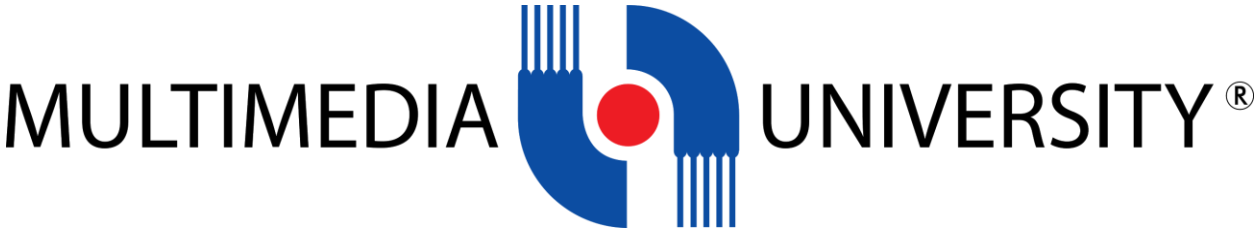


TGD3351 GAME ALGORITHMS

Trimester 1, 2015/2016

Game Project

Spell Alchemy: The Fortress Escape

Member

Chia JianFei 1131120642

Lim Wan Ping 1121119034

Introduction

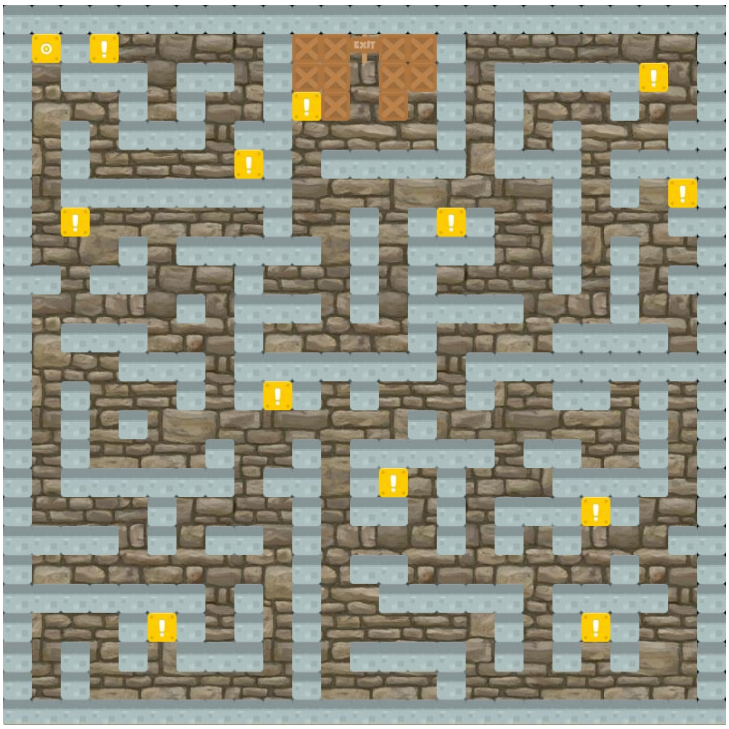
Spell Alchemy: The Fortress Escape is a basic hide-and-seek game. The stage was set as Chandler is inside the fortress and spotted by Dark Wizard. To escape the fortress, Chandler must use the invisible spell on specific location, provided that location have enough power to conceal his presence towards Dark Wizard. To win this game, player must make his way to the exit with all the quest items collected. The game score will then be computed based on the time taken to finish the stage mission and also the number of bonus stars looted.

Background Story

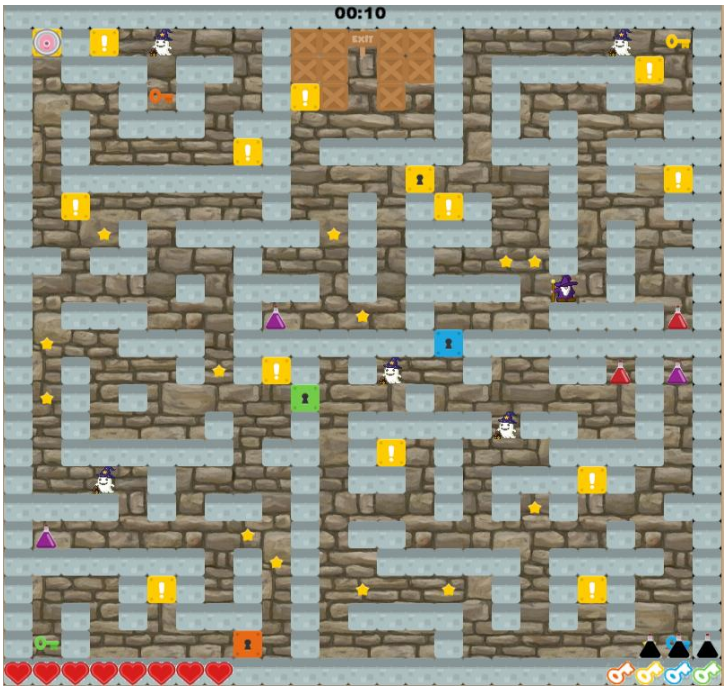
Chandler, a minor self-learning wizard, who want to save his friend who is cursed by a power spell. In order to lift the deadly spell, a magical spell potions is required and these potions only exists in Dark Wizard Fortress. To save Joe's life, Chandler must infiltrate the fortress, retrieve the potions and escape safely.

Game Design

Level Design



Level Design with Objects Layout & HUD

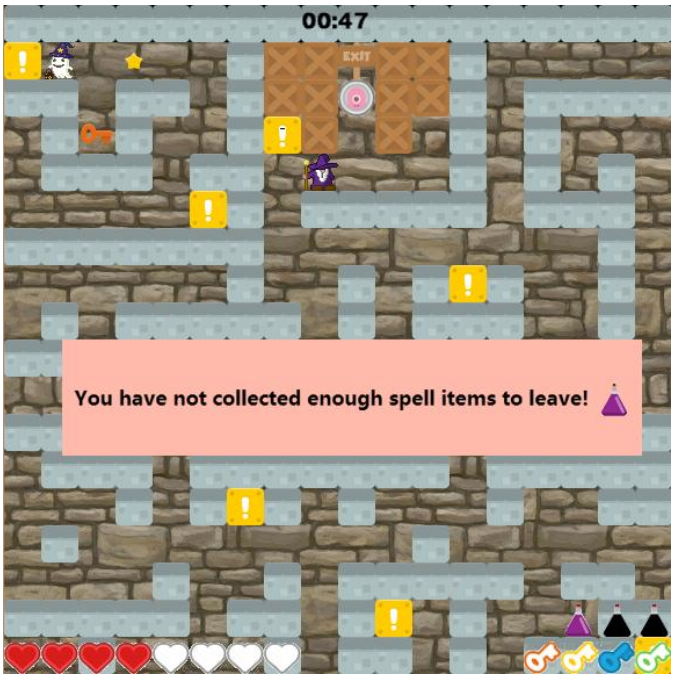


Home Screen Design

Game Over Screen

You Won Screen

Message box that pops out when spell items collected are not sufficient



Technical Design

Wall/Block Detection

Wall detection is one of the core technical implementations in tile-based game. In the development, all movement are implemented with unit synchronization (one unit equal to one tile, instead of pixel). Therefore, the game algorithm for detection can easily know the position of the wall, based on current tile position and the map references.

Path finding Algorithm

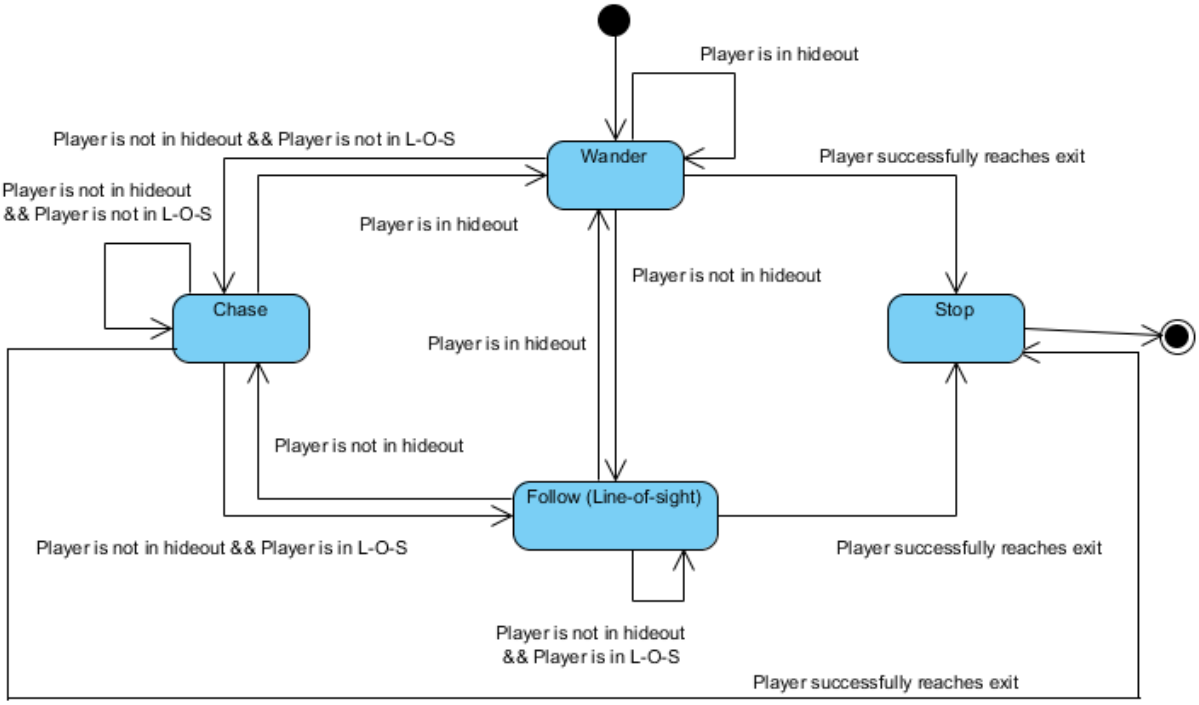
The path finding algorithm implemented in this game is A* algorithm. The algorithm search the adjacent walkable tiles, added into Open List and compute their cost to the destination. Once the computation is finished, the algorithm will pick the lowest cost alphabetically or sequentially and add that tile to Close List. The algorithm is prototyped using the pseudo code, provided in the lecture and the youtube tutorial on this topic. The reference link is as follows: https://www.youtube.com/watch?v=-L-WqKMFuhE

Finite State Machine

A Finite State Machine is introduced in this game on the Wizard AI. The involved states including wander, chasing, line of sight chase (follow) , and stop. All these actions are based on player's hiding status or whether he is in the line of sight by the wizard.

Wizard AI Behaviour:

To improve the thrilling fun factor of the game, the wizard will be constantly on chase mode to chase after player. When player is visible in line-of-sight, he will switch to follow mode where A* pathing is not required in the algorithm. He will switch to wander mode only if the player is hiding in the hideout. The dark wizard will only stop when the player makes it to the exit, as depicted in the state machine diagram below.



Minion AI Behaviour:

Minions will move back and forth between 2 points in their respective designated patrolling path that is technically calculated with a Starting Patrolling Position and the coverage (value) of the path.

Score System

The score is computed using the following formula:
Score = Estimated finishing time (in seconds)/Total finishing time (in seconds) * 100 +
Number of stars looted * bonus marks a star carries

Game development issues

Attraction Point

The main attraction point of this game is the hide-and-seek mechanics itself. As the almighty Dark Wizard will keep chasing after player when he is not in the hideout, the difficulty designed has provided a significant boost to the thrilling fun factor to the game.

Changes Made

1) The idea of collecting coins to increase special point and use that special points to unlock ability was removed. Instead, the coins are replaced with star, which is part of the score calculation. From technical point of view, implementation of ability is achievable. However, on playability point, such ability might decrease the difficulty of the gameplay, which makes the concept of hiding become optional, and not a key feature.

2) The idea of implementing Checkpoints is also removed. Technically, it should be achievable by remembering and storing the state when a checkpoint is hit. However, due to low priority and time constraints, this idea is omitted.

3) Wizard now does 3 damages to the player upon contact instead of killing the player instantly. In terms of playability, it is extremely difficult to play if the player is instantly killed upon touching the wizard.

4) Penalties are removed, and power-up are altered. Instead of placing “Illusion” or “Invisible” power-ups, “HP potions” are implemented. Technically, the trap can be randomized on walkable paths. It is not difficult to implement, but in term of playability, it can make the hard game even harder, and that is the reason we would like to omit the idea of implementing penalties. “HP potions” are implemented to decrease the game difficulties.

5) The idea of alchemy is simplified into collecting multiple key items to leave the fortress, instead of mixing the items to obtain the key item. This change was made majorly due to time constraints.

Discussion

If more time is given in this project, few elements can be developed better to enrich the player’s gameplay experience. Such elements including the type of minions, different moving and decision making behaviours. Secondly, more levels could be designed with different themes and layout, similar mechanism with more complicated maze designs. Additionally, more power-ups will be designed to accommodate different stronger enemies in the game.

The current alchemy system is simplified. If more time is given, we would like to incorporate a fusion system to the current existing alchemy system, which gives better look and feel to players.

User manual

At the home screen, simply left click or press space on the keyboard to start the game. The player can move by using arrow keys (Up, Down, Left, Right) or using 'WASD' keyset. The main mission is to collect all 'Quest Item' in the fortress, and make your way out to the exit without getting captured by the dark wizard. The player can use of the enchanted location as hideout to avoid being detected by the dark wizard. However, there is a time limit (few seconds) you can stay in a hideout, exceeding the time limit would force you out of it. There are also red potions in the fortress that restore portion of player's health when looted. Besides, there will be random stars scattered around the fortress that are meant to add bonus marks to the game score. The game score will vary depending on the total time player spend on clearing the stage and the number of stars looted.


CHANDLER


ENCHANTED LOCATION


THE MINIONS


THE DARK WIZARD


THE KEYS


THE LOCKS

EXIT SIGN


STARS


HP POTION


QUEST ITEM

Contribution Statement

In this project, we have come to a consensus to share the points evenly (50%-50%). Although some arguments happen during the development, but we believe it is essential to deliver good product.

Highlight of the workload:
Hyrex:
Tile Loader, Camera, Player Movement,
Pathing Algorithm, Assets Finding and editing,
Sound Assets and Implementation
Code Modulation and refactoring.

WanPing:
Map designs, HUD, Items mechanism,
Hideout timer, Score systems,
Screen designs and transition,
Sound Tweak, Minions Patrol Behavior,
Code Refactoring.

Honor Code & Reference

Sprite Sheets

Generated by TILED Tile Engine Editor, Image Sources (Mostly) : Kenney's Asset. Kenney is a game and app development studio. The images are the open source provided by the studio in the official site and opengameart.org. Reference: **http://kenney.nl/assets**

Musics

Background Music: Tomba! Game OST, retrieved from youtube.
Button, Footstep, Loots sound, retrieved from youtube channel called 'gamingsoundfx'. Reference: **https://www.youtube.com/user/gamingsoundfx**

Algorithm

Pseudo code source: Lecture Slides and youtube video on this topic. The reference link is as follows: **https://www.youtube.com/watch?v=-L-WgKMFuhE**

TroubleShooting

The Camera and Tile system is referring to the tutorial from XNA Resources. Modification was made for the best fit in our project. XTiled was previously used but then removed because the incompatibility of the map import. Therefore, we create our own way to load the map by using the partial code in the tutorial, and Faris tutored us on the concept of "how camera works".
Reference:**http://www.xnaresources.com/default.asp?page=Tutorial:TileEngineSeries:1**