CSCI 4120 Final Project Phase 2

Group 8 - Project Report

3D Tower Defence Game

LEUNG Ho Fung

LIU Lap Kin

YANG Fu Chia

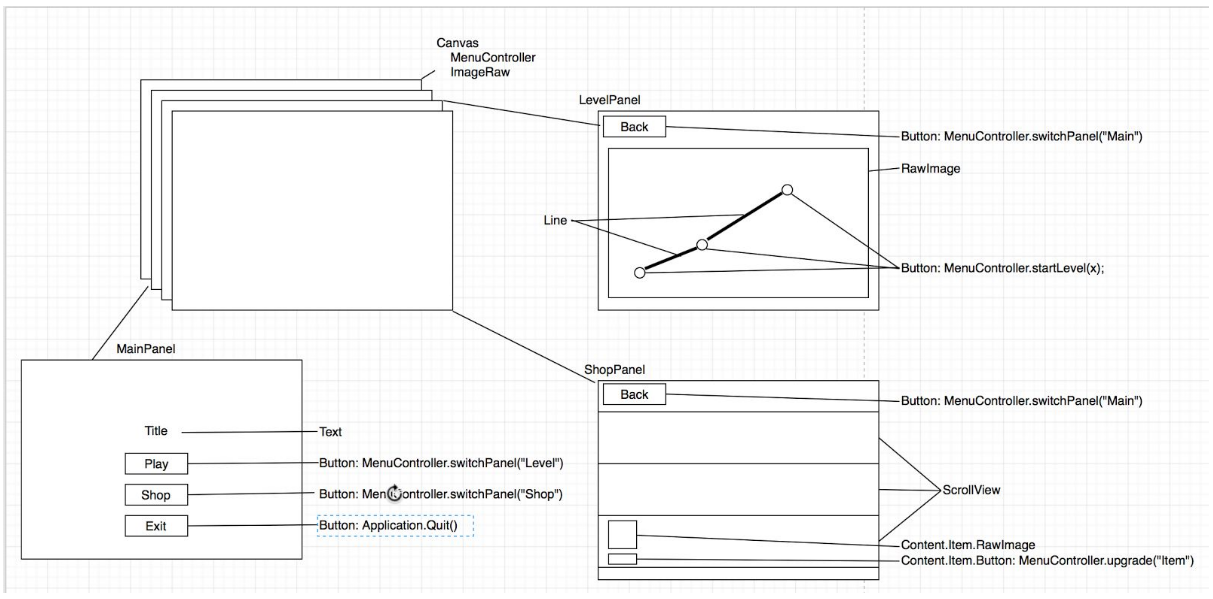
YOO Gigyun

1. Introduction

Mixing different types of games together helps to make the game become more attractive. Therefore, this game not only contain the traditional characteristics of the tower defense, but also includes the role-player game element. The players are no more just acting as a god who builds the towers and looks at the towers attacking the monsters, they can play as a mobile tower to kill the monsters. They can use their guns, traps or barriers in order to protect the castle from waves of monsters. Since the players can walk on the map to attack the monsters or build the towers, the monsters will also attack the players too. The game will be failed when the castle is destroyed, time is over or the players' health are less or equal to zero. This game provides different levels to players for challenging themselves. As the later levels are more difficult, shop system is introduced. The players can get money from killing the monsters and use the money to buy and upgrade the towers or guns as to make themselves more powerful. The aim of players is to clear as many levels as they can.

2. Design

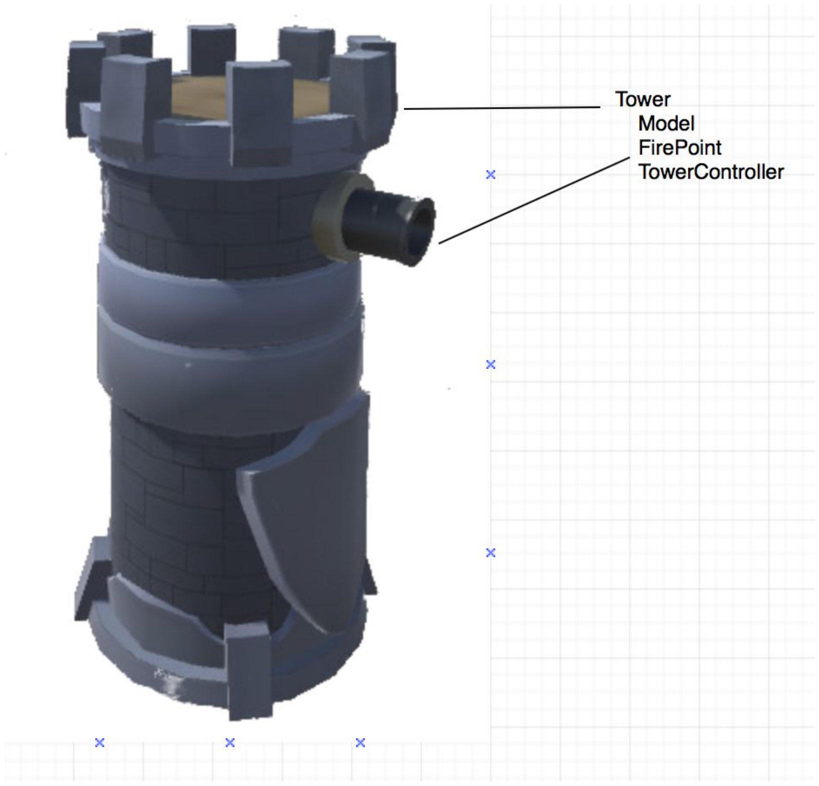
2.1 Menu


Canvas
MenuController
ImageRaw
LevelPanel
Back
Button: MenuController.switchPanel("Main")
RawImage
Line
Button: MenuController.startLevel(x);
MainPanel
ShopPanel
Back
Button: MenuController.switchPanel("Main")
Title
Text
Play
Button: MenuController.switchPanel("Level")
Shop
Button: MenuController.switchPanel("Shop")
Exit
Button: Application.Quit()
ScrollView
Content.Item.RawImage
Content.Item.Button: MenuController.upgrade("Item")

MenuController extends MonoBehaviour

public function	
Start	extends function of MonoBehaviour, set up levels and shops' information. Data are stored in PlayerPrefs, affecting namespace "TowerX", "BarrierX", "WeaponX", "Money", "Progress"
switchPanel	switching between panels. params: string: name of panel, accept "Main", "Shop" and "Level"
exit	providing callback function for Exit button, directly call Application.Quit()
startLevel	load the corresponding level scene SceneManager.LoadScene("level" + level); params: int: number of level, start from 0
upgrade	increase tower/barrier/weapon level, reduce money. params: string: object to upgrade, accept "TowerX", "BarrierX", "WeaponX"

2.2 Tower

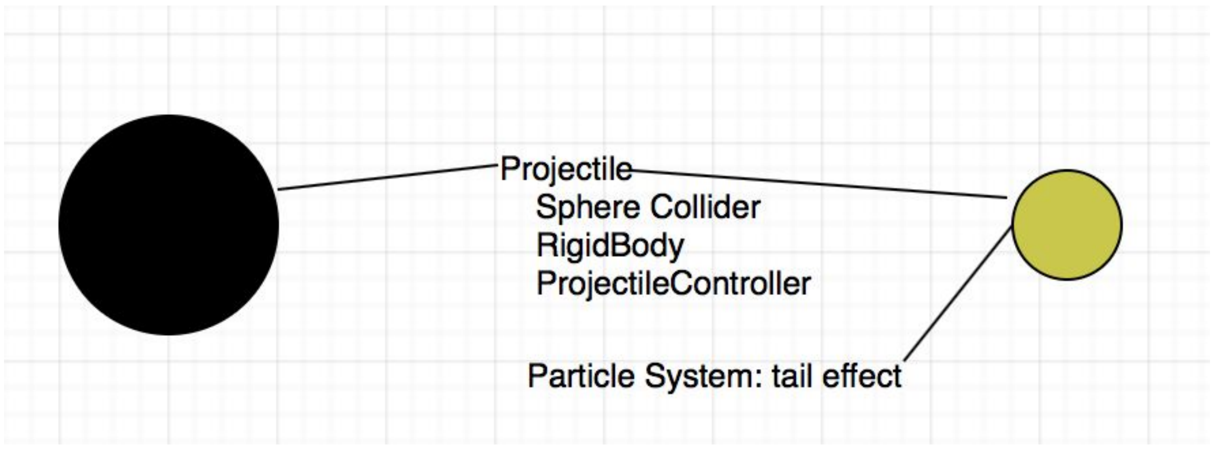


TowerController extends MonoBehavior

variable		
tag	string	“Tower”
firePoint	GameObject	reference position for creating projectile
cd	float	attack cooldown
Projectile	Prefabs	Prefabs to spawn when fire
range	float	range that the tower can fire
public function		
Start	extends function of MonoBehaviour. initialise RangeIndicator. register to map	

Update	extends function of MonoBehaviour.
	if cd < 0
	    Find GameObject with tag “Monster” && in range
	        rotate to face target
	        instantiate Projectile w/ ProjectileController
	        set direction of Projectile
	    cd -= Time.deltaTime	

2.3 Projectile



ProjectileController extends MonoBehaviour

variable		
tag	string	“Projectile”
speed	float	speed of the projectile
direction	Vector3	direction to fly (normalized)
damage	int	damage cause to hit Monster
public function		
Start	extends function of MonoBehaviour. initialise RigidBody.vecolity = direction * speed	

OnCollisionEnter	triggered when hit collider
	
	param:
	    Collision: collision information
	
	if collision.transform.root.tag == “Monster”
	    apply damage on Monster
	destroy self	

2.4 Barrier

BarrierController extends Hurtable

variable		
tag	string	“Barrier”

2.5 Trap

TrapController extends MonoBehaviour

variable		
tag	string	“Trap”
type	string	type of the trap, accpet “range” or “single”
range	float	range of affected, only useful when type == “range”
damage	int	damage cause to hit Monster
public function		
Start	extends function of MonoBehaviour. initialise RangeIndicator. register to map	

OnCollisionEnter	triggered when hit collider
	
	param:
	    Collision: collision information
	
	if collision.transform.root.tag == “Monster”
	    if type == “single”
	        apply damage on Monster
	    else if type == “range”
	        Find GameObject w/ tag == “Monster” && in range
	        apply damage on Monsters
	    unregister to map
	    destroy self	

2.6 AttackableController

AttackableController extends Monobehavior

variable		
tag	string	(depends on the inherited class)
hp	int	amount of health indicates how much damage the object can endure.
attack	int	strength that indicates the damage of each attack
defense	int	defense point that reduces damage of each attack

range	float	range that the object can attack
speed	float	speed that the object can attack
cd	float	cooldown for next attack
isAlive	bool	to check whether the object is alive. The object could be dead but still playing dying animation.
public function		
Start	extends function of MonoBehaviour. initialize variables and register to map	

Update	extends function of MonoBehaviour
	
	if isAlive == false
	    this.GetComponent<MeshRenderer>().material.color.a = Mathf.Lerp(1, 0, t)
	    t += Time.deltaTime * (1.0f / 2)
	    if t > 1.0f
	        Destroy(gameObject)	

OnCollisionEnter (abstract)	triggered when hit collider param: Collision: collision information	

hurt	called when this entity being attacked
	
	params:
	  int: damage
	
	if this.isAlive == true
	    this.hp -= damage
	    if this.hp <= 0
	        this.isAlive = true
	        die()	

die (abstract)		

2.7 Monster

CharacterController extends AttackableController

variable

tag	string	“Monster”
money	int	amount of the money that the monster drops
weapon	Prefab	weapon which the monster carrying
route	List<Vector3>	monster will follow route to base
public function		
Start	extends function of AttackableController. initialize variables.	

Update	extends function of AttackableController.
	
	if distance(base) <= range
	    Attack base
	else
	    follow route	

die	player addMoney(this.money)	
attack (abstract)		

2.8 Character

CharacterController extends AttackableController

variable		
tag	string	“Character”
money	int	amount of the money that the character have
holding	GameObject	Item which currently carrying (weapon)
towers	List<>	list of towers that the character created
barriers	List<>	list of barriers that the character created
weapons	List<>	list of weapons that the character created
traps	List<>	list of traps that the character created
public function		
Start	extends function of MonoBehaviour. initialize variables and Lists. register to map	
Update	if (mouse.clicked)	

	switch (holding.type)
	    Weapon:
	        fire()
	if (keyPress)
	    switch (key)
	    1:
	        changeWeapon
	    2:
	        buildTower
	    3:
	        buildBarrier
	    4:
	        buildTrap
	
	if selectNewItem
	    Destroy holding
	    instantante new item

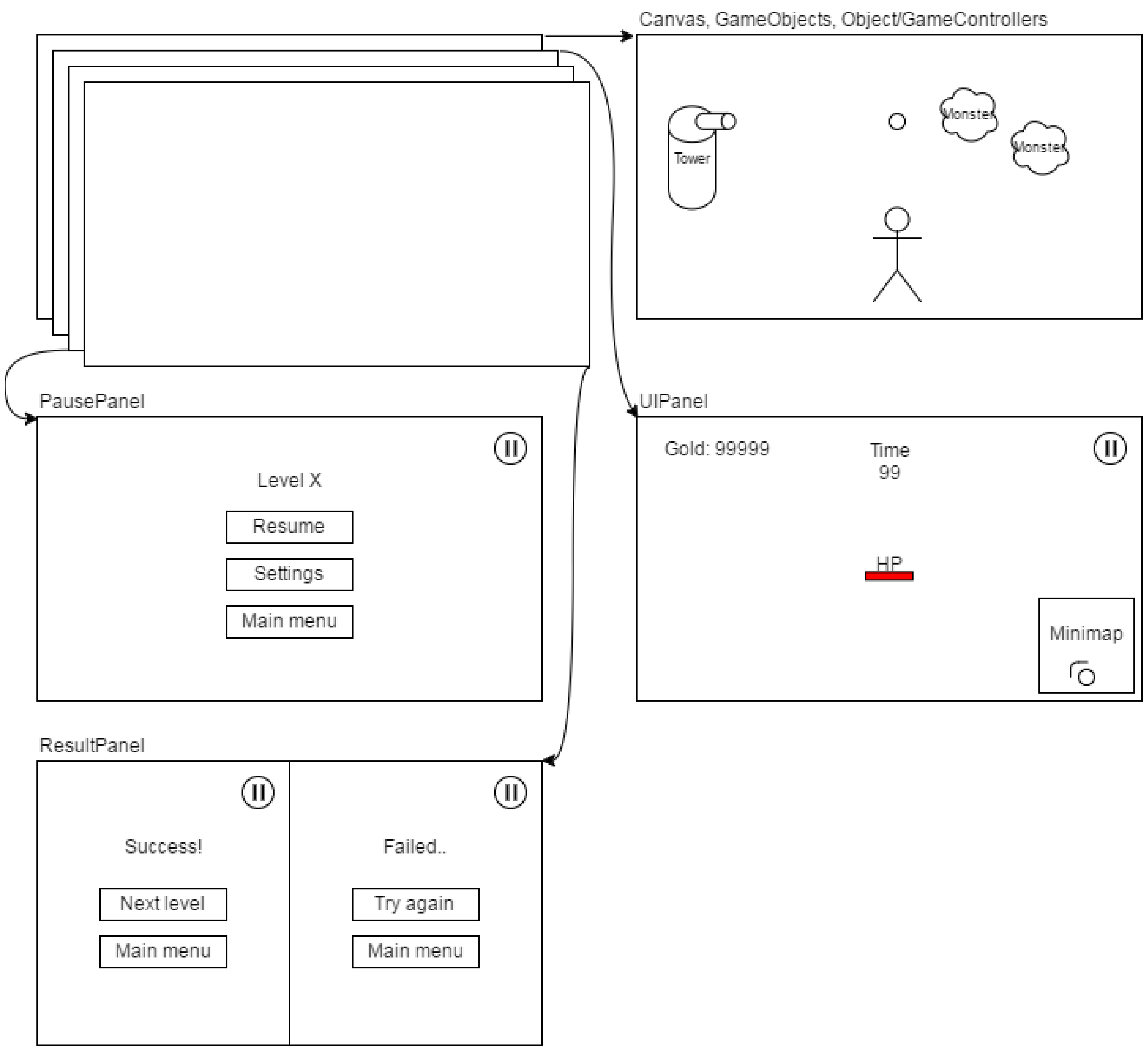
buildTower	triggered to build tower
	
	param:
	    Vector3: pointing position
	    int: tower id
	
	int[][]: target = getGridPoint(pointing)
	if (target is placeable && dont have tower)
	    instantiate new Tower

buildBarrier	triggered to build barrier
	
	param:
	    Vector3: pointing position
	    int: barrier index
	
	int[][]: target = getGridPoint(pointing)
	if (target is walkable && dont have barrier)
	    instantiate new Tower

changeWeapon	triggered when select weapon param: int: weapon index

fire	if this.cd <= 0
	    instantiate bullet
	    set bullet damage
	    set bullet transform
	    set bullet velcoity
	    this.cd = 1 / this.speed

2.9 Game



GameController extends MonoBehaviour

variable	
Time	time remaining to play
public function	
switchPanel	switching between panels. params: string: name of panel, accept "Main", "Pause" and "Result"

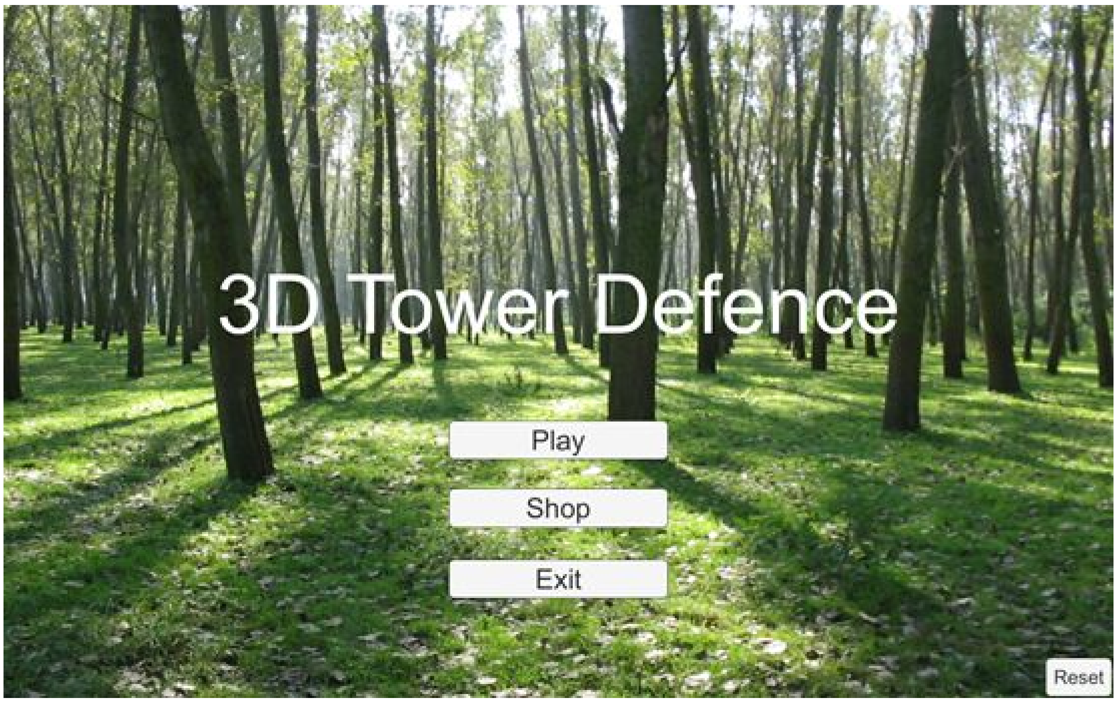
nextLevel	providing callback function for “Next Level” button
	
	levelName = SceneManager.GetActiveScene().name
	level = levelName.Substring(levelName.Length - 1)
	SceneManager.LoadScene(“level” + level + 1)

Update	extends function of MonoBehaviour (calculate time, update map objects info(e.g. HP/minimap))

3. User Menu

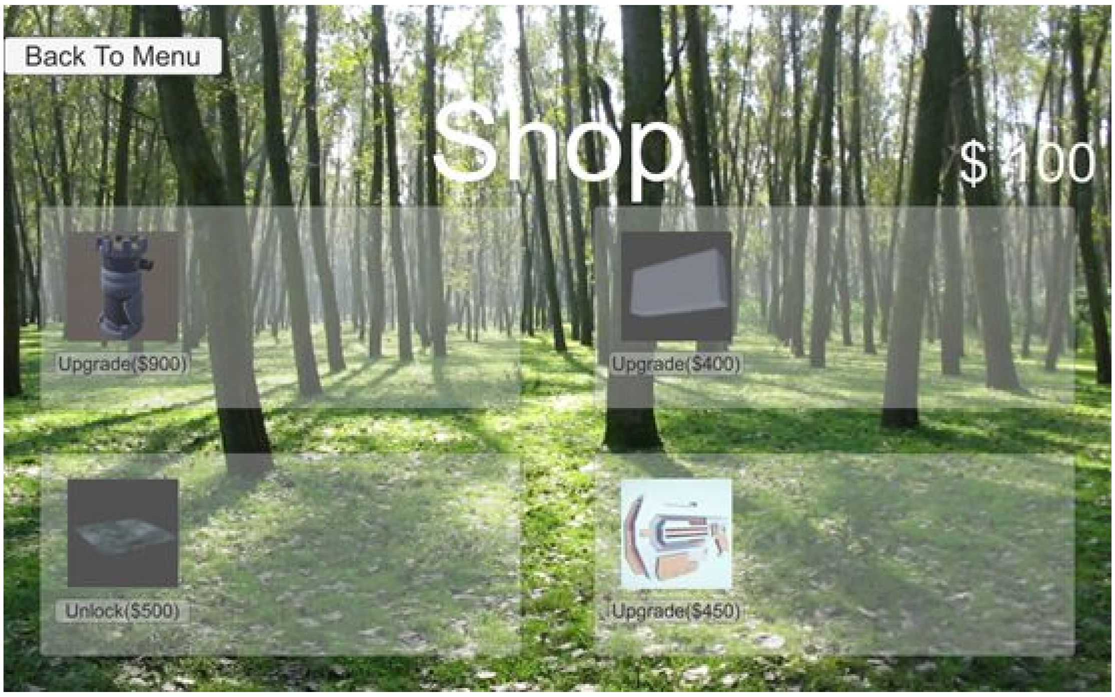
3.1 Start Menu

The start screen that can let players choose different options such as play the game, go to shop or exit.

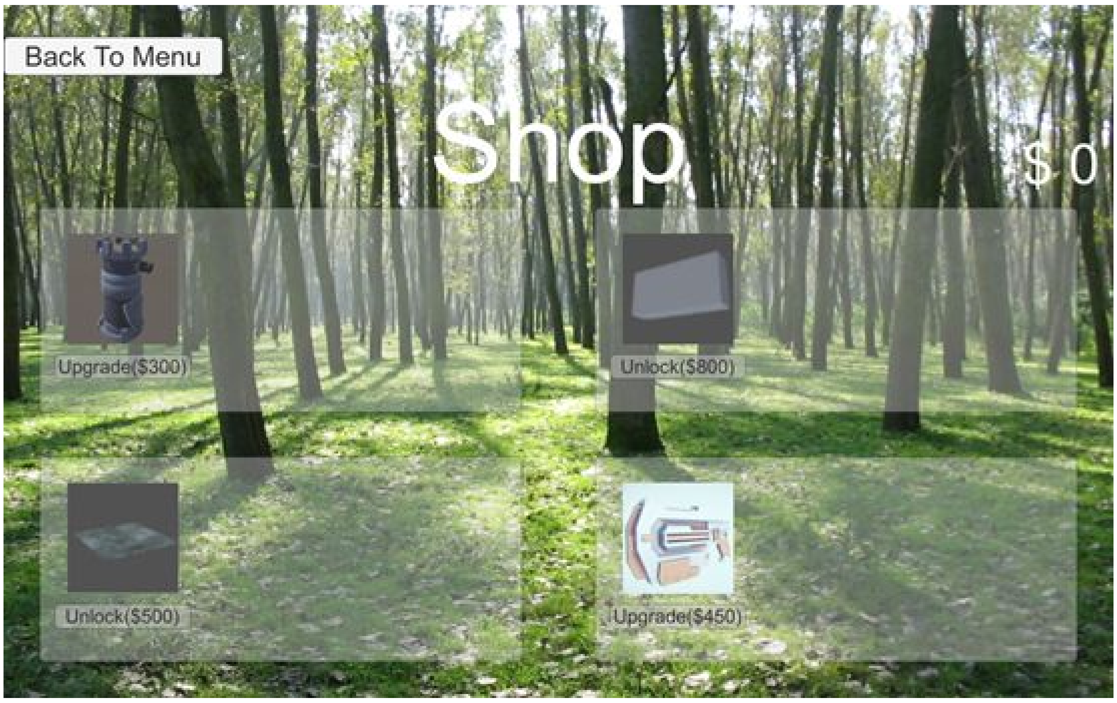


<u>Play</u>: Go to level menu
<u>Shop</u>: Go to shop menu
<u>Exit</u>: Terminate the program while storing the process
<u>Reset</u>: Reset the game to the initial, including the money, levels and items

Before reset:

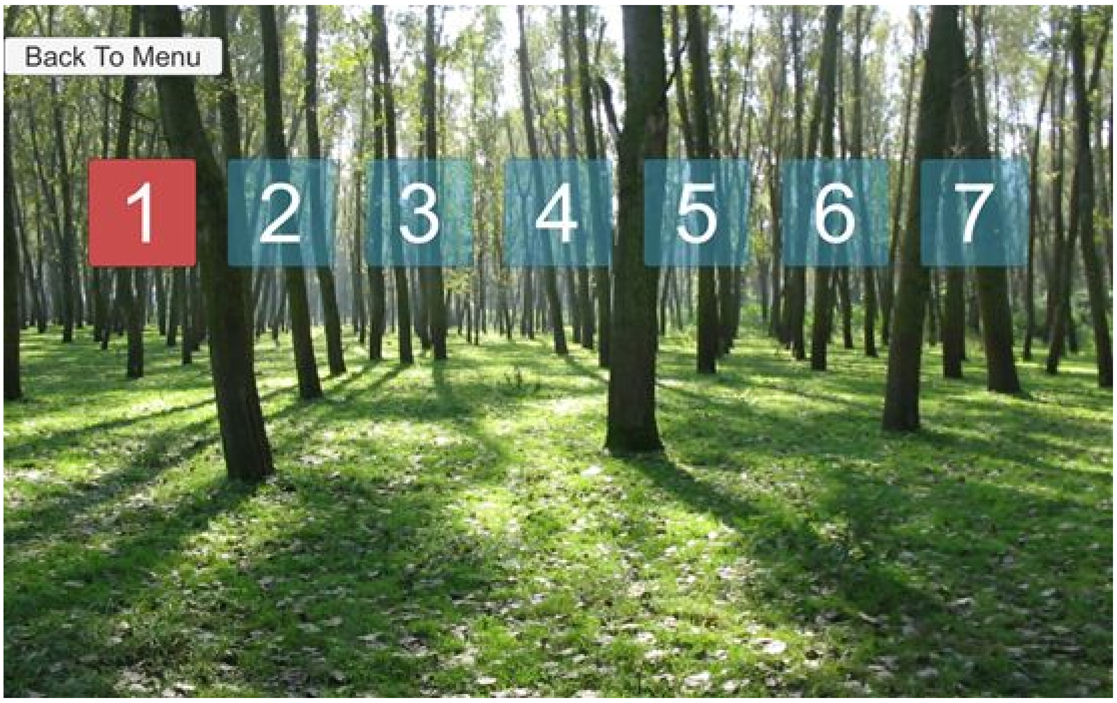


After reset:



3.2 Level Menu

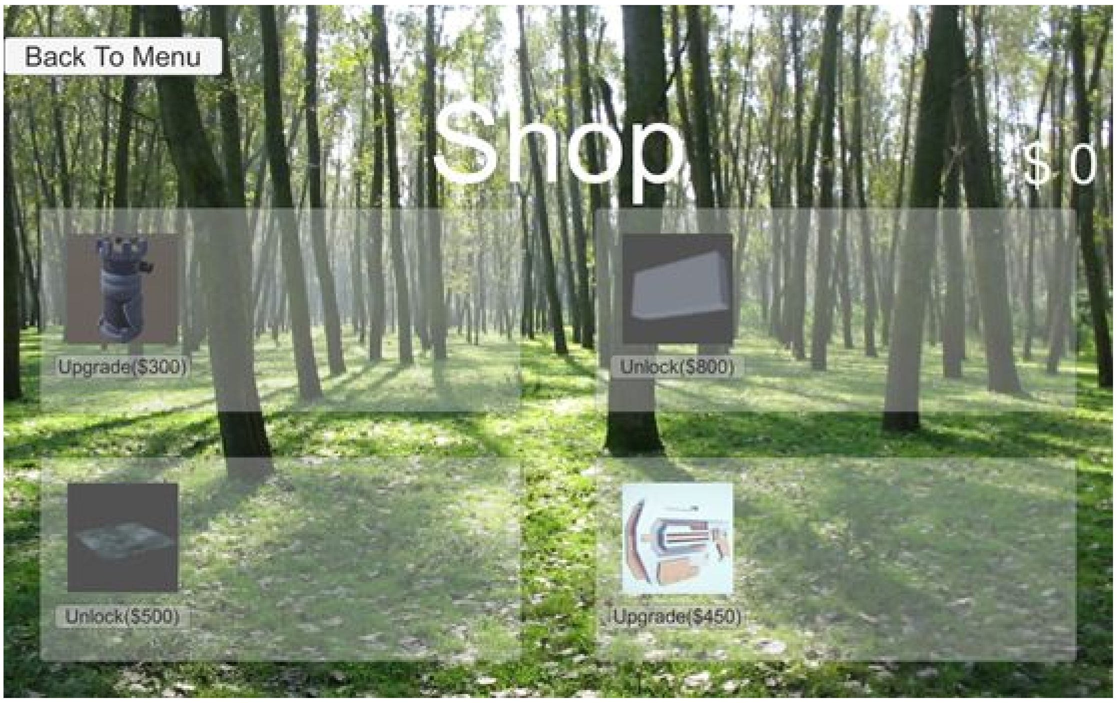
Level screen that let players choose which level they want to play.



Back To Menu: Go to start menu
Level 1/2….6/7: Play the specific level. Red icon means that level is in challenge mode.

3.3 Shop Menu

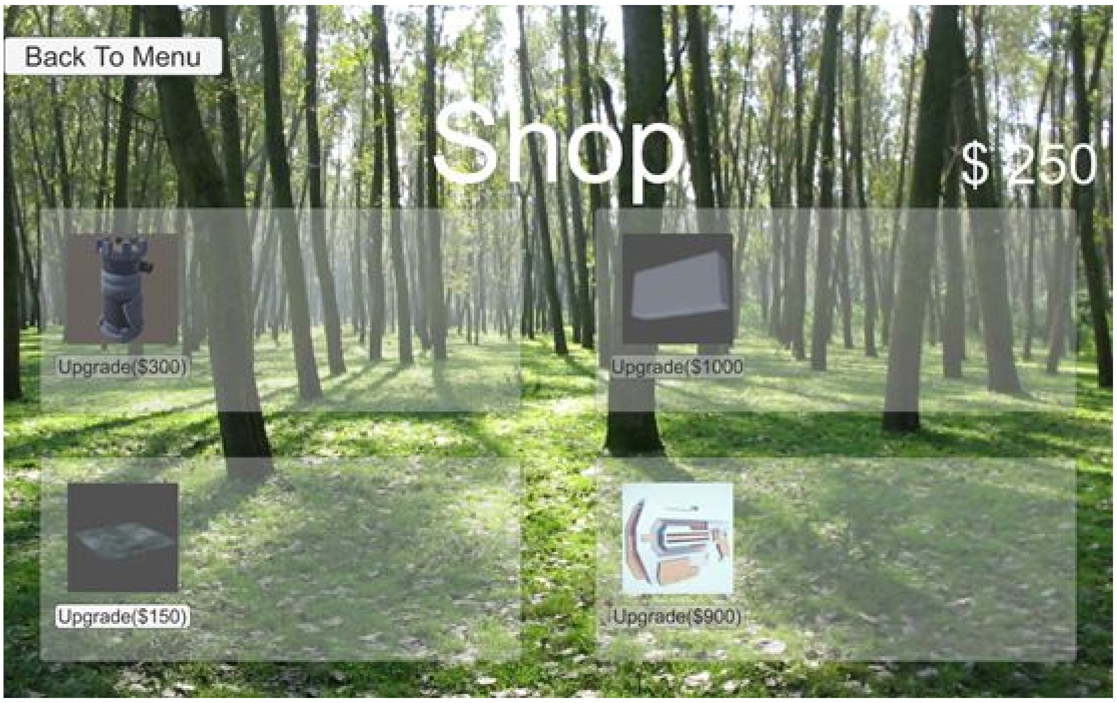
Shop screen that let players buy or upgrade the items.



<u>Back To Menu</u>:	Go to start menu
<u>Upgrade/Unlock</u>:	Upgrade or unlock the item, this will cost specific amount of money
Money:	The amount of money that players own is shown at the top right corner

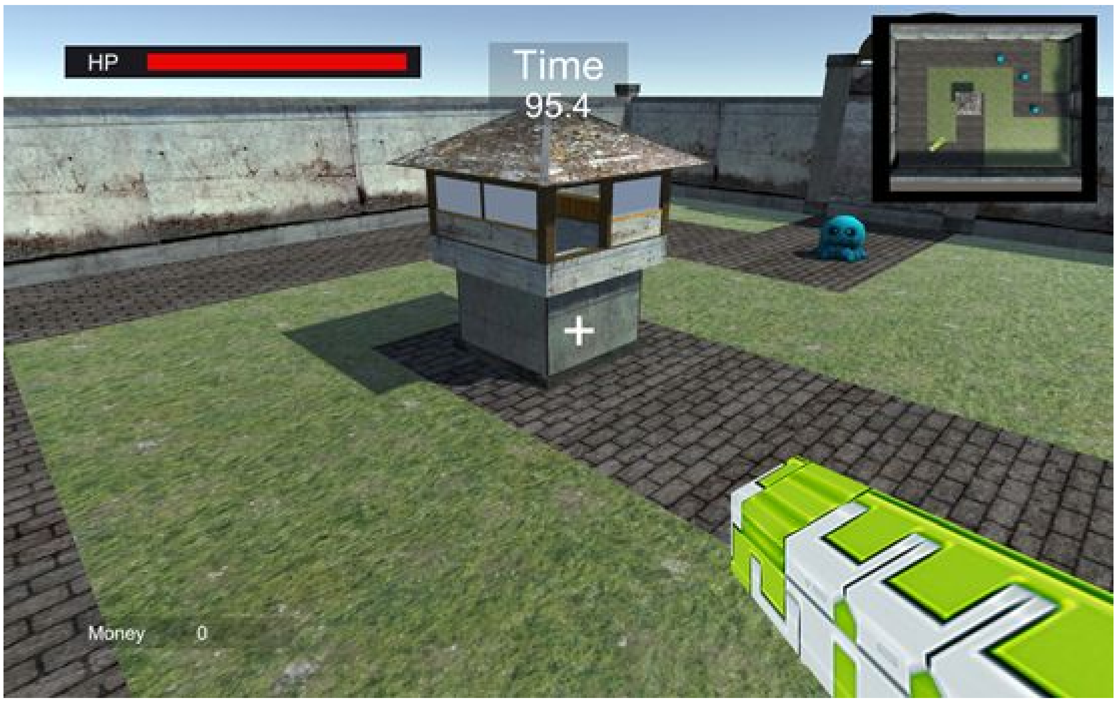
Money can be got from killing the monsters in game.

The players can upgrade or unlock the items only if they have enough money



3.4 In Game

3.4.1 Basic

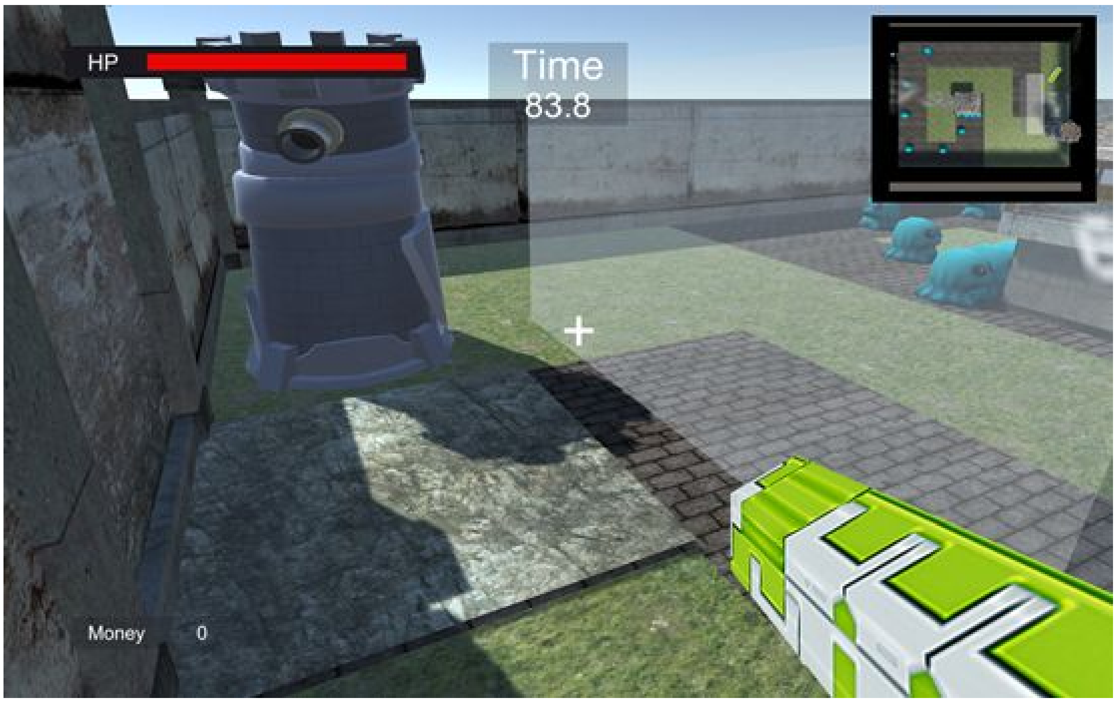


<u>Movement</u>:	"w" for forward, "s" for backward, "a" for leftward, "d" for rightward
<u>Jump</u>:	"space bar" for jumping
<u>Fire</u>:	"left mouse button" for firing the players' weapons, the power is according to the level of the gun.
<u>Tower</u>:	"2" for placing the tower, the tower will automatically attack the monsters, the power is according to the level of the tower.
<u>Barrier</u>:	"3" for placing the barrier, the barrier with a durability will stop the monsters from moving, but the monsters will attack the barrier. If the durability of the barrier is reduced to zero, it will be destroyed. The durability is according to the level of the barrier.
<u>Trap</u>:	"4" for placing the trap, the trap will cause damage to monsters if the monsters are moving on it, the power is according to the level of the trap.

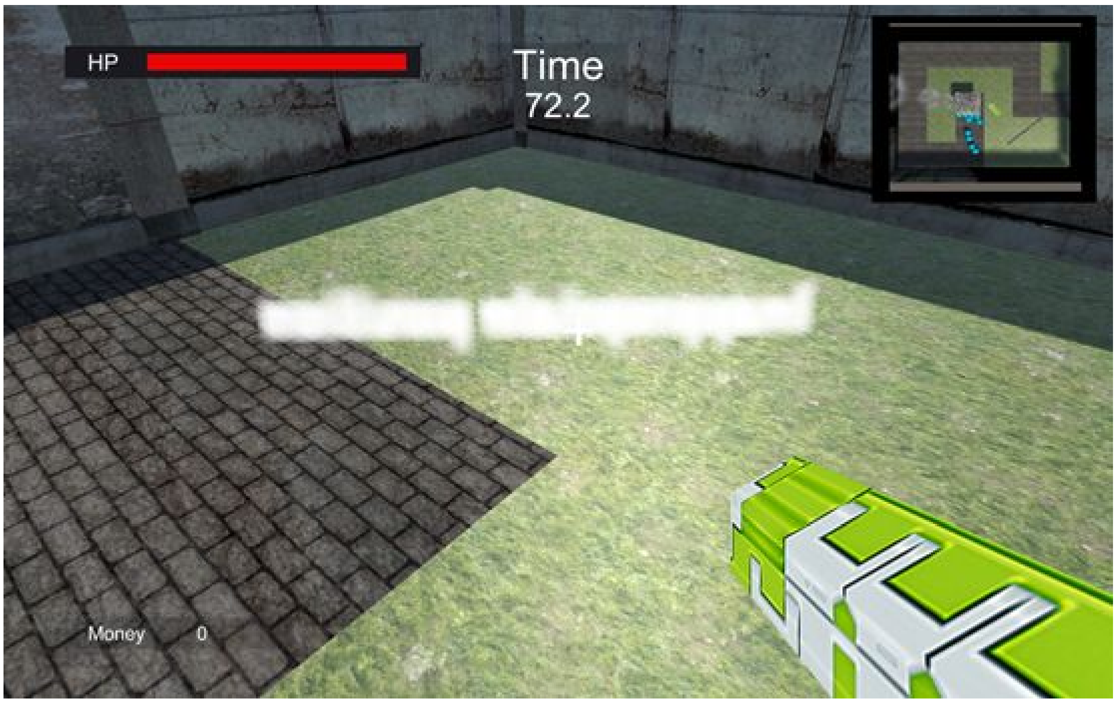
HP: The remaining health of user is shown on the top left corner
If the HP bar is reduced to zero, the game will be failed.
Time: The remaining time is shown on the top middle
If the remaining time is reduced to zero, the game will be failed.
Mini-map: The mini-map is shown on the top right corner
The mini-map shows different information, such as the route of the monsters will be passed, how the towers, barriers and Traps are placed, where the castle is and where the remaining monsters' positions are.
Money: The amount of money that the players earn is shown on the bottom left corner
Different monsters will gain different amount of money.

3.4.2 Item Placement

Towers, Barrier and Trap could only be placed on the specific region. Barrier and Trap could only be placed on the route that monsters will pass while Tower could only be placed on the position that monsters will not pass. Moreover, one grid can only has one item. Following is the correct example of placing the items.

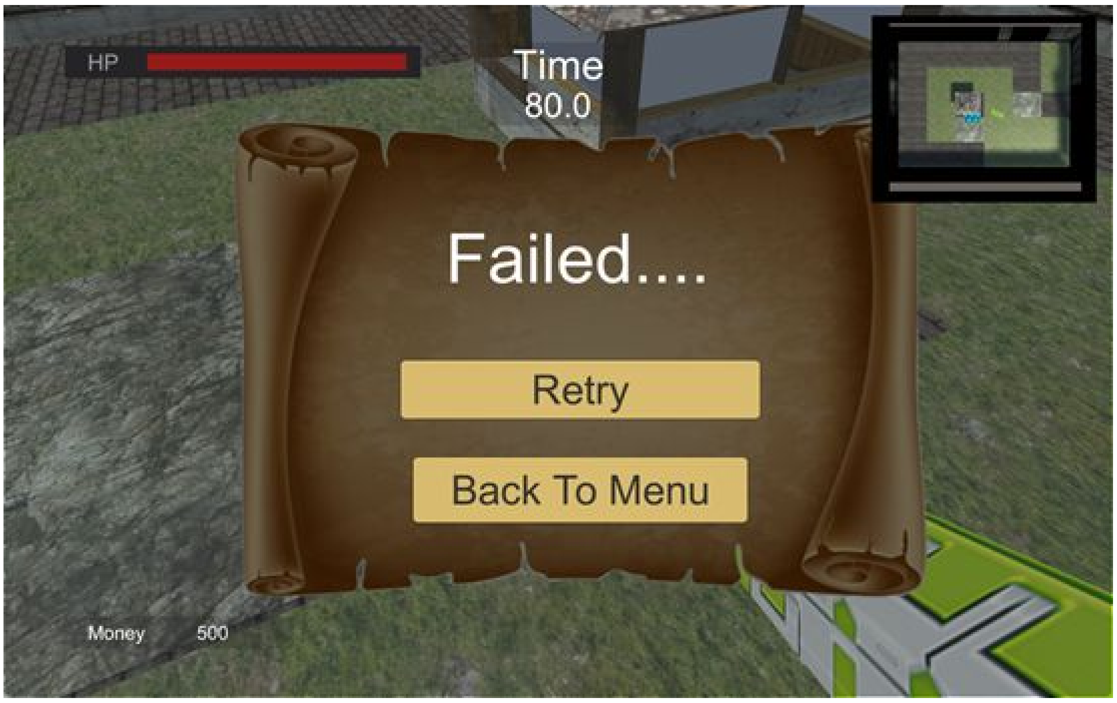


If the players place the items wrongly, an alert message will come out. This will tell the players they cannot place the item there.



3.4.3 Game Failed

The game will be failed based on these situation:
-Player health is reduced to zero
-Castle is destroyed
-Time is over

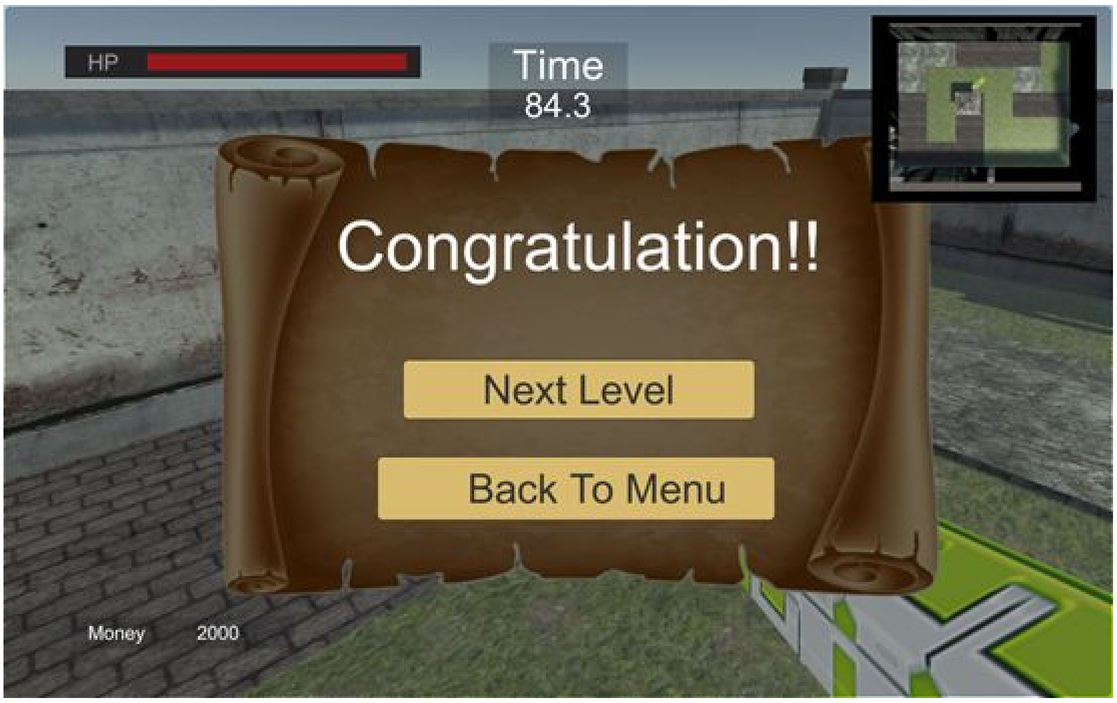


<u>Retry</u>:	To play the same level again
<u>Back To Menu</u>:	Go to start menu

If the game is failed, players cannot get the money they earn in this level.

3.4.3 Game Passed

The game will be passed when player kill all the monsters without reaching the fail situations.



If the game is passed, players can get the money they earned in order to upgrade or unlock items in shop.

4. Different to initial design

The final game is very exactly what we expected in the proposal as we have two experienced game designer in the team to design the game. The only thing different to our expectation is not enough scenes (game levels) to demonstrate our design. Originally, we hope to have more difficult scenes, such as a level with multiple path and a level with tortuous path, so that player need to use his/her intelligence to design where to put the towers/barriers/traps. However, due to time limitation, we can only build one level. Nevertheless, we added a challenge mode which is when player already finished that level once, the level will be more difficult.

5. Difficulty Encounter

In the development, we had encounter two difficulty which causing huge trouble to us:

1. Building Levels

   As mentioned in section 4 above, building levels is very time consuming. We need to design the path, adding walls to ensure player will not jump out of the world. And also, the wave of monsters also need to be design and their properties (health, moving speed, defence, etc.). Therefore, we finally can only finished 1 level with 2 difficulty (normal mode and challenge mode)

2. A Preivew Object

During the development, we had though to implement a preview object that when player chose that item, there will be an item on his hand and a preview item on the world to preview the effect of that item. However, it need to be create 2 extra prefabs for each object (1 for hand and 1 for world) which wasted too much time. Therefore, we finally throw away this idea and becomes the player can only holding weapons.