

VENTS

Vents are holes in the wall that teleport the player to another vent that is connected by a small conduct, indicated by color. When the user taps on a vent, the player chases it and then crouches and crawls inside, coming back out of the connected vent.

GAS

Gas areas make the player consume Oxygen, at a rate defined by the constant `PLAYER_O2_DISCHARGE_RATE`. Current Oxygen level will be displayed over the player character's head if it is not equal to `PLAYER_O2_MAX_CHARGE`. When the Oxygen level reaches 0, the player dies and the game resets. If the Oxygen level is not full and the player is not inside a Gas area, the player will gain Oxygen at a rate defined by the constant `PLAYER_O2_RECHARGE_RATE`. An alert will be shown when the Oxygen level is under the value defined by `PLAYER_O2_ALERT`. Gas areas always cover a confined space (delimited by walls and doors).

LEVEL DESIGN

LEVEL DESIGN OVERVIEW

Sneakout has a different level every day. These levels are procedurally generated by stitching together a series of handcrafted level chunks, using the date of the day as the seed for the random number generator.

BASIC STRUCTURE OF THE LEVEL

A level is made up of 12 normal checkpoints plus the extraction point at the end of the level, numbered with a descending order, starting with 12 at the start and 1 just before the final challenge. Before every checkpoint, 2 handcrafted level chunks are placed. Before the extraction point, there's a final challenge chunk. At the start of the level, before the first 2 chunks, there's a starting chunk and in the middle of the chunks 12 and 11, there's another 'easing' chunk. Every level follows this structure:

FIRST CHUNK	SECOND CHUNK	THIRD CHUNK	CHECKPOINT
Chunk 12-A (start)	Chunk 12-1	Chunk 12-2	Checkpoint 12
Chunk 11-1	Chunk 11-A (Easing chunk)	Chunk 11-2	Checkpoint 11
Chunk 10-1	Chunk 10-2	N/A	Checkpoint 10
Chunk 9-1	Chunk 9-2	N/A	Checkpoint 9
Chunk 8-1	Chunk 8-2	N/A	Checkpoint 8
Chunk 7-1	Chunk 7-2	N/A	Checkpoint 7
Chunk 6-1	Chunk 6-2	N/A	Checkpoint 6
Chunk 5-1	Chunk 5-2	N/A	Checkpoint 5
Chunk 4-1	Chunk 4-2	N/A	Checkpoint 4
Chunk 3-1	Chunk 3-2	N/A	Checkpoint 3
Chunk 2-1	Chunk 2-2	N/A	Checkpoint 2
Chunk 1-1	Chunk 1-2	N/A	Checkpoint 1
Final Chunk	N/A	N/A	Extraction Point

Safe VeryEasy Easy Medium Hard VeryHard

TYPES OF CHUNKS

There are six types of chunks: Start, Easing, Checkpoint, Extraction Point, Normal Chunks

and Final Chunks. Start, Easing, Checkpoint and Extraction Point chunks are safe so there are no detection objects nor guards, and isolated, so nothing in other chunks affects them, and only the player can enter them.

- **Start chunk:** This is a unique chunk placed at the start of every daily level. In this chunk, the plot and goal of the game should be hinted to the player. In this chunk, the player will be taught how to move and open doors.
- **Easing chunk:** This is a unique chunk placed between chunks 11-1 and 11-2. This chunk is used for pacing, to introduce the daily level smoothly, give the player a small break and also, give some free tokens so he has an abundance of tokens at the start, making it easier to save checkpoints in the early stages of the game.
- **Checkpoint:** This chunk has a very visible sign showing the checkpoint number, and the checkpoint button placed in the centre of the chunk.
- **Extraction point:** This chunk shows an outdoors environment. The extraction point button is placed at the top of this chunk. This chunk contains the results screen object that is activated when activating the extraction point button.

Normal chunks and the Final chunks are where the action happens, with detection systems and guards looking for the player.

- **Normal chunks:** They are ranked by difficulty tier. This difficulty tier will be judged by the level design team. They are separated by mechanics. The Basic set includes lasers, surveillance cameras and noisy floor tiles, which can be present on any chunk. Then, there are lists for normal chunks using each major mechanic: Lasers, Switches, Gas, and Vents. Every day, two of this mechanics are used in the level.
- **Final chunks:** This chunk uses one of the two special mechanics for the level of that day, and is never repeated again until every other final chunk has been used. Final chunks have a special floor tileset.

CHUNK DESIGN NOTES

When designing chunks, there are some general principles to consider. These are not rules, just some considerations to take into account:

- The camera is offset to show more space towards the upper part of the screen. This makes downwards movement more difficult, which is not always bad and in certain cases can be what the designer wants sometimes.
- Checkpoints cost 20 tokens to save, and there are generally 2 chunks per checkpoint, so the normal thing to do would be having around 10 tokens per chunk if we wanted the player to always have tokens to save. Instead, we want to force the player to take the risk of not making the next checkpoint or avoid that risk by

watching an ad, so we need to have a token deficit in the game. Very Easy and Easy chunks must have 8 tokens per chunk, while Medium and Hard chunks must have 6 tokens.

- The playfield is only 9 tiles wide (plus walls) but infinite tiles long. Normal chunks are 21 to 42 tiles long, and final chunks are between 63 to 84 tiles long.
- When designing chunks, they can be subdivided in smaller isolated subchunks with safe zones that act as soft pseudo-checkpoints to adjust difficulty.
- Symmetry is not a priority and can be harmful in the design of the chunk.
- When there's only one mean of detection (a single guard, or a single camera or laser) in the chunk, it can be almost impossible to fail that chunk.
- Chunks with only one possible solution are not necessarily harder, and they can be more boring than having multiple solutions.
- Difficulty can escalate quickly when there are many moving parts.
- Noisy floor can be a pain in the ass for the player.
- Lasers slow players down and can become a chore if not combined with other mechanics that add the right kind of tension to be entertaining or challenging, and not just a drag.
- The player character can move in the rows superior to a one-way door, then go down and be able to pass the previously blocked door.
- Sometimes, it's nice to let the tokens become their own challenge, while making traversing that chunk the easy part. Let the player take the risk and the reward.
- It's important to add some variety in the scenery and decoration, otherwise it becomes boring and doesn't give the player the feeling of progress. Feel free to experiment and use props and tilesets in unexpected ways. However, always keep the floor tiles and framing consistent.
- Patrol paths can be closed laps, they can loop in a more linear manner or they can be very intricate and hard to predict.
- Points in a path don't need to be just corners. Middle points are interesting because guards stop there.
- Some chunks are only hard because the solution is obscure, they become very easy once the player has figured them out. Those chunks are dangerous because you can lose the player if he gets stuck for too long, and they can become a chore if good execution and timing is not important.

- It's ok to have chunks that are a piece of cake, if they are quick to beat so they don't become a chore.
- Length is perceived as a hurdle by players.
- A series of easy challenges that require proper execution can be combined to make a harder challenge as a whole.
- Avoid chunk designs that highlight bugs in the system. Make up for the bugs and weak points of the game by designing levels that don't expose those too much.
- In gas chunks, player can be killed by a surveillance camera if alert mode is activated and the doors won't let the player go through.
- Gas chunks act basically as timed chunks.
- One-tile-wide corridors are not forbidden but they can become a UX problem, handle with care.
- Red lasers with a shutdown timer set to 0 can be used as walls for the player that the guards can pass.
- Switches must be connected to the object they control via power lines in the floor.

PROCEDURAL GENERATION

Procedural generation happens when the player selects a level to play. While in the loading in screen, the generator selects the chunks to stitch together using the date of the day selected as the random seed, ensuring all players will see the same procedurally generated level for that day, every time they play. Each chunk is cloned into the level on top of the previous one.

CHUNK POOLS

The generator will use 5 different tiers of chunks to select the appropriate difficulty for a chunk, depending on the structure of the level. Each of these tiers will be subdivided by mechanic, with separate pools for Basic, Laser, Gas, Vent, and Switch chunks.

- Tier 0 (very easy difficulty)
- Tier 1 (easy difficulty)
- Tier 2 (medium difficulty)
- Tier 3 (hard difficulty)
- Tier 4 (final chunks)

CHUNK ORDER

The level generator will create the level from the bottom up, stacking one chunk on top of



the previous one, using the level structure shown in page 16. Safe chunks are fixed, while the others will be randomly selected from their corresponding tier. A Chunk will never be used more than twice in the same level, and never in consecutive chunks.

FINAL CHUNKS

The Final chunks are an exception: Tier 4 will have a hardcoded order, selecting chunks in that tier by the number of days elapsed since the last version went live.

If the number of days elapsed for this version is bigger than the number of chunks in the pool, the generator will subtract the number of chunks from the days elapsed and try with the pool for the first update, then the second, and so on until the number matches.

This way, if there are 45 final chunks for this update, and the update lasts for 45 days, each one will be unique, but on the 46th day of the update, the final chunk will be the same as the 1st day of the game, and so on, using the oldest final chunk possible as a last resort measure so the game can be active forever.

ADDING NEW CONTENT (UPDATES)

The generator may follow different rules with each version of the game, but all versions of the generator must be kept intact so the game can generate the levels for previous version and maintain consistency. Each change of generator will be signaled in the calendar menu.

ONBOARDING

SCREEN FLOW FOR FIRST-TIME USERS

The first time a user opens the game, he will experience the following sequence of events:

1. Boot-up
2. Title Screen
3. Loading
4. Tutorial Level
5. Results Screen
6. Menu

BOOT-UP, TITLE SCREEN AND LOADING SCREEN.

These will be the same as always.

TUTORIAL LEVEL

The tutorial is a pre-made level that overrides the loading of the daily level on startup, until it has been completed. It is meant to be the first level that will be played. In this tutorial, the player will play a very condensed version of the game, with only 6 chunks and 3 checkpoints. Completing the tutorial will be rewarded with the Rookie badge, which is worth 8 points, so the player will receive a custom skin right away.

RESULTS SCREEN

It will display the text 'Tutorial Complete!' instead of 'Level Clear!'

MENU

Players will be returned to the menu and they will get a prize after the prize bar is filled up with the sneaking points from the rookie badge.

RATE OUR GAME

After saving the 8th checkpoint for the first time (the tutorial is excluded) a 'RATE OUR GAME' sign will be displayed on the bottom of the screen after saving a checkpoint, until the player leaves the checkpoint zone. When clicked, a pop-up will be shown, prompting the player to visit the store page and rate the game.