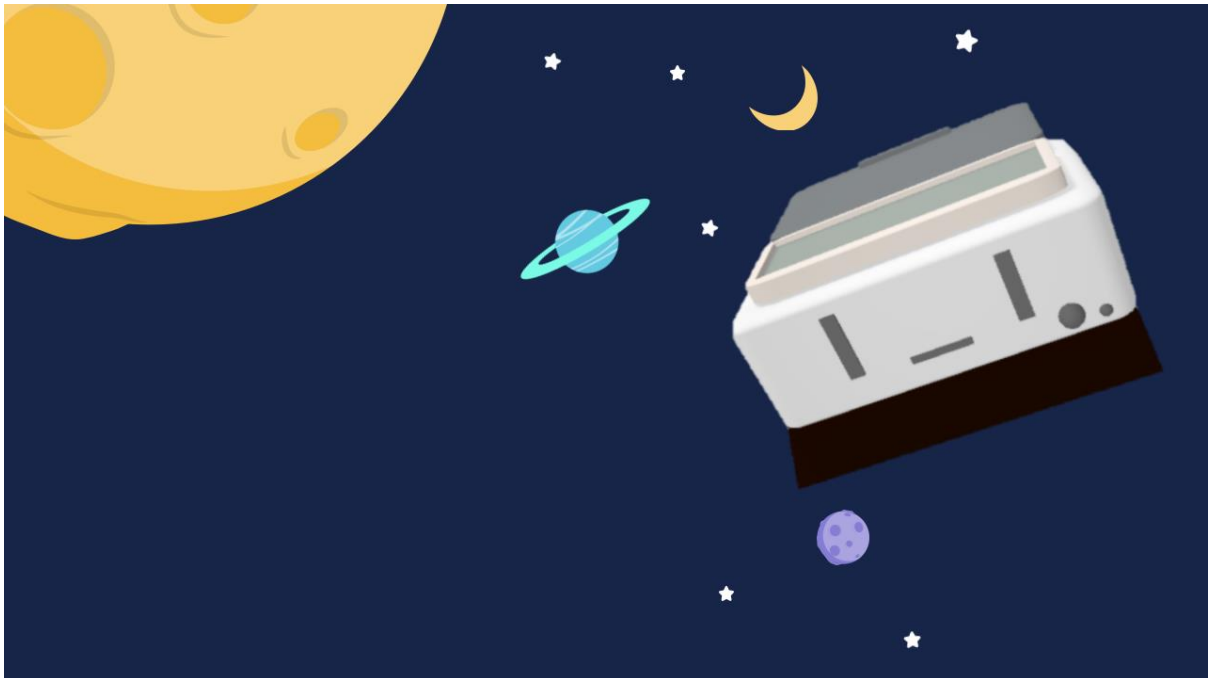


T1: Team Project

sigma – Dom Hawang Cha~



Team Member: 2017311829 Dongjun LIM,

2017313135 Dongmin KWON,

2019312518 Jiye Park

Index

- **Abstract**
- **Storyline**
- **Game Play**
- **Implementations**
- **Sources**

Abstract

Let's go over 2D/3D to solve puzzles to escape the witch's house. You have to solve puzzles and escape while avoiding game over within the time limit. Good Luck!

Storyline

Zetbot's monologue

My name is Zetbot. I'm robotic vacuum. I can't live in this house cleaning anymore! The structure of the witch's house is really strange. I'm tired of cleaning this house. I'm going to escape! It's a good time because the witch just left the house. If my battery runs out or the witch returns, I fail.

Game progress

As with everything, there seems to be an order in unlocking the room. Zetbot unlocks rooms by solving several puzzles. Obstacles and enemies are lurking everywhere. Zetbot wisely navigates these crisis. Zetbot struggles to figure out how to get the key.

Game Ending

Zetbot barely escaped the house before the witch returned! Zetbot is now free!

Zetbot has been captured. Zetbot has to start cleaning again.

Game Play

Step 1. Introduction

- Short intro-story & manual
- You can turn pages by pressing the 'N' key.

Step 2. Difficulty selection

- The number of rooms and the time limit are different for each difficulty.
- Others may differ depending on the difficulty level.

Step 3. In-Game Play

- 2D/3D mode can be selected by pressing 'F' key
- In 2D mode, you can only move forward and backward.
- In 2D mode, if it is blocked by an orthogonal object, it cannot move.
- In 3D mode, you can move up, down, left and right.
- In 3D mode, you can move a specific object by pushing or pulling.
- Solve puzzles in each room to get the key to unlock the next room
- The battery will shrink over time and can be recharged at any charging station.
- Enemies may exist on the map. Avoid enemies and clear the map, or eliminate them in a specific way

Step 4. Game Over

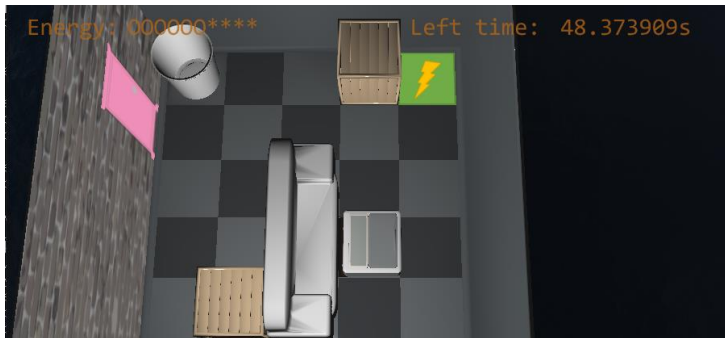
- The moment you eat the key is saved.
- When your character's battery runs out, you start over from the save point.
- If the time limit runs out, you will start over from the beginning!
- Game over if the character is obscured by orthographic objects when

transitioning from 3D to 2D.

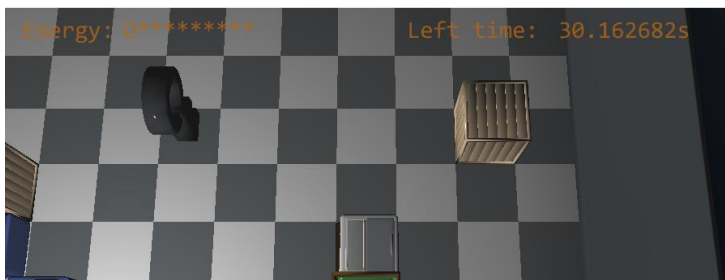
- If you get too close to an enemy, it's game over.

How to Solve each difficulty level

Level 1:

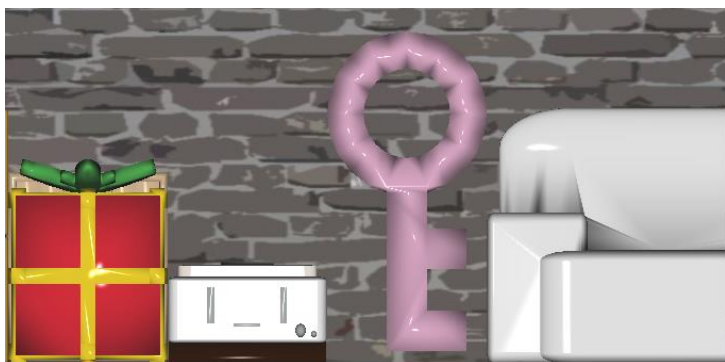


Move wood box into beacon and activate the door



Kill the enemy for getting key

Level 2:



Get key on 2D view



Move double_woodbox to activate bedroom's door(if the height of wood box is 1, it can't push beacon)



Get black key in 2D view(hided by double wood box)

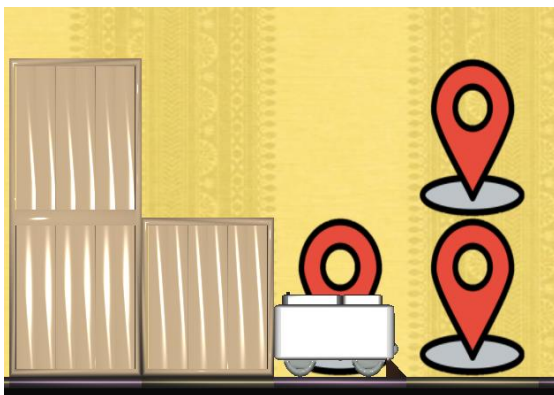


Have to kill enemy to escape.

Level 3:



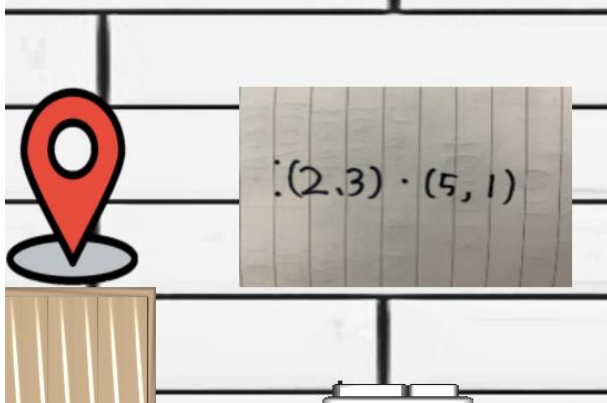
Kill enemy to get key



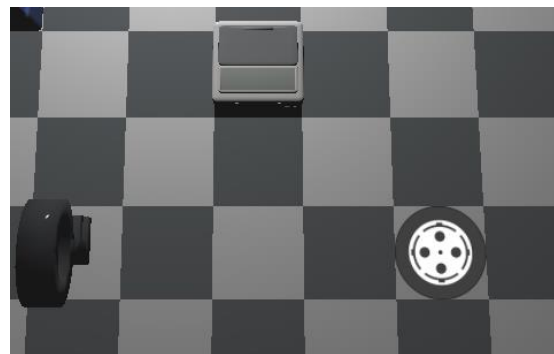
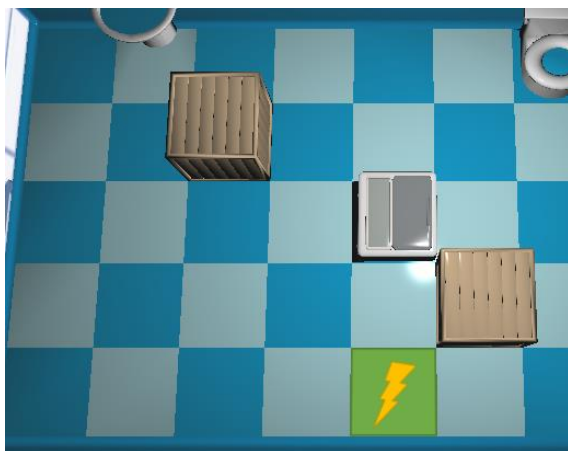
Move wood box to specific position to blind all check image in 2D view to activate next key.



Move wood box to check image's position



This message is related to position of wood box in bathroom.



Setting the woods position to activate living room's key and beacon

Implementation

Data Structure

Model.h

For object rendering, we add some features to model_t structure.

ID : It is related with What model It is.(wood box, rooms, keys) And it's used in render function as index of pMesh vector.

Active : if "active" 's value is true, it can be rendered on screen.

Movable : if it's value is true, then it can be moved by hero.

Cur_pos : if the map is grid(expressed in x and y coordinates), it indicates the position of

model object.

Theta : it is related to model's rotation

Wall.h

Wall_t is 2D object's structure. To set the vertex, we have to call create_wall function. In this function, we set position and norm texture. It has just 4 vertex and 6 index. To make 2D object, just change the image(jpg or png) and bind texture to wall.(door, 2D key, beacons)

Pos : if the 2D object's position is in floor, it indicates the position of the object on floor.

Direction : if the 2D object's position is wall(room's wall), the direction value 0 means that the object is on left wall, else if the direction value is 1, it means that the object is on front wall.

Wallpos : it is where the object position on the room's wall(not on floor)

Wallpos_z : We can set the z-position of object by changing this value.

Map.h

We set the map as grid to make easy to deal with position or moving objects. There are 5 maps(rooms) for game implementation. So we express that rooms by 2D map and set each position's value as number that indicates object's index in specific rules.

ID : it is room number.

0 – warehouse

1 – living room

2 – kitchen

3 – bedroom

4 – bathroom

Grid : size of room

Map : object setting in the room

State

To make checkpoint, we have to save state of save points. So when the hero get keys. We set all room's setting to save_states structure. There are scene, map setting, model's state(moving pattern, position, rotation... everything of object), wall's state to save.

Hero State

To deal with time and charging we make separated structure as "herostate". We can set the initial amount of energy, total_time, decrease_rate by difficulty.

Text/Image rendering

Text and images are output fixedly relative to the screen. The image is positioned slightly behind the text so that the text is output over the image.

To make Text resizable we make function "txtposx" and "txtposy" to calculate the position of text.

Sound rendering

The sound source file imported from the outside is used in the load. During game over, the game over sound is output and the background sound stops. When you clear the game, the ending voice is output.

Projection/Camera/Light

In 2D mode, we use an orthographic projection and a camera view from one side. Also, if in 2D mode, the light is also directional light coming from the camera direction (visibility issue).

3D mode uses a perspective projection and an oblique camera view from above. (To give it a 3D feel). In 3D mode, the light is a point light source hanging from the ceiling of the room.

Keyboard input(checkpoint, reset)

In game:

R – go back to checkpoint

B – reload the level

P – pause(time pause, and can't move)

Q – reload the game(go to init page)

H, F1 – help screen

Game clear:

R – reload the game(go to init page)

Skybox

Using Cube Map. The coordinates were slightly different, so I edited the image appropriately.

Model Setting

There are two types of model objects. 3D Objects(Map, Character, Enemy), 2D Objects(Wall, door, beacon...)

Certain objects can only be interacted with in certain modes.

In the case of a 3d object, unique vertices and textures were used, and in the case of a 2d object, a quad was directly drawn and individual textures were applied on it.

Particle

It mainly occurs when an object is affected, and each particle has a different shape and movement. Particles are composed of 2d images.

Sources

- Except for the wall texture, all images were made by ourselves.
- For wall textures, images that are free for non-commercial use were used.
- In the case of the 3D model, some of the models made by hand from 0 and the models released as open sources were imported and modified and used.
- As for the sound source file, the background sound is popular music, the game over

sound is BGM file, and the game ending sound is self-produced and used.