

# Advances Games Technology

## Milestone 2

### 1. Overview

“Random Tower Defence Game” is, as the title mentions, a tower defence game. The objective is to place and upgrade towers, which attack enemies that travel along a predefined path, to achieve a high score. The player’s score is used to pay for the towers and their upgrades. This requires them to find an appropriate balance between spending and saving to achieve the highest score.

This game takes place in a bedroom and where the player uses household items; hazard hats, candles and toy guns to defend themselves against their toys. The toys are toy spiders, “Claptrap” figures, mechs and “Ironman” figures.

### 2. Asset List

#### License Codes:

- CCA3 - Creative Commons Attribution 3.0 Unported License
- CC0 - CC0 1.0 Universal (CC0 1.0) Public Domain Dedication
- N/A – Creating by author (Peter Farkas) and/or no license
- PUL – Personal Use License
- FEU – Free Educational Use
- DGL – Dar Golan’s Royalty Free Music License (View license file in audio folder)

Name	Link	Date of Download	License
Wood.jpg	<a href="http://www.humus.name/index.php?page=Textures&amp;start=112">http://www.humus.name/index.php?page=Textures&amp;start=112</a>	20/11/2021	CCA3
“W Hotel”	<a href="http://www.humus.name/index.php?page=Textures&amp;start=112">http://www.humus.name/index.php?page=Textures&amp;start=112</a>	20/11/2021	CCA3
button_background.png	-	-	N/A
candle_icon.png	-	-	N/A
controls_text.png	-	-	N/A
controls_text2.png	-	-	N/A
fire_ring.png	-	-	N/A
fire1_64.png(tga)	<a href="https://opengameart.org/content/animated-fire">https://opengameart.org/content/animated-fire</a>	2/12/2021	CC0
game_title.png	-	-	N/A
green.png	-	-	N/A
grey.png	-	-	N/A
health.png	-	-	N/A

lighting.png	-	-	N/A
orange.png	-	-	N/A
path.png	<a href="https://opengameart.org/content/seamless-snow-texture-0">https://opengameart.org/content/seamless-snow-texture-0</a>	1/12/2021	CC0
“Outdoor Skybox Set”	Provided in tutorials.	-	-
toygun_icon.png	-	-	N/A
wizard_hat.png	-	-	N/A
wizhat_icon.png	-	-	N/A
“Claptrap”	<a href="https://free3d.com/3d-model/claptrap-21687.html">https://free3d.com/3d-model/claptrap-21687.html</a>	3/12/2021	PSL
Only_Spider_with_Animations_Export.obj	<a href="https://free3d.com/3d-model/spider-animated-low-poly-and-game-ready-87147.html">https://free3d.com/3d-model/spider-animated-low-poly-and-game-ready-87147.html</a>	6/12/2021	PSL
IronMan.obj	<a href="https://free3d.com/3d-model/ironman-rigged-original-model--98611.html">https://free3d.com/3d-model/ironman-rigged-original-model--98611.html</a>	3/12/2021	PSL
“Mech_F_432”	<a href="https://free3d.com/3d-model/mech-robot-f-432-354558.html">https://free3d.com/3d-model/mech-robot-f-432-354558.html</a>	1/12/2021	PSL
“Nerf handgun”	<a href="https://sharecg.com/v/60880/browse/5/3D-Model/Nerf-handgun">https://sharecg.com/v/60880/browse/5/3D-Model/Nerf-handgun</a>	28/11/2021	CC0
bbc_electric.mp3 (Edited)	<a href="https://sound-effects.bbcrewind.co.uk/">https://sound-effects.bbcrewind.co.uk/</a>	10/12/2021	FEU
bbc_gunshot.mp3 (Edited)	<a href="https://sound-effects.bbcrewind.co.uk/">https://sound-effects.bbcrewind.co.uk/</a>	10/12/2021	FEU
bbc_outdoor_short.mp3 (Edited)	<a href="https://sound-effects.bbcrewind.co.uk/">https://sound-effects.bbcrewind.co.uk/</a>	10/12/2021	FEU
Chill Jazz Hop - Dar Golan - 85bpm - 02-47.mp3	<a href="https://www.dargolan-free.com/ambient-music">https://www.dargolan-free.com/ambient-music</a>	10/12/2021	DGL
Epic Battle - Dar Golan - 128bpm - 02-02.mp3	<a href="https://www.dargolan-free.com/dramatic-music">https://www.dargolan-free.com/dramatic-music</a>	10/12/2021	DGL

### 2.1. Asset Software

All textures made by the author were made using the free software “Krita” or Microsoft paint. All textures were drawn digitally by using fill, straight line and various brush tools.

### 2.2. Additional Libraries

No additional libraries were used for this project. Only those available in the AGT\_TEMPLATE were used.

## 3. Part 1

### Intro-screen

The intro-screen has remained virtually since Milestone 1. An additional page of controls has been added using the same game-object swapping method as defined in the first report.

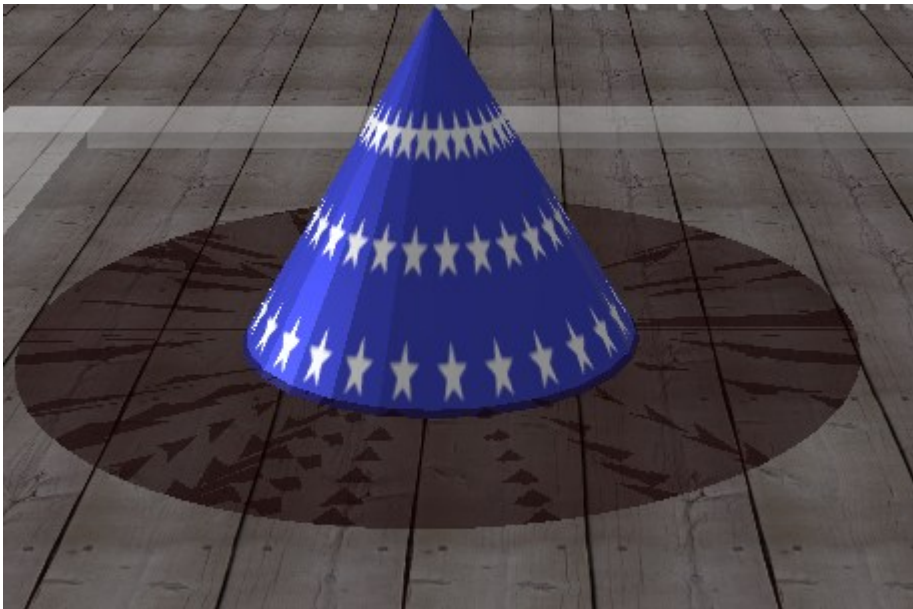
### Primitive-based objects

Three new primitive-based game objects have been included beyond the cone from Milestone 1.

The first is the circle, used to show the range of the towers in the game. This is a single layer of triangles rendered to give the illusion of a circle. To calculate all vertices, the circle is split into a given number of triangles allowing the calculation of the angle to iterate by. Then going in a circle using the following formulas the two points of each triangle along the edge of the circle can be found:

$$x = r * \sin(\alpha)$$

$$y = r * \cos(\alpha)$$



*Figure 1 Circle used to display range of a tower*

The second primitive-based object used in the game is a cylinder. The top and bottom of the cylinder is created by using the same method of the circle. The challenge comes when needing to interpret the walls of the cone. As OpenGL only draws triangles, the squares created between the edge points along the top and bottom triangle points must be split into two triangles. One is interpreted by drawing a triangle from the bottom left to the bottom right and then the top right vertices, whilst the second one is drawn from the top right to the top left and then the bottom left vertices. For this to function correctly, the applied indices must be generated in the correct order, provided and added challenge.



Figure 2 Cylinder and pentahedron

Figure 3 Code snippet show the correct generation of wall vertices and indices

The third new primitive-based object is a pentahedron. This is interpreted by mapping out four vertices for the base, drawn using two triangles the same way as for the cylinder's wall. Then from the center of said base a peak vertices arises to which a triangle is drawn from all edges of the base.

### Audio

The game contains two soundtracks, one that plays by default and another that plays during every 10<sup>th</sup> wave, when the "Ironman" enemies spawn. The volume of these can also be changed at any point by using the up and down arrow keys.

An electric spark sound effect has been implemented for the wizard hat when it unleashes its radiant lightning attack. When the toy gun tower shoots a bullet, a shorts gun shot sound effect plays. At the start of every wave, a bell chimes to warn the player of incoming enemies. This is especially useful should the player stop paying attention for a moment and need reminding of the incoming threat.

### HUD

The game's heads up display consists of multiple elements. On the right-hand side of the screen are icons depicting the different towers and their prices. This is achieved by rendering a partially translucent black quad object behind another quad object with the icon texture applied to it, which has a translucent background. Below these, text is rendered that depicts the prices of the different tower types.



Figure 4 All elements of the HUD are visible

On the top right of the screen the wave number can be found, that increments with each new wave, informing the player about their progress. To the left of this the score is displayed, which increments with each defeated enemy and decrements with each life lost or tower or upgrade purchased.

Below the score, in between waves, a timer counting down is displayed to inform the player about their time left to prepare.

In the top left corner of the screen a heart and the numerical value of the player's health can be found. The size of the heart depends on the player's health value, it will shrink as the player loses health.

At the bottom of the screen a black can be found when a tower is selected. Two smaller boxes show whether the player can afford the upgrade on a given side (when green the upgrade can be purchased, when grey the player lacks sufficient funds). Should a player have unlocked a power up, a small green rectangle will appear above this area to confirm this.

```
//scale heart appropriately
if (m_player.health() > 0)
    health_transform = glm::scale(health_transform, glm::vec3(0.01f * m_player.health() + 0.3f));
else
    health_transform = glm::scale(health_transform, glm::vec3(0.3f));
```

Figure 5 Hearth icon being scaled appropriately

#### 4. Part 2

##### Camera motion technique

A second camera type has been implemented beyond the default motion camera. When a player selects a candle tower they can press "C". This will activate the candle turret camera where the player will be stationary located just underneath the flame to avoid visual displeasure. The player's mouse movements will be tracked to control the camera's rotation. The purpose of this camera is for the player to be able to shoot fire balls at enemies that deal an area-of-effect damage when they hit the ground.

##### Mesh-based objects

5 mesh-based are used in the game. The first one is the toy gun, the only tower to not be a primitive-based object. A player can place one of these anywhere they wish, given they are able to afford it. The gun will face the target with the least remaining distance to travel that is also in its range and shoots bullets at the enemy.

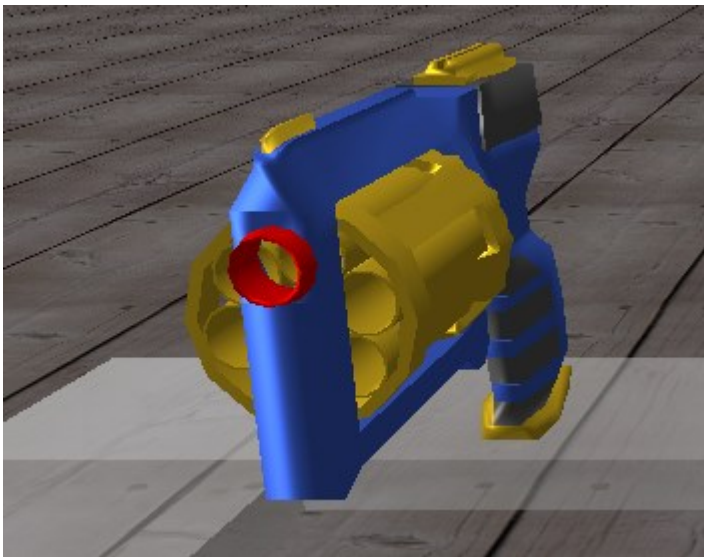


Figure 6 Toy gun model

The other four mesh-based objects are the enemies, i.e. spiders, Claptrap, Ironman, and a mech. There are four type, each with varying strength, health, and speed. These spawn during waves and moves along the depicted path to the towards the end.





*Figure 7 Spiders and a Claptrap in front of a wizard hat tower*

### **Lighting**

The game uses a directional light that has a grey color to give the game a bit of a dimmed look. This is also used to illuminate the menu screen.

Each candle tower has a point light attached to its flame, to make the flame feel more real. This effect can be seen on Figure 2 on page 4. The limit of point lights handled by the game has been set high enough to where the player should under normal circumstances not reach it.

A point light is also attached to the leading enemy highlighting them. The position of the light is constantly updated to share the x and z coordinates of the enemy with the least distance left to cover, whilst having a consistent y value of 0.1f.

```

int example_layer::lead_light_target_index()
{
    int return_index = NULL;
    engine::ref<enemy> current_target;
    //only compute if there are active enemies
    if (m_active_enemies.size() != 0)
    {
        //initialize current target to be the first entry
        current_target = m_active_enemies.at(0);
        for (int i = 0; i < m_active_enemies.size(); ++i)
        {
            if (m_active_enemies.at(i)->distance_covered() > current_target->distance_covered())
            {
                //override current target if a more suitable one is found
                current_target = m_active_enemies.at(i);
                return_index = i;
            }
        }
    }
    return return_index;
}

```

Figure 8 Algorithm used to find leading enemy

## Special Effects

Using the billboard method demonstrated in one of the example codes provided in the module, a fire effect has been added to the candle tower as well as all its fired fireballs. This technique draws all images on a sprite sheet in a given time to give the effect of a continuous motion, in this case a burning flame. This flame effect is attached to the fireball for clarification purposes as well as it being visually appealing. These flames constantly update their positions to match that of their assigned sphere. (Apologies for not provided a good screenshot, it is more difficult to make one than expected.)

When the wizard hat tower attacks a series of lightning bolts extend from its base. This is the same effect used for the fire ring when a fireball hits the ground. A horizontal texture is expanded over a given time to give the effect of a shockwave style expansion.

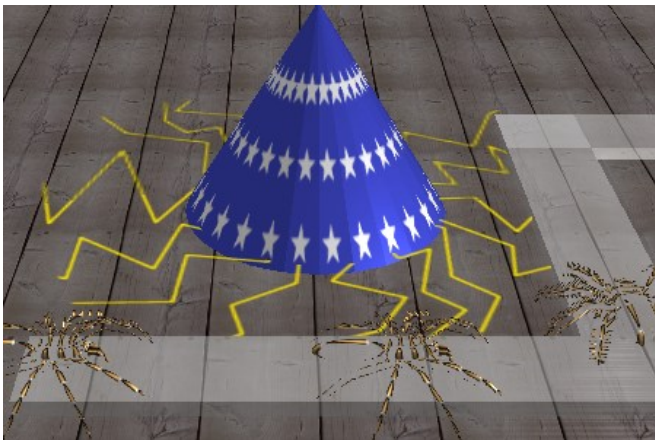


Figure 9 Wizard hat tower attacking



## 5. Part 3

### Game physics

Collisions are used a lot in this game in unique ways. There is the obvious use hit detection between enemies and bullets, but there is also the collisions detection used for selecting towers. When the mouse is ray casted, at the position along the ray with a y value of 0, a bounding box is located. When this box collides with the bounding box of a tower, this collision is then interpreted as selection, should the left mouse button be held down.

Collisions between towers, other towers and the enemies' path is also detected when placing a new tower. The placed towers are not allowed to obstruct the path of the traversing enemies and they are also not allowed to collide with each other.

The fireballs fire by the player when in the candle turret mode use physics to achieve the trajectory. When fired, they are applied a given velocity and acceleration, whilst also being affected by gravity. By applying the starting force for a short period of time rather than just once, the fireballs can achieve a natural feeling arch. This feels similar the distance travelled by an ember leaving a crackling fireplace.

An idea that was scrapped, was for the Ironman enemy to fire a healing ball up into the sky, that would, once landed, heal all enemies in the ball's range. This was left out due to weird errors occurring for which solutions were not found and ultimately also due to time constraints.

Further physical elements were avoided to not give the game an unnatural feel and look.

### NPCs

The game has four different enemy types: Spider, Ironman, Claptrap and Mech. Each of these enemies vary in stats, i.e. their speed, health, and strength values. The enemies move along a predefined path using a checkpoint system in conjunction with a state machine.

Each enemy starts of in the START state, where they travel towards the first checkpoint. Each checkpoint marks the end of a section of the predefined path. The enemy's orientation is defined by the state they are in, i.e. in which section of the path they find themselves in.

The enemies move in a straight line towards their next checkpoint, defined by their state, which in return is defined by the enemies' position. With each new wave, more and more enemies will spawn. Spiders spawn ever wave, Claptraps spawn every second wave, Mechs every fifth and Ironmen only every tenth round when the music also changes.



Figure 10 Checkpoints highlighted in red

### Gameplay elements

Probably the most prominent gameplay feature is the free placement of towers in the map. This is achieved by ray casting the mouse, i.e. drawing an imaginary ray from the mouse's position into the screen. Using a slight variation of the binary search algorithm, a point with a y value between -0.1 and 0.1 is found. Prior to placing a tower, the player selects the type of tower they wish to place by toggling them using the "1", "2" and "3" keys. The toggled tower and its range are then rendered at the calculated position, however, with a y value of 0. This is done to speed up the search for the position and due to the difference of y from being 0 is so small that such inaccuracies can be accepted.

```
glm::vec3 engine::raycaster::point_on_surface()
{
    glm::vec3 start = m_camera_position;
    glm::vec3 end = m_camera_position + m_current_ray * m_ray_range;
    glm::vec3 point = start;

    while (!(point.y <= 0.1f && point.y >= -0.1f))
    {
        point = start + m_current_ray * (glm::length(end - start) / 2.f);

        if (point.y < -0.1f)
            end = point;
        else if (point.y > 0.1f)
            start = point;
        else
            break;
    }
    return glm::vec3(point.x, 0.f, point.z);
}
```

Figure 11 Modified binary search for finding a point on the ray with a near 0 y value

Another crucial gameplay element is the upgrade system of the game. Each tower has two branches of upgrade, a left and right branch, each with two levels. When the player selects a tower, they have the option to purchase the desired upgrade by trading in a sufficient amount of their score. Upgrades include faster fire rate, a larger range and longer stun durations. As the player progresses in the game, more enemies will spawn and hence better towers will be needed.

In between rounds, a timer counts down to let the player know about the remaining time they have, to prepare for the next wave. This is achieved by incrementing a value by delta time and then subtracting this value from the time limit.

## 6. Reflection on the project

The game contains good implementations of features such as ray casting and trigonometry. Whilst the selection of towers provided to the player is rather small, the concepts behind the towers are quite unique, especially the candle tower. It stuns enemies in its range whilst the player can control the embers of the flame and shoot fireballs at enemies.

The strength of the game lies within the mechanics surrounding the towers, e.g. the maths behind their models, their placement, and general functionalities. New tower types are easy to add as their common features are handled by the tower class and only a well define subclass is required.

The game lacks however in visuals. This has less to do with to graphics of the game, but rather the feel of the scenery and the lack of an atmosphere created. Potentially some walls with textures to acts as walls could have been implemented. Maybe some other objects that resemble furniture or

other home appliances could be added to succeed with the living room floor play zone aesthetic that was aimed for.

To further expand the game, the example layer class would need to be developed into a level handler class. This would handle the switching between different levels and environments. This would also help alleviate the immense dependency the game has on that class. It pretty much pulls all the strings to hold the project together.

A more flexible HUD system would also need to be added to help develop better and more interactive heads-up display. A system of using the mouse to interact with UI elements, e.g. upgrade buttons, volume sliders, etc., is also required as to ease the controls and better the user experience. Ideally, the entire game could be controlled by solely the mouse.

The game also requires a drastic graphical overhaul from developer graphics something rather eye catching. Whilst, graphics do not matter much for interim builds and test projects, should this game be further developed to be released, a better and more consistent art style would be required.