Game Design Document

By:

Alejandro Flores Lerdo de Tejada (40174159)

Danny Roux-Dufault (40112027)

Paul Grippa Vento (40174007)

Roxane Morin (40191881)

Sami Ibrahim (40156134)

Shiraz Siwadjian (40136555)

Course:

COMP376: Introduction to game development

Fall 2022

Presented To:

Kaustubha Mendhurwar

Table of Contents

Executive Summary

What is 'Serial Cleaners'?

"Clean and don't get caught!"

Serial Cleaners is a local 3D multiplayer game, where two players have to work together to clean a crime scene left behind by a serial killer. Their goal is to leave the crime scene squeaky clean in the shortest time possible before the next residents arrive. The players must work in a team, using their precision skills and puzzle-solving abilities, to complete multiple levels and get the highest score.

Genre

The Genre of the game can be classified as a party game and puzzle game. It contains cartoon elements and assets while still appealing to horror enthusiasts.

Setting

The game contains multiple levels, each in a different setting such as a home, an industrial space and a cabin. After completing a level, the players can unlock other levels to start cleaning more challenging crime scenes. Using custom-made assets, the setting of each location stands out and gives a better look to the game we intended to make. The setting makes the game more fun and enjoyable with its cartoon characters and assets while still containing some gory elements of a crime scene.

Target Audience

Serial Cleaners is a game that will appeal to young and older adults. Specifically, people who enjoy completing task-based games and puzzle strategy games. It should also appeal to players who like playing with friends and helping each other to succeed.

Platform

The game is intended to be played on a PC.

Selling Points

Some of the key selling points of the game include the cartoony look and feel to appeal to a greater audience, visually pleasing assets, custom blood-cleaning mechanics, interactive puzzles and multiple cooperative challenging levels.

Overview

Major Game Qualities

The major game qualities include co-op gameplay, custom cleaning mechanics, and level designs, which are used in combination to provide a unique and entertaining experience to the players.

Co-op game (Teamwork and communication)

Serial Cleaners incorporates many unique game qualities that make it stand out from other games. The main goal of the game is quite simple and intuitive, yet it offers a challenging experience for the players by forcing them to communicate and work in a team. Many of the puzzles and objectives require both players to work together to move objects around or to help each other in specific ways to complete the required challenges. It encourages the players to establish good communication with one another to allow them to succeed.

Additionally, the levels have a timer that puts a time pressure element on the players, meaning they must work together efficiently to avoid wasting time and complete all the required tasks. Otherwise, if they run out of time and have not yet completed all of the challenges, then they will lose. Also, based on how fast the players complete the levels, they will be awarded more or fewer stars accordingly.

Custom blood cleaning and physics mechanics

The concept of the game relies on cleaning up a crime scene and re-organizing the level assets. To achieve those goals, we decided to create a custom cleaning mechanic, where the players can pick up mops and other cleaning supplies to clean the blood splatters and messes left behind by the killer. The cleaning mechanic helps set the tone of the game and introduces a fun and visually appealing aspect where the player gets immediate visual feedback on how they are cleaning, allowing for very interactive gameplay.

Alongside the blood cleaning, having 2 players work together gives us more freedom and flexibility to incorporate the cleaning mechanic and other cleaning challenges that require teamwork to complete. For example, a level might contain body bags that need to be moved out of the space. The bodies can require both players to grab each end and navigate through tight spaces in order to dump it somewhere else. Another example is if the players need to move a couch, they have to work together to move it and place it elsewhere. These assets can have different weights to them, slowing down their movements.

Level Designs

The level designs are a crucial aspect of the game. Our game will contain a tutorial level, made to allow the players to get comfortable with the movements and basic mechanics needed. The rest of the levels will get increasingly difficult and will introduce more mechanics as the game progresses. The level designs are part of what makes the game easy or difficult. Harder levels will contain more narrow spaces, harder cleaning tasks and less time, while easier ones will have a generous time frame, more open spaces and easier cleaning tasks.

To make the levels more appealing and consistent, we have decided to custom fabricate our own assets. Although it will require more time and effort rather than using existing assets, it will allow us more freedom and creativity to design our models and incorporate them into our game.

Related Games and Inspiration

We drew inspiration from a few games that we all liked and wanted to create something similar. Moving out, Overcooked and Plateup! were among a few games we looked at.

Moving out

- **Publisher:** Studio DevM
- **Genre/Platform:** Puzzle, action / PC, Consoles
- **Year:** 2020

Players in Moving Out take on the role of movers, moving marked furniture and appliances (such as couches, refrigerators, and microwaves) from a house to a moving truck under a time limit. Along the way, obstacles (such as rakes, fires, ice, and even ghosts) may be encountered. Some heavy objects require two people to move, while other objects are fragile and may be easily broken. Objects may be thrown. Players are ranked on a bronze, silver and gold scale, based on how quickly all of the objects are packed into the moving truck. The levels also have optional objectives, such as breaking all of the house's windows or packing an unmarked object. Additional bonus levels may also be unlocked.

Moving out was a big inspiration for our game and we have decided to incorporate many elements from it into our game with more twists and features. For example, our game contains corpses that need to be disposed of, which requires moving them, similar to how furniture is moved in Moving out. Another similarity is time control. We decided to add a timer to the levels to keep track of the player’s speed and give stars based on their performance.

Image of gameplay from Moving Out

Overcooked 2

- **Publisher:** Team17 / The Label
- **Genre/Platform:** Simulation, Party / Consoles
- **Year:** 2018

Overcooked 2 is a cooperative cooking simulation video game. In the cooking simulator game Overcooked 2, teams of up to four players cooperatively prepare and cook orders in absurd restaurants. Players gather, chop, and cook ingredients, combine them on plates, serving dishes, and wash dishes. Some levels have moving floors and other obstacles that complicate the cooking process, including portals, moving walkways, and impassable fires. Other levels transition between settings and recipes, such as one that begins with preparing salads in a hot air balloon and ends crashlanded in a sushi kitchen.

We wanted to have a similar co-op team-based gameplay in our game and decided to have 2 players working together in pairs to complete the cleaning tasks, similar to how the players prepare and serve food in Overcooked 2. Having the players work together to clean the level makes it more challenging and enjoyable to play, while allowing us, to create better puzzles and challenges for the players.

Overcooked 2: Time to head back to the kitchen

Plateup!

- **Publisher:** Yogscast Games
- **Genre/Platform:** Management, Rougelite / PC, Consoles
- **Year:** 2022

Plateup! is a small indie game that focuses on managing a store. In the game, there is a time limit per level increasing the intensity of the choices the player makes in order to complete every task as fast as possible.

The same idea applies to our game. One of the tasks that are shared is cleaning the dirty parts, in our case, it will range from blood stains and other fluids on different surfaces, compared to Plateup! which is mostly dirty floors with food.

Plateup! Indie Video Game

Player Composites

Sasha Shakur

Age: 15-20

Job Status: employed

Education: working in a Falafel shop.

Sasha plays games at home after a long day at work, so between 8:00 pm and in the evening. She also plays on her breaks at work using her boss' laptop. She gets her games from her boyfriend, on her birthday or at Black friday. Sasha likes to play on the laptop she brings to school, so only plays card games or low requirement games. She also plays AAA games on the Playstation5 she has at home from time to time. If she plays on the laptop, she typically plays 2 hours on the dinner table, or 40 mins on breaks at school. When playing on the Playstation5 however she can easily play 3 hours. Sasha plays day in day out, and more time on the holidays. She likes to play multiplayer games, as they provide quality time with her friends and her 2 years old sister.escaping reality is the goal she has in mind when she plays a game. Sasha likes collaborative games and originality. She really enjoys eating sugary food, and simple pleasures like sleeping 11 hours every day. Sasha does not have a boyfriend that brings her gifts but her parents buy her gifts from time to time.

Luna Smith

Age: 33-40

Job status: belly dancer, influencer, “content creator”.

Education: Went to college and got a Degree.

Luna dances on Tiktok for a living. To inspire her work, she plays lots of games that have a certain aesthetic feel that is interesting to her, such as guitar hero and dancing monkey. She plays games at her ex-husband's house on the weekends, and practices dancing while playing them. She buys her own games by saving tips from her work . She plays on the PC that she uses to edit her Tiktok videos as well. Luna spends 2 hours gaming every day on average, and plays with her best friend's husband in order to practice on dances that require two people. She loves how playing a game can be funny and healthy at the same time. She also likes games that are very relaxed, because she is already stressed enough from work. Luna loves Tim Hortons coffee in the morning and she really enjoys watching illegal dog fights. She spends about 9000$ per month on treating herself, buying luxury clothes, gambling and hunting wild animals. When she isn’t gaming, she’s dancing and vice versa.

World

The game takes place in a world where serial killers, supernatural scourges, and mad scientists not only exist, but call upon the services of the mysterious Serial Cleaners. This shadowy enterprise sends its agents to clean up the crimes of those who would not want to be discovered, thus protecting the evil doers of the world.

The public is not aware of the Serial Cleaners nor of the monsters and maniacs they trail behind, only of the victims who mysteriously disappeared. Law enforcement agencies fare little better, finding only pristine order in the place of expected crime scenes. Among those in the known - the criminal, cultists and creatures who call upon them- the Serial Cleaners are both trusted, honored… and feared. Even to them, the reach and inner workings of the organization remain a mystery. Is it merely the enterprise of some rich and repugnant mortal, or does its corporate aesthetic hide something altogether more eldritch? The Serial Cleaners are perhaps less human than they seem…

Characters

Lore

Who are the agents of the Serial Cleaners? How were they induced into the organization, and for how long have they been working?

Some might once have been ordinary folk who fell on hard times, followed dark paths or sold their souls. Others were base cultists and murderers who found a higher calling in assisting this world's evildoers. Others still were never human, but supernatural entities with their own mysterious agendas. In the end, it does not matter who these people are, only that they are here to do a job and do it well.

Originally, Serial Cleaners' characters were meant to be pledges to a demonic cult who had to prove themselves through lowly menial tasks. But our style and lore changed as the project progressed. The Cleaners took on an aesthetic inspired by archetypical 'Men in Black', and the characters became more and more anonymous.

Visuals

Our initial placeholder models were realistically proportioned humans wearing red and yellow hazmat suits. We decided to create our own character model still wearing a hazmat suit, though its proportions are more stylized. However, the suit's colour is now randomized upon starting the game. This contributes to the characters' anonymity by implying to the players that they are taking the role of different agents in each play session.

Progression Chart

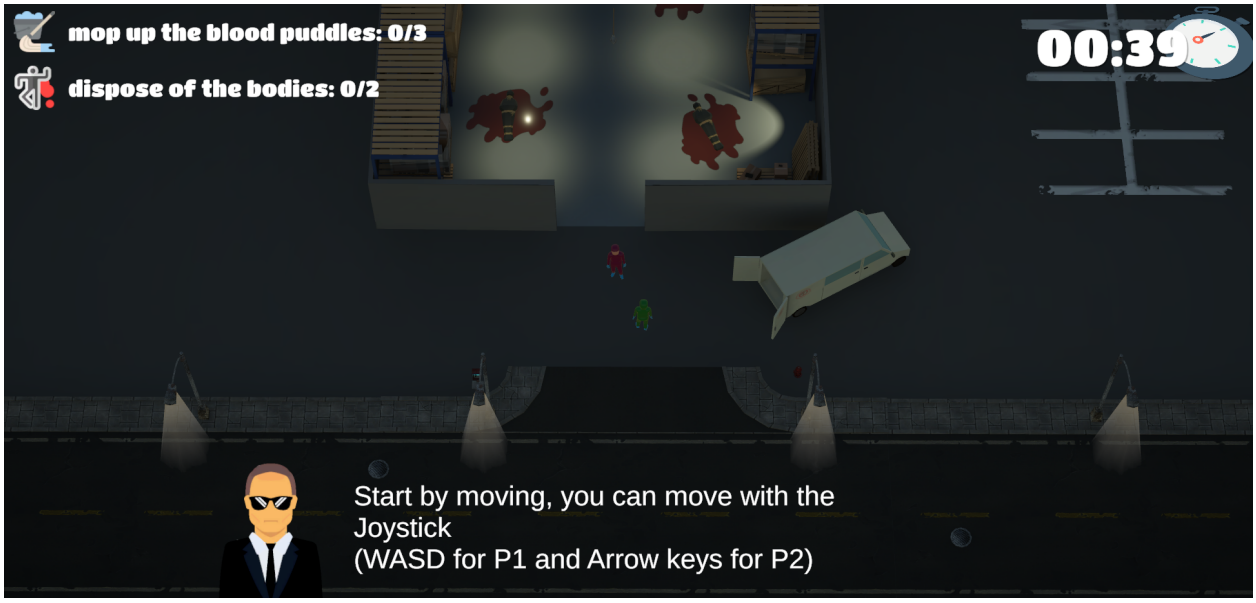
Art Direction

We originally had in mind a bloody but cutesy aesthetic with models in the style of Animal Crossing. In practice, it proved easier to find and produce assets in a low poly, colorblocked style. We ended up using mostly low poly assets, which were easier to find and more had more abundant options that we liked. However, we still tried to keep the cute cartoony aspect of the game and also kept a bit of the horror aspect.

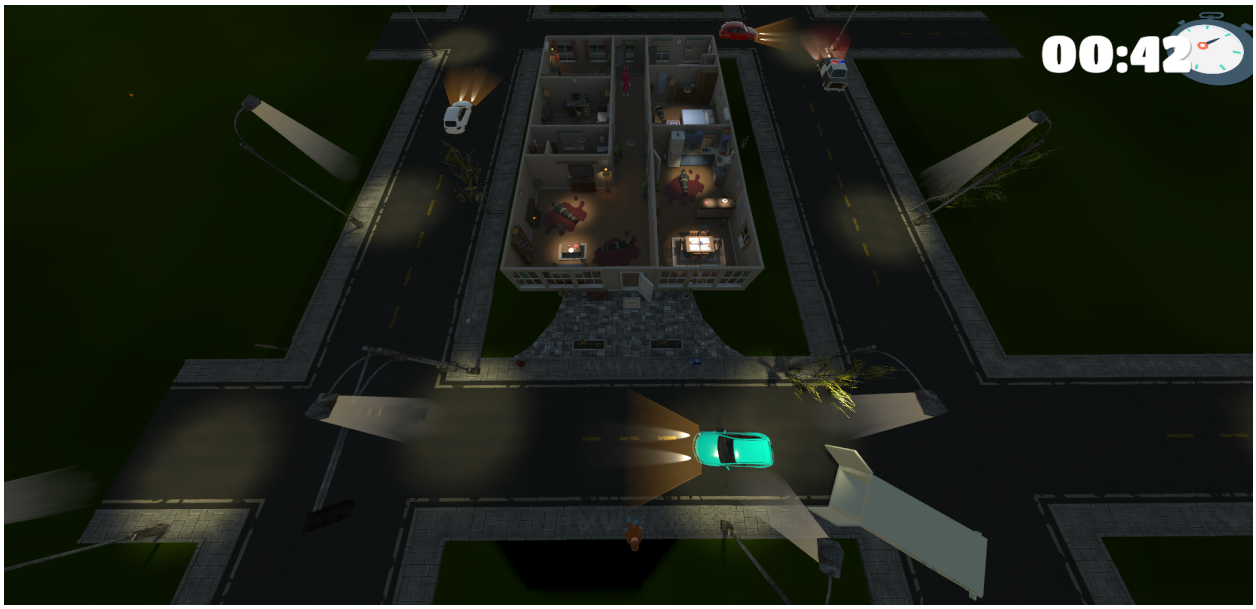
Much of the game's ambiance is communicated through lighting. We aimed for lighting that was dramatic and gloomy but fit diegetically in the world. Thus the house's lamps are dim but relatively warm. The hospital's lamps glow an ashen shade of green, and the spotlights of exit signs and medical equipment accent the level. The industrial laboratory is illuminated by powerful yellow ceiling lamps and an eerie red ambient glow.

Levels

Tutorial



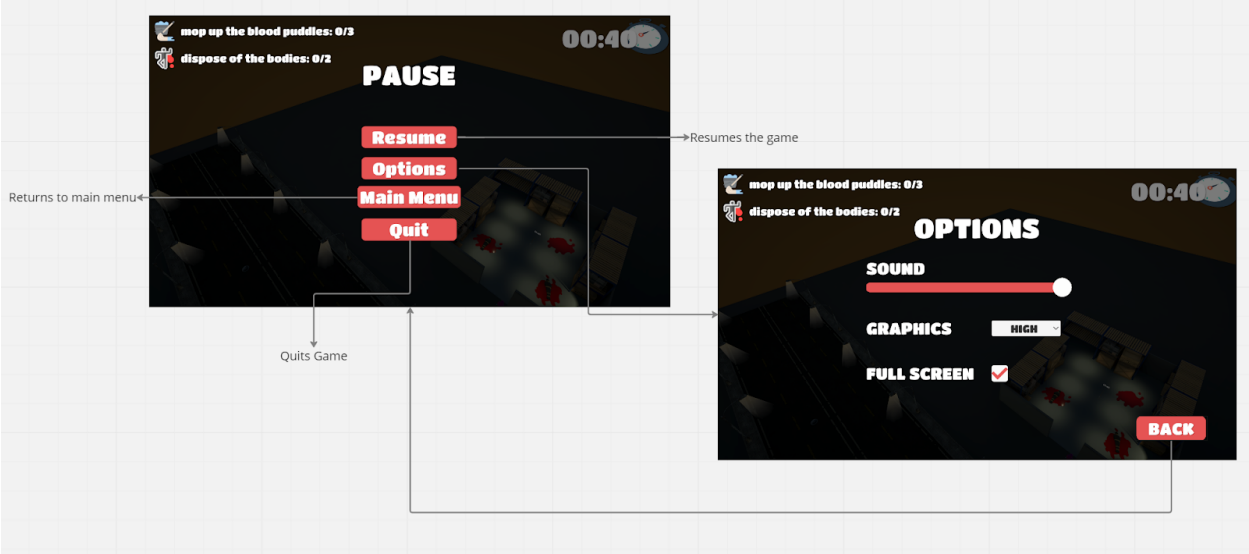
Level 1



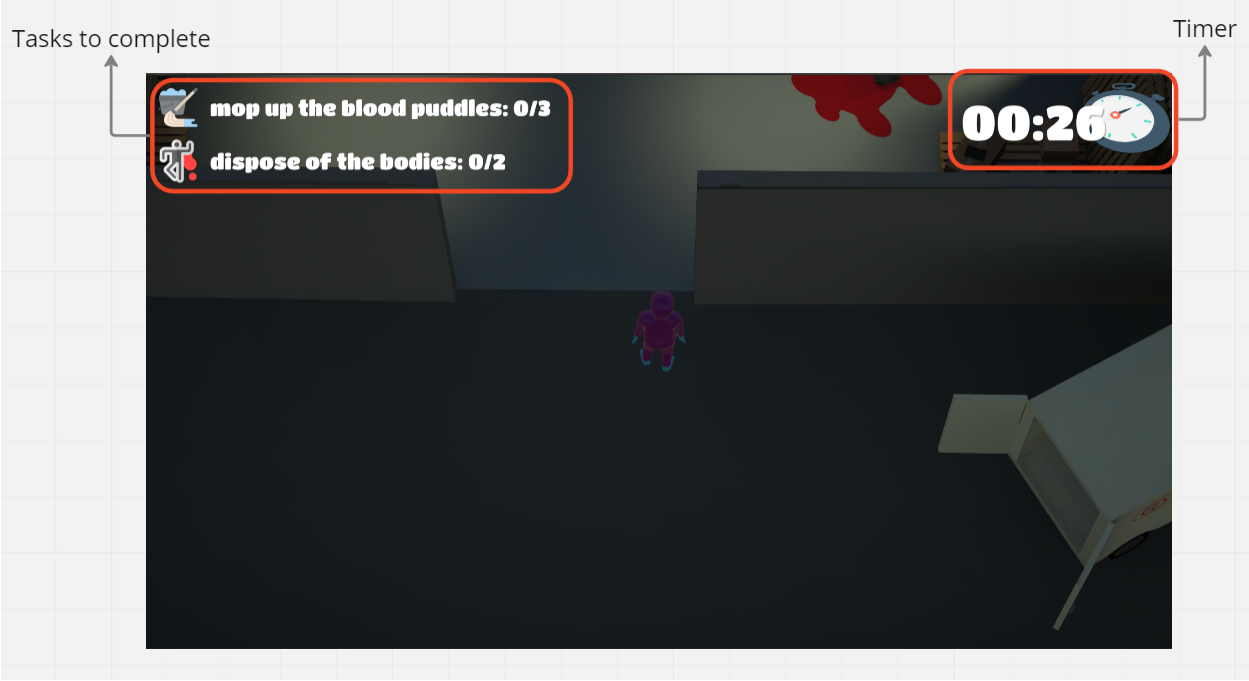
UI Storyboard

Main Menu

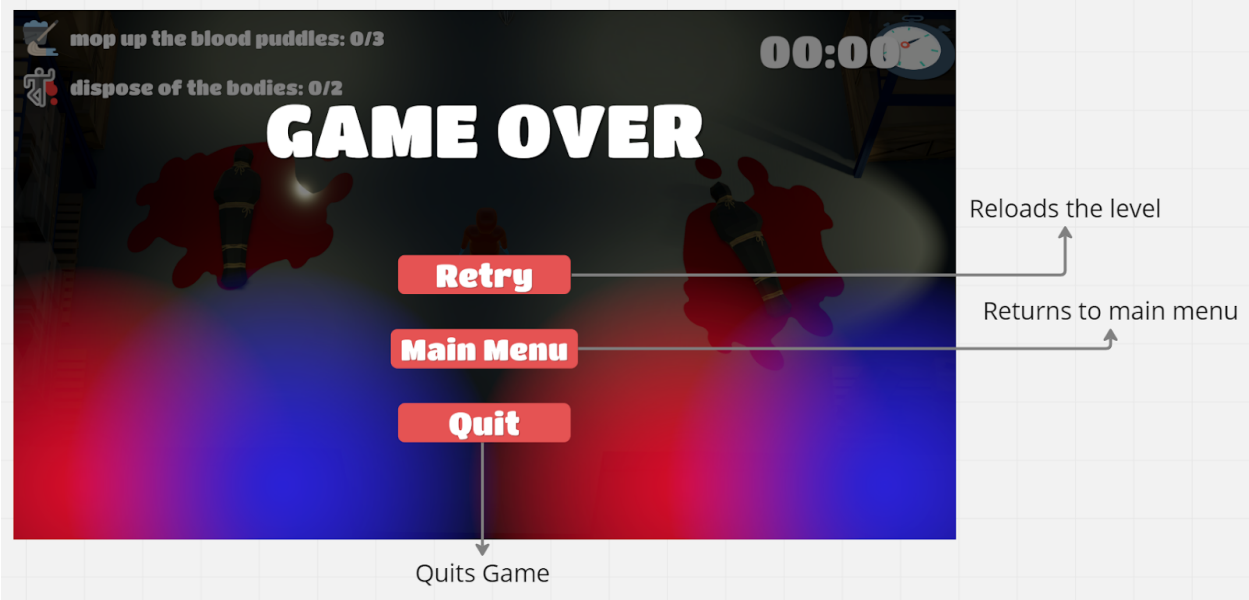
Pause Menu



Main HUD



On Game Over


mop up the blood puddles: 0/3
dispose of the bodies: 0/2
00:00
GAME OVER
Retry
Main Menu
Quit
Reloads the level
Returns to main menu
Quits Game

On End Level


Time remaining
99:99
LEVEL COMPLETED!
TIME REMAINING:
00:00
Number of Stars for the level
Main Menu
Quit
Returns to main menu
Quits Game

Systems

Serial Cleaners' level and task systems were designed to be modular and easy to extend upon. Since having multiple levels and task types were some of our main goals, it proved important to establish a good infrastructure for them. In practice, said infrastructure consists of a few different subsystems: 'global' and 'local' level managers and databases, as well as different types of tasks and their managers.

Level Management System

The level management system is composed of three main components: level parameters, a 'local' level manager, and a global one.

Level parameters are data objects making use of Unity's scriptable object architecture. They are used to keep track of each level's basic information: its name and ID number, its default duration, and a list of the tasks found in the level as well as the number of times each must be completed. This information will be referenced and used by the subsequent systems.

The 'local' level manager (simply known internally as the "LevelManager" class) is a singleton object which exists once in each of the game's levels. It references the current level's parameter data and handles both its initialization and runtime. Its first action is to create the various managers associated with the tasks listed in its parameters and to link them to the relevant scene objects and UI components. Its main runtime purpose is then to handle the level's timer, which is decreased by a coroutine. If it runs out before the players can complete all the tasks found in the level, the level manager calls a failure event. And every time an individual task is completed, the manager verifies whether all others are also marked as complete. If that is the case, it is called a victory event; the players may move on to the next level. The level manager also controls the ability to pause the game and hosts the pause and option menu screens.

Global Level Management System

The global level manager is responsible for keeping track of the progression of each level in a global scope. On Awake, it creates a static dictionary and indexes each level with an integer to a Level Info class. The Level Info class holds the basic data for each level. The component is attached to an object and uses a singleton design pattern to prevent any other Global Level Manager from instantiating. As well as the object was not being destroyed on load so it kept after loading another scene.

When finishing a level, the Level Manager calls a static function in the Global Level Manager class, using the index of the current level, it uses it to get the level info of the index in the dictionary and marks the level as completed, gets the score and time remaining and stores all of it into the level info class.

When the player goes back to the main menu, the main menu manager checks with the Global level manager which level is completed and its corresponding score and time updating the UI accordingly.

Task System

The task management system is assembled from similar components. Each type of task possesses its own data type, specialized manager and interactive game object(s).

Like the level parameters before them, task data is represented by scriptable objects. Each of these contains the textual description and icon reference used by the task list UI element, as well as a string containing the class name of its manager.

Every task possesses a dedicated manager class. But though each has its specialties, they all inherit from a parent TaskManager class and share the same underlying architecture. Informed by the current level's parameters, a task manager tracks how many instances of the said task must be accomplished, and the number of those which have been completed. It will be notified of each instance's completion and, in turn, update the relevant counter in the task list UI. And, as needed, will inform other scripts and systems of the task's current completion status. It also keeps a list of the game objects relevant to its task, though this list is only used locally. Certain task managers, such as the blood puddles', also hold additional information referenced by the associated task objects.

Said task objects are varied: blood puddles and the mop and product to clean them, items of furniture and their target positions, bodybags, incriminating evidence and their dispatch zone. Each has its own kind of MonoBehaviour, though all carriable objects share an 'Interactable' parent class. These scripts handle the tasks' actual interactivity: moving and destroying items, updating their visuals, notifying the relevant managers, etc. The blood puddles are more complex than most as they not only have to be 'cleanable' but also dry over time and conditionally require a different kind of interaction.

Blood Puddles

Part of the Task to complete, the blood puddles are their own system. Using a shader to create a mask material for the shape of the shader and then accessing through code. Creating a template mask from the actual base mask texture, I set that texture to the material using the shader. Then check if the mop is inside the boundaries of the plane that holds the material, if it is, it creates a raycast and the hit point will loop through the texture and apply to multiply the brush texture to the mask texture, therefore each point that is black will be multiplied to the texture making go transparent as the mask uses black as the alpha.

Another feature of the puddles, to make them more difficult, is that they dry over time. When they are instantiated they start a timer that will make them go darker and when they are fully dried they are no longer cleaned with the mop. To clean them, the player will need

now to get the cleaning agent, wets the mop and now the mop will be able to clean the puddles again for 10 seconds (atomic parameter).

Cars and Police

We made a car patrolling system using a NavMesh that patrols on the roads. There are two different entities: the police car and the civilian cars.

The police car patrols around the map, faster than the other civilian cars and has a wider field of view, to detect players in it's path. The fieldviewer finds players using the Physics.OverlapSphere method, who are in a given radius from the car and inside the view angle of the car. If those conditions are true, the police lights start flashing and a police siren rings. The players lose the game and must restart. The goal of the AI police car is to add an extra challenge where the players must wait for the police car to pass in order to dump the bodies in the van, which is strategically placed at the opposite of the road to force the players to wait. If the police spots any player, a game over menu appears, showing that they have lost and must restart.

Similar to the police car, the civilian cars patrol around the map and also have a field of view that can detect players. The only difference is, that if the civilian cars spot a player, they stop and start honking their horns and wait for the player to move. Additionally, the player loses 3 seconds from their time and the timer turns red to show that they have lost time.

Additional Retrospective

The level and task managers could well have been adapted to spawn the various task objects at semi-random locations in the levels. Due to the lack of foresight and time, such a feature was not implemented. Thus the various task objects had to manually be placed in each scene. This represents a flaw in our systems as it can result in a mismatch between the number of task instances expected by a level's scriptable object, and that of the actual task interactable found in its scene. This could result in levels that cannot be won as, for example, the level manager could expect three bodies to be disposed of when only two exist in the scene.

In the end, most of the existing tasks share the same basic logic: pick up an object, and carry it to a specific area. Though we would have liked to implement more diverse tasks, some of our initial ideas proved too difficult or time-consuming for the scope of the project. The two-person carry of bodies and/or furniture was such a victim. Other non-implemented feature ideas include corpses that would've left a trail of blood as they were dragged, having to bag bodies and objects before disposing of them, distractible bystanders, and the presence of an active killer still 'goring up' the crime scene.

Software & Technologies

Visual Studio - IDE developed by Microsoft, used to edit cs and shader files

Unity - Game engine used for 2D and 3D rendering, and scripting.

GitHub - A code manager in the cloud, to share the source code and collaborate on the project.

Github for Windows

Gitkraken - GUI to use Github

C# - Programming language by Microsoft for Object Oriented Programming.

Google Drive - Google cloud based service to write up our documents and make our presentation slides, as well as edit them simultaneously.

Blender - Open source 3D creation tool, for some of the assets of the game.

Assets & Packages

Model Assets:

https://assetstore.unity.com/packages/3d/environments/low-poly-nature-197552

https://assetstore.unity.com/packages/3d/environments/low-poly-chemical-lab-186606

https://assetstore.unity.com/packages/3d/environments/urban/hospital-lowpoly-82552

https://assetstore.unity.com/packages/3d/vehicles/land/low-poly-police-car-01-142826

https://assetstore.unity.com/packages/3d/props/interior/low-poly-cartoon-house-interiors-167425

https://assetstore.unity.com/packages/3d/environments/urban/low-poly-street-pack-67475

https://assetstore.unity.com/packages/3d/environments/urban/simple-city-pack-plain-100348

Unity Packages:

Animation Rigging

Cinemachine

Input System

Text Mesh Pro

JetBrains Rider Editor

Post-Processing

Unity Test Framework

Timeline

Unity UI

Universal Render Pipeline

Version Control

Code Editor Package

Meeting Minutes

08/10/22 - Meeting for Project Presentation & Preparation

- Come up with the game name
- Find similar assets / images to see what our game will look like
- Defining the gameplay, game mechanics
- Spilt up the work for the presentation
- Started writing GDD
    - Added game scenario ideas, Mechanics, Aesthetics
- Set the Next meeting
- Who Came:
    - Paul
    - Shiraz
    - Sami
    - Roxane
    - Alex
- Meeting Time 1 hour 15 min

09/10/22 - Meeting for Project Presentation & Preparation [Part 2]

- Discussed more in depth about the presentation
- Clarified game dimension
- Started adding slides
- Reviewed eachother's parts
- Came up with a list of things to add, modify and fix
- Scheduled a new meeting to finalize the presentation for Tuesday, oct. 11.
- First meeting with our Mentor Dann
- Who Came:
    - Paul

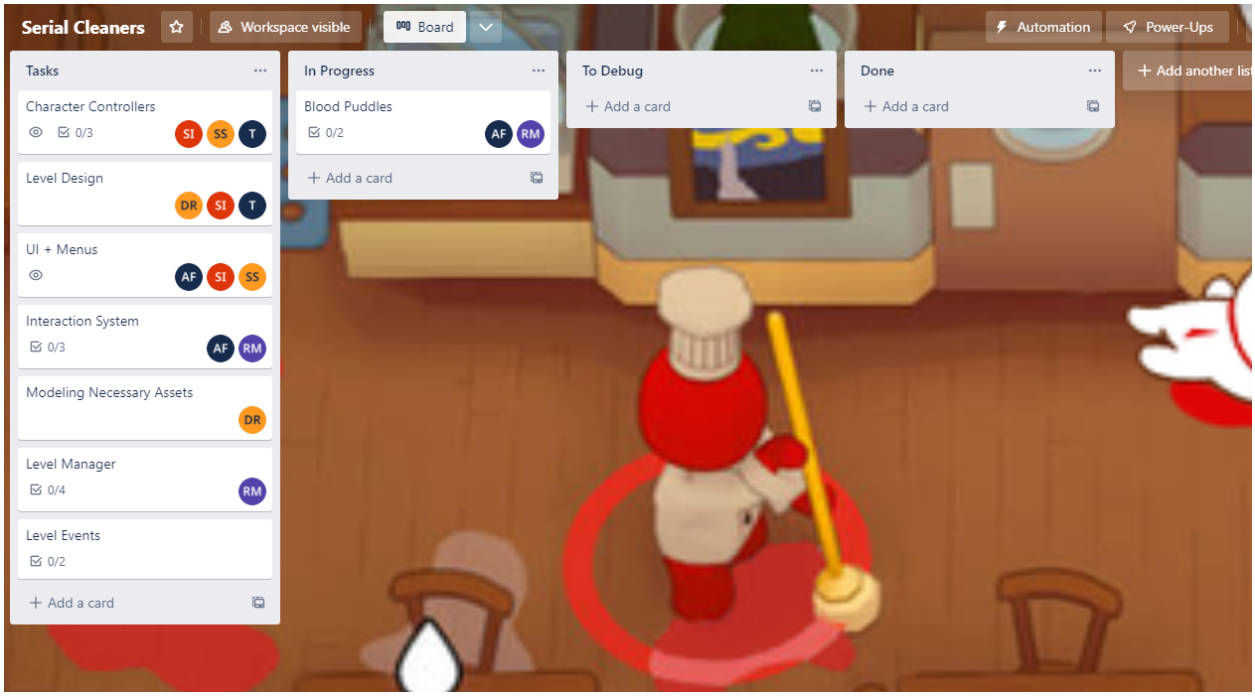
    - Shiraz
    - Sami
    - Roxane
    - Alex
    - Danny
- Meeting Time 40 min

11/10/22 - Meeting to Review the Project Presentation's Power Point

- Reviewed the power point to make sure all information was included and well presented.
- Discussed our plans for the presentation.
- Discussed future task assignments.
- Scheduled next meeting
- Meeting time ~35 minutes.
- Who Came:
    - Paul
    - Shiraz
    - Sami
    - Roxane
    - Alex
    - Danny

15/10/22 - Meeting to Assign Tasks and to get started on the Project

- Going over the Project Presentation and feedback from the Professor and TA as well as the classmates
- Made trello for everyone to work on
- Assigned tasks to everyone and deadlines



-
- Who Came:
    - Paul
    - Shiraz
    - Sami
    - Roxane
    - Alex
    - Danny
    - Dann (TA)
- Meeting time 1 hour
- Scheduled next meeting and came up with the goals we want for the next meeting

22/10/22 - Progress meeting

- Worked on aspects of the game
- People discussing expectations of work that will be accomplished
- Agreed on time frame for serious work (most of us had midterms)
- Came up with a list of things to add, modify and fix
- Who Came:
    - Paul
    - Shiraz
    - Roxane
    - Alex
    - Danny
- Meeting Time 25 min
-

29/10/22 - Progress meeting 2

- Worked on some aspects of the game
- Updated trello tasks

- Talked about the work done
- Assigned larger amount of work
- Agreed on time frame for serious work (most of us had midterms)
- Came up with a list of things to add, modify and fix
- Who Came:
    - Paul
    - Sami
    - Roxane
    - Alex
    - Danny
- Meeting Time 15 min

05/11/22 - Progress meeting 3

- Put together important aspects of the game
- Updated trello tasks
- Talked about most essential work to be done
- Assigned larger amount of work
- Agreed on time frame for serious work (most of us had midterms)
- Came up with a list of things to add, modify and fix
- Who Came:
    - Paul
    - Shiraz
    - Sami
    - Roxane
    - Alex
    - Danny
- Meeting Time 50 min

09/11/22 - Progression demo meeting

- Discussed what has to be done
- Assigned Very high priority tasks
- Worked on GDD
- Worked on slides
- Discussed who’s speaking or not
- Who Came:
    - Paul
    - Shiraz
    - Sami
    - Roxane
    - Alex
    - Danny
- Meeting Time 60 min

10/11/22 - Pre-Progress demo meeting

- Practiced Presentation part
- Made sure to remember which parts to demo
- Who Came:
    - Paul
    - Shiraz
    - Sami
    - Roxane
    - Alex
    - Danny
- Meeting Time 60 min

11/11/22 - Post-Progress demo meeting

- Reviewed what went right and wrong in the presentation
- Assigned more tasks that have to be done
- Who Came:
    - Paul
    - Shiraz
    - Sami
    - Roxane
    - Alex
    - Danny
- Meeting Time 60 min

19/11/22 - Feedback meeting

- Talked about presentation Feedback
- Updated tasks that should be done
- Reassigned tasks new tasks that should be done
- Who Came:
    - Paul
    - Shiraz
    - Sami
    - Danny
- Meeting Time 30 min

20/11/22 - Feedback meeting 2

- Talked about tasks with people who weren't present
- Who Came:
    - Shiraz
    - Sami
    - Danny
    - Roxane
    - Alex
- Meeting Time 15 min

03/12/22 - Progress meeting 4

- Merged more parts of the code
- Updated tasks that should be done
- Did testing on the levels
- Fixed bugs that were present
- Examined 2 player pick up extensively
- Who Came:
    - Paul
    - Shiraz
    - Sami
    - Roxane
    - Alex
    - Danny
- Meeting Time 40 min

04/12/22 - Progress meeting 6

- Merged more parts of the code
- Updated tasks that should be done
- Did testing on the levels
- Found more problems with the code
- Who Came:
    - Shiraz
    - Sami
    - Roxane
    - Alex
    - Danny
- Meeting Time 30 min

05/12/22 - Progress meeting 7

- Merged more parts of the code
- Updated tasks that should be done
- Did testing on the levels
- Fixed more problems with the code
- Decided to cut some features
- Who Came:
    - Paul
    - Shiraz
    - Sami
    - Roxane
    - Alex
    - Danny
- Meeting Time 30 min

06/12/22 - Progress meeting 8

- Merged more parts of the code
- Updated tasks that should be done

- Did testing on the levels
- Fixed more problems with the code
- Decided which parts are ones that should be showcased
- Who Came:
    - Shiraz
    - Sami
    - Roxane
    - Danny
    - Alex
- Meeting Time 50 min

07/12/22 - Pre-Final Demo Meeting

- Fixed the most of what could be fixed
- Discussed which parts are best to showcase and which to remove
- Added last minute quality of life features
- Who Came:
    - Shiraz
    - Sami
    - Roxane
    - Alex
    - Danny
- Meeting Time 1 h 30 min

References

https://www.gamesradar.com/moving-out-review/

https://stevivor.com/reviews/overcooked-2-review-time-head-back-kitchen/

https://www.geekyhobbies.com/plateup-indie-video-game-review/