STRIKE
PRODUCTIONS

- DESIGN DOCUMENT -

FOR

Intruder Alert

STEPHAN WELLS & ERIK SCERRI

GAME OVERVIEW

Design Goals

- Intruder Alert, the top-down stealth puzzle game, has a seemingly unoriginal and generic premise. Stealth games in the industry are plentiful. However, the main idea behind the game's design is to make stealth the core mechanic for other gameplay elements, rather than having it rule the entire gameplay.
- While the game is, at heart, a stealth game, puzzle elements are also a notable addition to the level design, elevating the goal of the game to more than just not getting caught.
- Good stealth mechanics come from solid AI, and good puzzle mechanics come from good level design, so attention must be put into both of those areas for this game to succeed.

Common Questions

- What is the game?
  - This game will be a cross between a stealth game and a puzzle game, with a main focus being on the stealth aspect to traverse through the levels. The traditional idea behind the stealth genre is to get from point A to point B while avoiding confrontation with the enemy. Generally, tools are given to the player to aid him/her in this goal, whether these are certain manoeuvres that the player can do (such as sneaking) or gadgets that each perform a unique function. It is also possible for some stealth games to implement a combat system, but the aim of this project is to implement gameplay that avoids confrontation.
- Why create this game?
  - The reason behind this game is to highlight the synergy between puzzle and stealth elements, and to use stealth games with patrolling guards as an approachable and suitable domain to implement path-finding AI.
- Where does the game take place?
  - It takes place in Raven HQ, the headquarters of an oil company named Raven Guard.
- What do I control?
  - The player controls an anonymous government agent.
- What is the main focus?
  - The main focus is to navigate through various top-down levels, using gadgets and good stealth mechanics, to get past the guards without being seen.
- What's different?
  - The inclusion of puzzle elements sets this apart from other stealth games.

STEPHAN WELLS & ERIK SCERRI

SAMPLING THE MARKET

Game A: Metal Gear Solid

Metal Gear Solid, a 1998 game by Konami for Sony's PlayStation 1, is quite possibly the game that pioneered the stealth genre for video games, popularising it and implementing it in 2.5D (that is, 2D with 3D elements to give a sense of depth to the game's graphics). It is a story-based, cinematic stealth game with a combat system used primarily as self-defence, contrasting it from a typical shooter game in which you have to fight your way through to make progress.

Most of the AI in the game can be found in the generic patrolling guards. These have set patrol paths and a line of vision. If the player is detected within their line of vision, then "Alert Phase" is triggered, during which guards will chase and attack the player. If the player stays out of sight for a brief period, "Evasion Phase" is triggered, and the guards will actively search for the player. Once that is over, they assume their default patrols and everything returns to normal. Game mechanics such as footsteps, noises (such as knocking on the wall), and anything suspicious can interrupt guards from their routine patrol.

There are various aspects present in Metal Gear Solid which are also present in this game. For starters, both games use a minimap (it is referred to as the soliton radar in Metal Gear), and in a similar fashion, with vision cones. Both also employ a top-down camera perspective, and, naturally, both contain patrolling guards. The main difference comes in the fact that Metal Gear Solid included a combat system with weapons (guns), whereas this game has no combat. The puzzle elements in Metal Gear weren't nearly as prominent. Needless to say, the fact that Metal Gear Solid is rendered in 3D also allows for more interesting camera angles and stealth tactics.

Game B: Abe's Oddysee

Oddworld: Abe's Oddysee is a game produced by the game company Oddworld Inhabitants and published by GT Interactive in 1997, again for the PlayStation 1, but this time for Windows and DOS too. It is another 2.5D game, but does not make use of a top-down camera perspective like Metal Gear. It is primarily a puzzle/adventure game, with heavy platforming influences and some stealth elements.

The AI in Abe's Oddysee is particularly interesting since, unlike in Metal Gear Solid, there is a large variety of different AI. The most abundant of them is a Slig, who wields a gun and shoots on sight, killing the player instantly. Scrabs are a species of animal that chase after the player upon detection, but will choose to fight their own kind over chasing the player if a choice is presented. Paramytes have more clever AI, and try and lure the player into traps or corner the player before killing him/her. The player must use stealth and clever puzzle-solving to progress through the levels in one piece.

This game was sampled not because it is in a similar genre to Intruder Alert. Far from it. However, the core concepts of stealth and avoiding combat were certainly present, and were used as reference points for this game. Concepts such as staying in a hiding spot until an enemy passes are present in both games. Getting caught in Abe's Oddysee is also a rather detrimental situation, as you will die by instantly by nearly anything that attacks you, and you have little to no means of fighting back.

GAME DEMOGRAPHICS

Target Audience

As this is a casual game, the target audience for it will be casual gamers. The term "casual gamer" encompasses a wide variety of different people, but generally, casual gamers look for games that do not require commitment or contain complex mechanics. Thus, the game will have a simple installation, limited amount of gameplay mechanics/menu options, and a very shallow learning curve that introduces the player to the game's concepts gradually.

Gender, nationality, and race are irrelevant in the target demographic, but age is considered. While the core mechanic for the stealth game is simply to avoid detection, mechanics such as items/gadgets introduce a form of complexity that may be difficult for someone of a very young age to understand. Thus, the game will be primarily suited for people around the age of ten or over.

Suffice to say, an interest towards gadgetry and stealth in general may make the game appeal more to the player. Stealth gameplay involves planning, timing and puzzle-solving, so a fondness towards such attributes will make the game more enjoyable for the player.

Persona Studies

Name: Alexis Wazowski
Gender: Male
Age: 24
Occupation: Farmer
Disabilities: Deaf
Special Notes: None

This game is suitable for Alexis because, despite his hearing loss, the game should be playable without any need for audio. The only audio that will play is the background music and occasional sound effects. Through the use of visual cues, any information that may be picked up through sound may also be picked up through sight.

Name: Aladeen Al-Farooq
Gender: Male
Age: 35
Occupation: Bank clerk
Disabilities: None
Special Notes: Runs games on a low-spec laptop

This game is suitable for Aladeen because it will not be resource-intensive to run. There will not be much happening on screen beyond some simple animations on 2D graphics, and less than around five entities (including the player) will be on screen at any one time.

Name: John Smith
Gender: Male
Age: 11
Occupation: Secondary School Student
Disabilities: None
Special Notes: None

This game is suitable for John because it has fairly simple mechanics which will appeal to someone his age that may not be interested in a deep and complex game and prefers a casual experience. The fact that it can be played level-by-level and saved between each level also makes it an easy game to pick up as a brief break between schoolwork.

FEATURE SET

General Features

- Simple and casual gameplay that can easily expand to complex brain-teasers and level layouts.
- Top-down camera perspective, for a clear view of the surroundings.
- Hand-drawn 2D graphics.
- Simple sprite animation.
- A menu with instructions on how to play and credits.
- Original music and sound effects.

Gameplay Features

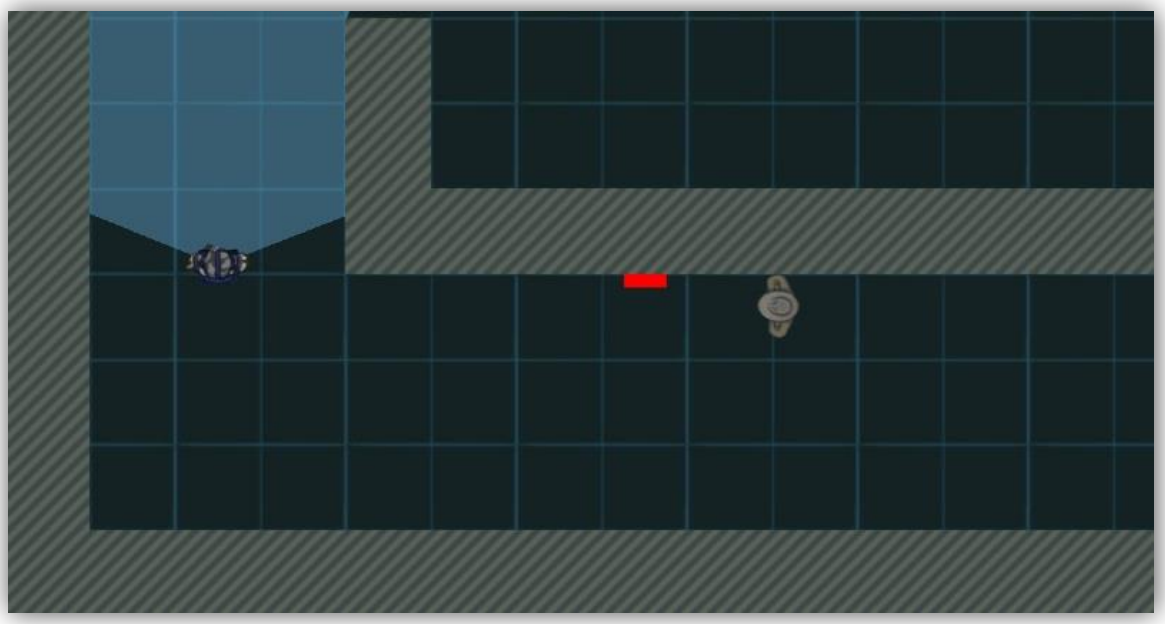
- The ability to walk (quicker) and sneak (slower).
- Minimap, showing the whole level.
- Various pick-up items to aid in solving puzzles.
- Different types of enemies.
- Multiple levels, each posing different challenges.

THE GAME WORLD

Overview

The game is set in a fairly typical headquarters called Raven HQ, which consists of a series of corridors and rooms, the former of which are often patrolled by guards, and the latter of which can be used as temporary hiding locations to wait for a patrolling guard to pass by.

A generally dark colour scheme was employed for the art, to fit with the stealth premise. A snippet can be found below.



Key Locations

Every level in the game differs from the last, but there are always some common key locations between each one. These are:

- Alarm switch:
  - This is a red button mounted on a wall, which a guard will run to if it notices you. If the guard successfully reaches the button, he will sound the alarm and the level will be lost.
- Key card location:
  - Every level has a location where the key card is stored. Only with this key card will it be possible to open the exit and finish the level.
- Exit:
  - To finish the level, the player must navigate to the exit location, which would have an automatic sliding door, which only opens up if you possess the aforementioned key card.

Travel

The player travels around the game world primarily by walking or by sneaking. On top of that, finishing a level allows the player to travel to the next level, and losing a level causes the player to spawn at the start again.

Objects

As explained earlier, there are various items which the player can pick up in the game world. These are as follows:

- Key card:
  - This item is used to open the exit door and finish the level.
- Shock trap:
  - This item is laid on the ground and any guard who walks over it will get electrocuted and stunned for an indefinite period of time.
- EMP device:
  - This item disables nearby radio equipment.

Presentation

The game is presented through a top-down camera perspective (using an orthographic camera that follows the player around via a script). This means that the viewpoint of the player will be overhead, and s/he will have a fairly expansive view of the level. One could argue that this cheapens the difficulty of the stealth element in this game, but it was decided that, due to the inclusion of puzzle elements, it would make more sense overall to use the top-down perspective. All sprites and animations thereof were rendered using Unity's default sprite renderer.

Game Engine

Intruder Alert's engine was built entirely in Unity, and scripted using C#.

The structure of the levels is primarily made up of walls and floors. Each wall is a 1x1 block and each floor is a 2x2 tile. Walls and floors are generated as multiple instances of a game object (or prefab in Unity terms). Standard Unity box collider and kinematic rigidbody components are added to the walls to make it impossible for the player or guards to walk through them, and to make it impossible for the walls to be moved.

NOTE: For DFDs and FSMs, see Appendix A.

CHARACTERS

Player

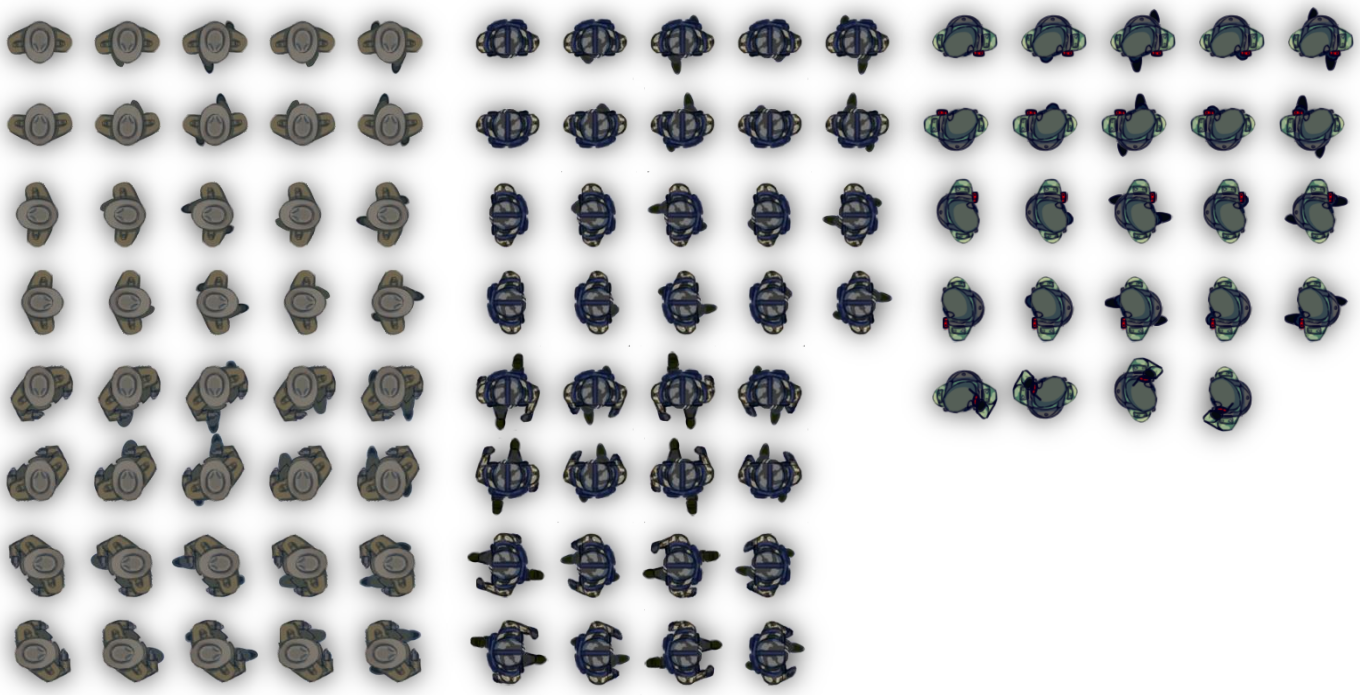
The player character is a male agent who has the ability to walk/sneak around, and use items. This character is outputted on screen and animated through the use of a sprite sheet.

Guard

Guards come in two forms:

- Basic guard:
  - This guard has a limited vision cone and patrols until he spots the player. When he does so, he runs to the alarm and activates it.
- Radio guard:
  - This guard patrols until he spots the player, and instantly calls for backup if he finds the player.

Both guard types patrol using a hard-coded array of waypoints in a loop. These waypoints correspond to locations in-game. They also both have a dynamic vision cone which gets blocked by colliders, and hence, guards cannot see the player past walls. When the player is found, the radio guard would simply sound the alarm from his radio device straight away, but the basic guard would run to the alarm switch. This is done through an adapted version of the A* algorithm. Like the player character, the guards also make use of sprite sheets.



USER INTERFACE

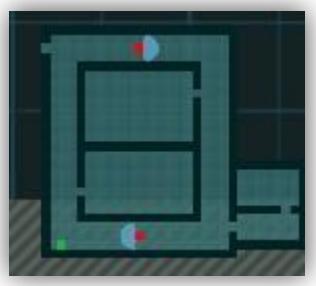
Overview

The user interface consists of two primary components: the item bar and the minimap. These are placed on the screen in an unobtrusive manner and are outputted on screen through the use of specialised cameras.

Item Bar

The item bar is used to store items picked up in the level. To use an item, the corresponding number of the item has to be pressed on the keyboard. The key-card is the only item in the game which cannot be used in this way, and is merely used when the player reaches the exit door that it corresponds to.

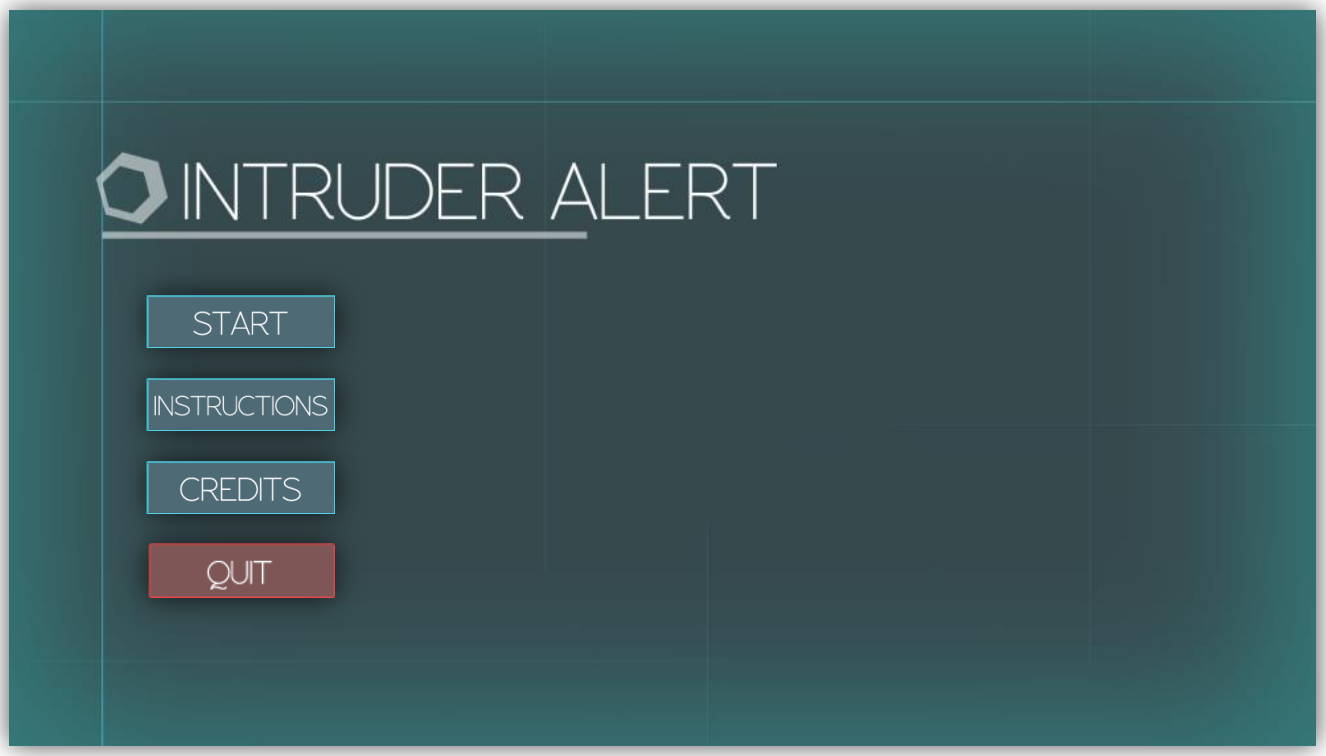
Minimap



The minimap views the layout of the whole level. The green dot symbolises the player, the red dot symbolises the enemies, and the blue shape in front of them symbolises their vision cone. The minimap was created using a specialised minimap camera which only views minimap versions of each wall/floor/character sprite. These minimap sprites are generated at the correct positions when the level first loads.

Menu

Before the game begins, a menu giving the user basic options is outputted.



MUSIC AND SFX

Music

The music chosen for the game is relatively low-key, with sections where tension rises to accentuate the inherent tension of a stealth game. It makes use of punchy drums, high, echoing bells and lots of atmospheric textures and effects.

There are two pieces of music used: the menu music and the music that plays in-game. Both are set to loop.

Sound Effects

Sound effects are a highly important part of any game, as they increase the sense of responsiveness of the game. Footsteps, alarm sounds, menu sounds, etc, have all been implemented.

SINGLE-PLAYER GAME

Overview

As the game does not implement multiplayer or any form of online functionality, there is only the single-player campaign to choose from. In this campaign, you play through a series of levels, getting from the starting point to the exit door. Every time the exit door is reached, a new level is loaded, and it is generally harder than the level before it. However, as this is ultimately a casual game, the difficulty curve will not be steep.

Campaign

The story/setting of the game is relatively simple and does not need to be known to get the full experience out of the game. The player plays as a government agent known only as Agent D, who has to investigate an oil company called Raven Guard, which is suspected for extorting citizens by sabotaging its competition's oil tankers. All of this must be done via espionage. The player has to infiltrate Raven HQ and obtain proof that justifies the aforementioned suspicion, without being found or going into combat.

The game does not take more than an hour or so to complete, as it appeals to casual players who simply want a short but fulfilling gaming experience. To win the campaign, one must successfully complete the final level and retrieve the proof.

FUTURE IMPROVEMENTS

Due to time constraints and lack of experience in this area, there were certainly quite a few aspects that could be improved on in the future. The following are some of these:

- Length of the game:
  - The first and most obvious area to work on would be the length of the game. While it is a casual game, there is no harm in expanding on the amount of levels in the game, and adding some difficult levels that may appeal to the gamers who prefer more of a challenge/brain-teaser. As long as the game's mechanics are kept relatively simple, more length would be welcomed.
- Amount of enemies:
  - For this version of the game, only two enemy types were implemented. One of them keeps patrolling normally until the player enters his vision cone, upon which he runs to the alarm and sounds it. The other also patrols until the player enters his vision cone, except when he does find the player, he sounds the alarm from his radio device. More guards which, say, investigate movement from far away, hear noises, follow footsteps, etc, could be implemented.
- Items:
  - So far, only three items are available. One of them is the keycard, which is used to get to the exit. The other is a shock trap, which can be laid on the ground and it stuns any guard who walks over it. The last is an EMP device, which disables nearby electronic equipment. There is potential for a lot more items that change the way the game is played, allowing for more interesting puzzles.
- Graphical art:
  - While the art thus far is satisfactory, due to time constraints, detail had to be sacrificed. Miscellaneous items like furniture, plant pots, lights, computers/laptops, water dispensers, fire extinguishers, etc… can add to the immersion and graphical appeal of a game, even if they cannot be interacted with in any way. On top of that, more variations of wall/floor tiles and more in-between frames for the animations are all details that were overlooked.
- Music:
  - The music, as it is now, does not loop. It simply stops and starts again, without a seamless transition. There is also a significant lack of different music to choose from. A future improvement would definitely be expanding on the soundtrack, and perhaps even implementing generative music.

CONCLUSION

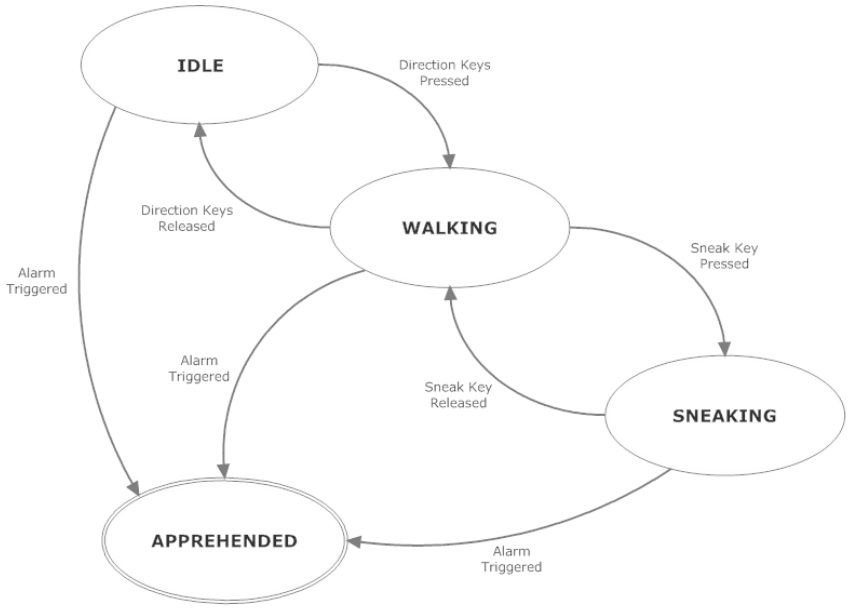
There are various improvements that can be done, but thankfully, with the game engine complete, expanding on what is already here is very much possible and feasible. Despite all that, Intruder Alert was an interesting experiment in combining elements of various genres of games (top-down games, stealth games, puzzle games) into one cohesive experience, and a good application of the fundamentals of pathfinding.

APPENDIX A - GAME ARCHITECTURE

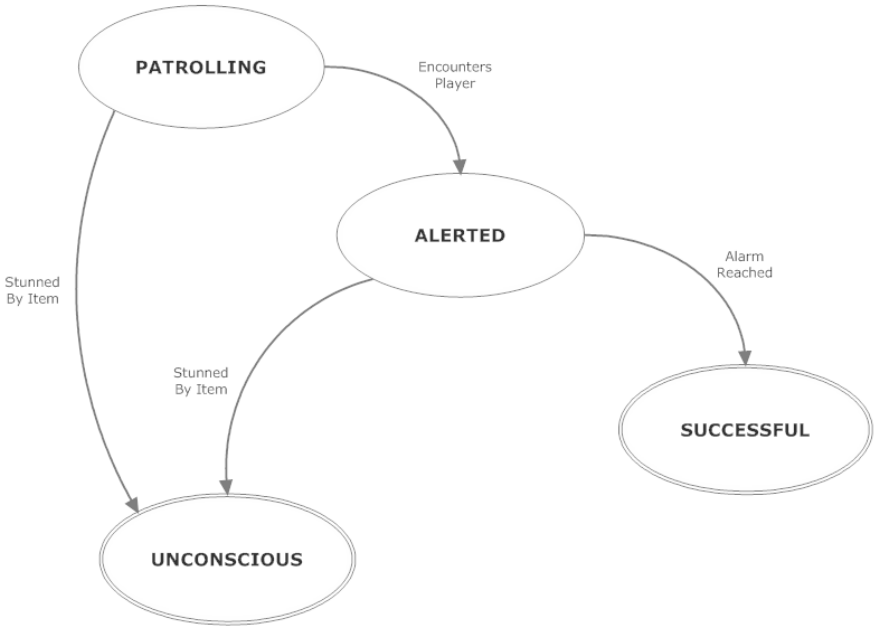
Finite State Machines

The following diagrams are a graphical representation of the different states of every entity, and how it will traverse the different states, including a basic idea of the kind of controls and conditions necessary to switch between states.

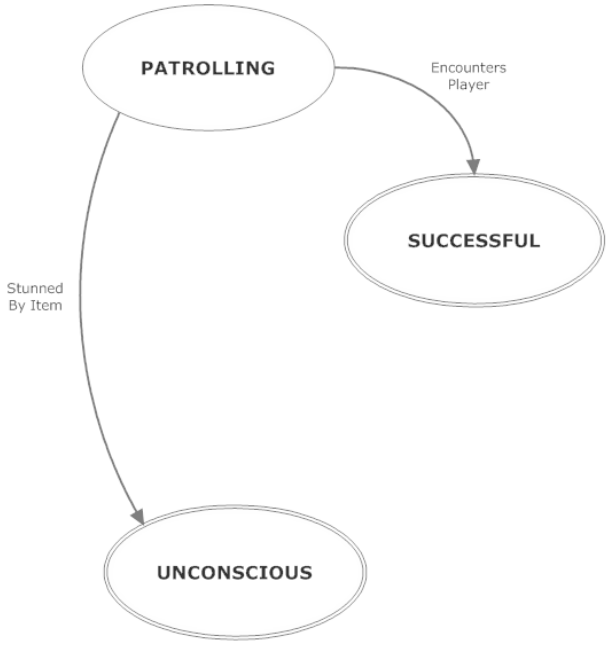
- Player:



- Basic Guard:

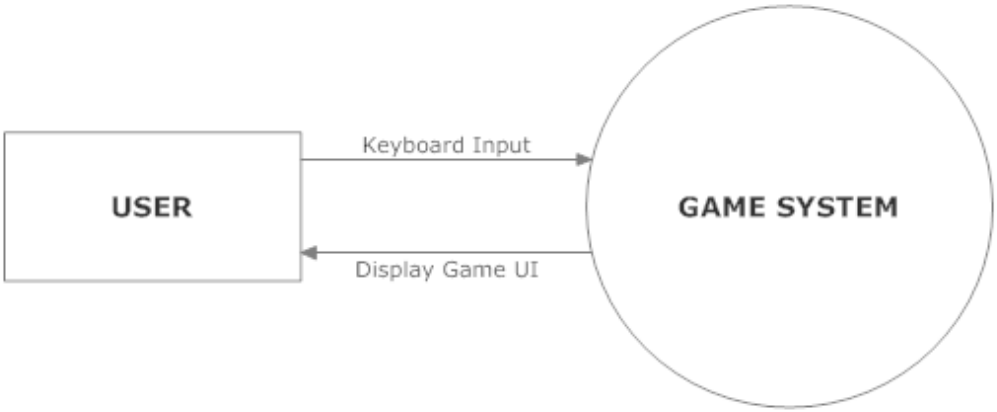


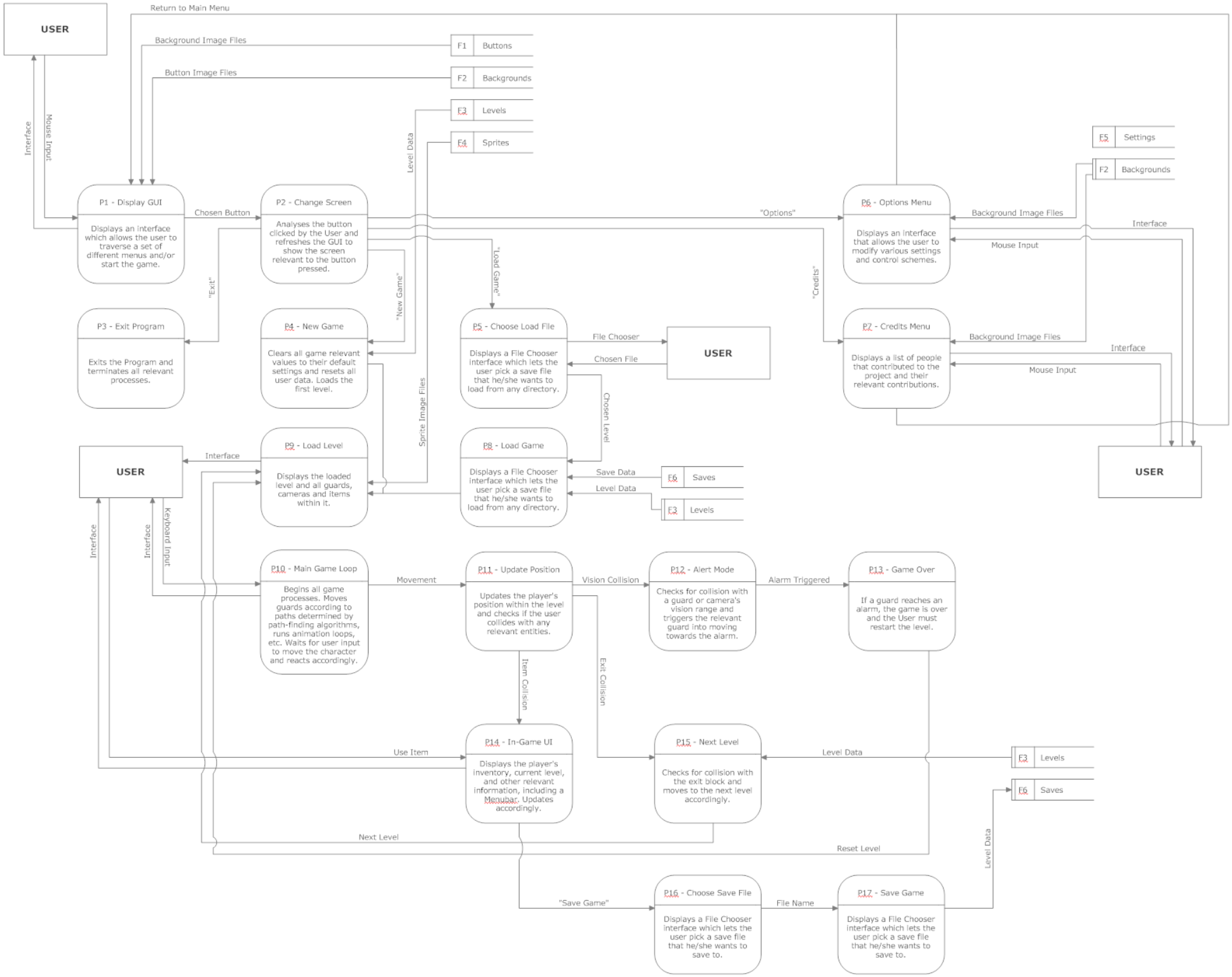
- Radio Guard:



Data Flow Diagram

The following diagrams are a graphical representation of how data flows through the system, including a simple conceptual model of input and output streams, and the transition of data from one process to another. The first diagram is a Level 0 DFD, showing the basic outline of the system. The second diagram is a Level 1 DFD, going further into depth regards the different procedures inside the Game System.




USER
Return to Main Menu
Background Image Files
Button Image Files
F1 Buttons
F2 Backgrounds
F3 Levels
F4 Sprites
F5 Settings
Interface
Mouse Input
Level Data
P1 - Display GUI
Displays an interface which allows the user to traverse a set of different menus and/or start the game.
Chosen Button
P2 - Change Screen
Analyses the button clicked by the User and refreshes the GUI to show the screen relevant to the button pressed.
"Options"
P6 - Options Menu
Displays an interface that allows the user to modify various settings and control schemes.
"Exit"
"New Game"
"Load Game"
"Credits"
P3 - Exit Program
Exits the Program and terminates all relevant processes.
P4 - New Game
Clears all game relevant values to their default settings and resets all user data. Loads the first level.
P5 - Choose Load File
Displays a File Chooser interface which lets the user pick a save file that he/she wants to load from any directory.
File Chooser
Chosen File
P7 - Credits Menu
Displays a list of people that contributed to the project and their relevant contributions.
Sprite Image Files
Chosen Level
P9 - Load Level
Displays the loaded level and all guards, cameras and items within it.
P8 - Load Game
Displays a File Chooser interface which lets the user pick a save file that he/she wants to load from any directory.
Save Data
Level Data
F6 Saves
Keyboard Input
P10 - Main Game Loop
Begins all game processes. Moves guards according to paths determined by path-finding algorithms, runs animation loops, etc. Waits for user input to move the character and reacts accordingly.
Movement
P11 - Update Position
Updates the player's position within the level and checks if the user collides with any relevant entities.
Vision Collision
P12 - Alert Mode
Checks for collision with a guard or camera's vision range and triggers the relevant guard into moving towards the alarm.
Alarm Triggered
P13 - Game Over
If a guard reaches an alarm, the game is over and the User must restart the level.
Item Collision
Exit Collision
P14 - In-Game UI
Displays the player's inventory, current level, and other relevant information, including a Menubar. Updates accordingly.
Use Item
P15 - Next Level
Checks for collision with the exit block and moves to the next level accordingly.
Next Level
Reset Level
"Save Game"
P16 - Choose Save File
Displays a File Chooser interface which lets the user pick a save file that he/she wants to save to.
File Name
P17 - Save Game
Displays a File Chooser interface which lets the user pick a save file that he/she wants to save to.