Game Dev Sim

Cat Burglar

Technical Design Document

# Basic Overview

**Genre and Theme:**

Cat Burglar is a stealth game with some strategy elements in the way the player approaches how to most effectively move about the game without being caught while still trying to steal as many items as possible to gain the highest possible score.

**Overview of Play**

The player will be tasked with trying to steal as many items of value as possible from the house they are in, to increase their score. The player will need to be quiet and use stealth to sneak about the house without waking up the residents. As the player collects more items, they will begin to make more sound as they move, which will increase the likelihood of alerting the residents of the house. If they player wakes the residents of the house, they will be chased by them and if they are caught they will fail the mission and lose the game.

**Overview of key mechanics and systems:**

Movement – a basic character controller will be used allowing the player to move up, down, left and right. All controls will be relative to the screen.

Steal – The player will be able to pick up items as they steal them, and the object will be then destroyed from the hierarchy and removed from the scene. So, in the game it will look like the item has disappeared.

Glowing Item – Items of value that the player can steal will have a glow or sparkle affect.

Score – As the player steals items they will be awarded a score based on the value of the item in dollars.

Noise Detection Area – The player will have a ring around them to indicate the amount of noise they are making and how far and the size of the ring indicates how far from them the noise can be heard. This ring will be increased depending on the player’s score.

Enemy AI – If the player creates too much noise they will wake up the resident/s. This will be triggered by the resident being inside the players noise detection area for an extended period (time frame to be calculated later). If the residents are awoken they will begin chasing the player if they are inside the players noise ring, if they catch the player it will end the game and the player loses.

Day/Night Cycle – This will indicate the amount of time the player has before time runs out. Once time runs out, all the enemy AI will wake up.

**Overview of Camera and Controls**

* The camera won’t have any controls and will be fixed in place as our game will take place on a single screen.
* The character controls will all be relative to the screen, as follows:
  + Player 1
    - W = Walk UP
    - S = Walk Down
    - A = Walk Left
    - D = Walk Right
    - Left Ctrl = Pick up items
  + Player 2
    - ↑ = Walk Up
    - ↓ = Walk Down
    - ← = Walk Left
    - → = Walk Right
    - Right Ctrl = Pick up items

# Development Environment

**Game Engine**

The Game Dev Sim team will be using the Unity game engine version 2017.3.0f3 to execute the production and build of the game Cat Burglar.

**Target Platform**

The target platform that the Game Dev Sim team will be deploying the game Cat Burglar to is PC.

**Source Control**

GitHub is the chosen method of source control being used by the Game Dev Sim team for Cat Burglar.

**Asset importation & implementation pipeline**

Each of the art assets that are required for Cat Burglar will be exported from Maya (or other software program) and then imported into Unity by the artist who creates it.

The music will be made by an external source and the sound effects will be recorded by an external source, they will then be either emailed or digitally transferred (depending on the size of the file) to the Game Dev Sim team and then they will be imported into Unity by the designer. Sound files may need to be converted from MP3 to WAV files using an online converter which will also be the responsibility of the designer.

# Technical Game System Breakdown

**A technical breakdown of the core gameplay**

Character Movement – Player can move in 4 directions relative to the screen which will not move.

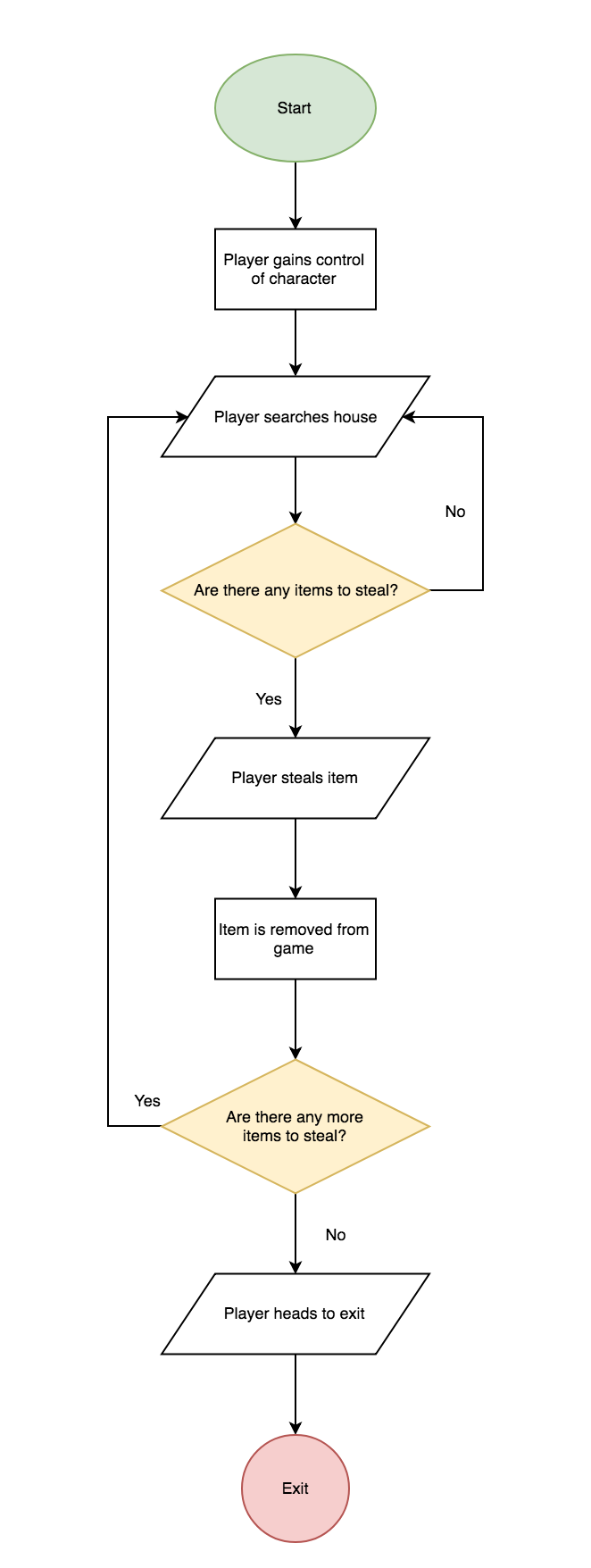
Pickups – Player can grab/take item by pressing the CTRL button. The item will be destroyed from the scene, and its value will be added to the players score.

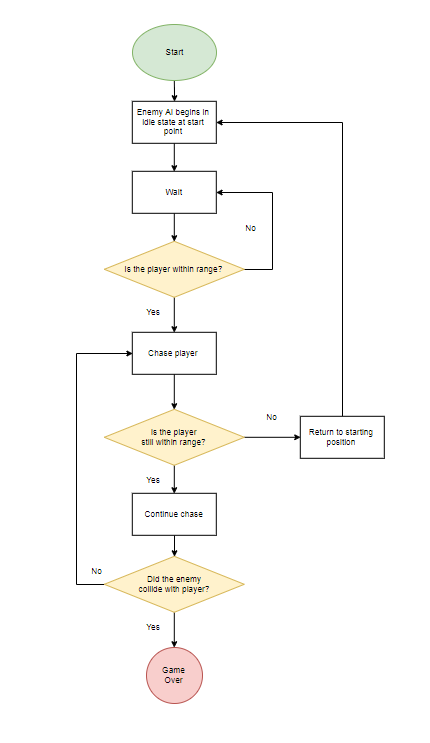
Scoring – Players score will increase as they steal items. The items will have a fixed score value attached to them. This score will increase the size of the noise ring, through a simple addition.

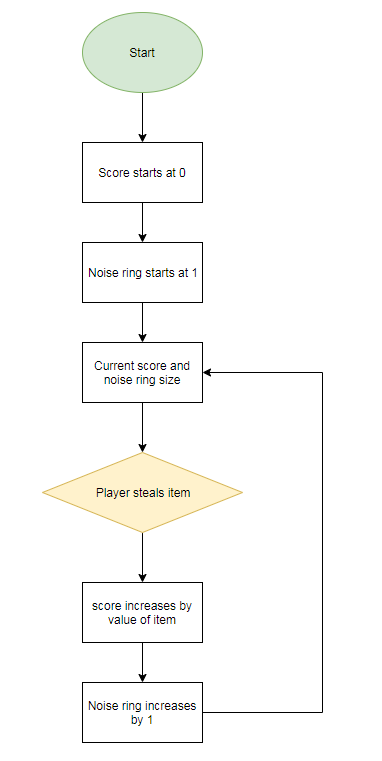
Noise Ring – A ring will appear around the player to indicate the area in which they can be detected by the resident/s (or dog) of the house. If the resident is inside this ring for an extended period of time, the resident will be alerted to the player and wake up and begin searching for the player. The enemy AI’s radius will share the same values with the noise ring, so that it is easier to display the noise radius of the player

Enemy AI – If the player creates too much noise they will wake up the resident/s who will begin to search for the player and then chase them. A check will be made, to make sure the player is still within the radius of the AI. Once the AI is out of the player’s noise ring, the AI will path back to its starting point.

Day/Night Cycle – The inclusion of daylight coming through the windows in the morning to indicate the player is about to run out of time. Once time runs out, all the AI will have their detection radius increased to the size of the level, and the noise ring of the player will be increased to the size of the level. This causes the AI to constantly chase the player until either, the player escapes, or the player gets caught.

**Diagrams (possibly UML style) outlining how game components interact**

****

****

**Detailed information on any specific systems that need to be built on top of what the game engine provides**

Cat Burglar is a single screen game, this is to remove the need for having the camera track the player on the screen. Due to this design, the players controls will function relative to the screen allowing for movement in the up, down, left and right directions.

The character controller we are using we have built so that we can have full control over all its functions, as our controls are minimal we do not want any unnecessary actions available to the player as we feel it will only distract or confuse the player. The character controller is set up in a way so that prevents the player from walking through walls but can collide with any trigger boxes we made use such as the items the player can steal.

The AI system we are using calls a function to check its own position relative to the players position. If the player is within a certain range, the enemy AI will begin chasing the player. We will also be using a NavMesh for the AI, to make sure that they do not path through walls, and to allow the AI to path back to its original position after the player is out of range. If the AI collides with the player, the game will end, showing a game over screen. If it is a 2-player game, the player that collided with the enemy AI will be taken out of the game, and the other player can still play, until either they escape, or they get caught, which in this case, there will be a game over screen.

The scoring system we are using will be able to handle the input of various scores based on the item the player steals. The scoring system will correlate to the approximate real-world value of the item so the player can attempt to figure out what is of a higher value simply by looking at the items. This will be achieved by giving different values to all the items the player can steal. The score will be set as a dollar value to indicate the wealth the player has accumulated.