OOP2021 期末專題報告

題目:大逃殺 3D (Battle Royale 3D)

應化 111-0712531-羅文昕(100%) 電機 1E-109511029-黃羿寧(100%) 電機 1C-109511295-王綉雯(100%)

A. 題目描述

這次的專題內容，我們打造了一款名為「大逃殺 3D」的生存遊戲。玩家需控制角色的移動以躲避敵人的追擊。本款遊戲結合多種模式、多重視角以及令人震撼的聲光效果，試圖帶給玩家刺激的遊戲體驗。

(A) 遊戲介紹：

1. Timer Mode（計時賽）：時間限制兩分鐘，若在時間內未被敵人抓到即獲勝。
2. Survival Mode（生存賽）：找到金鑰（傳送門）且途中沒被敵人抓到即獲勝，其中金鑰為隨機產生。
3. Inversion Mode（反轉大逃殺）：時間限制兩分鐘，若在時間內抓到敵人即獲勝。

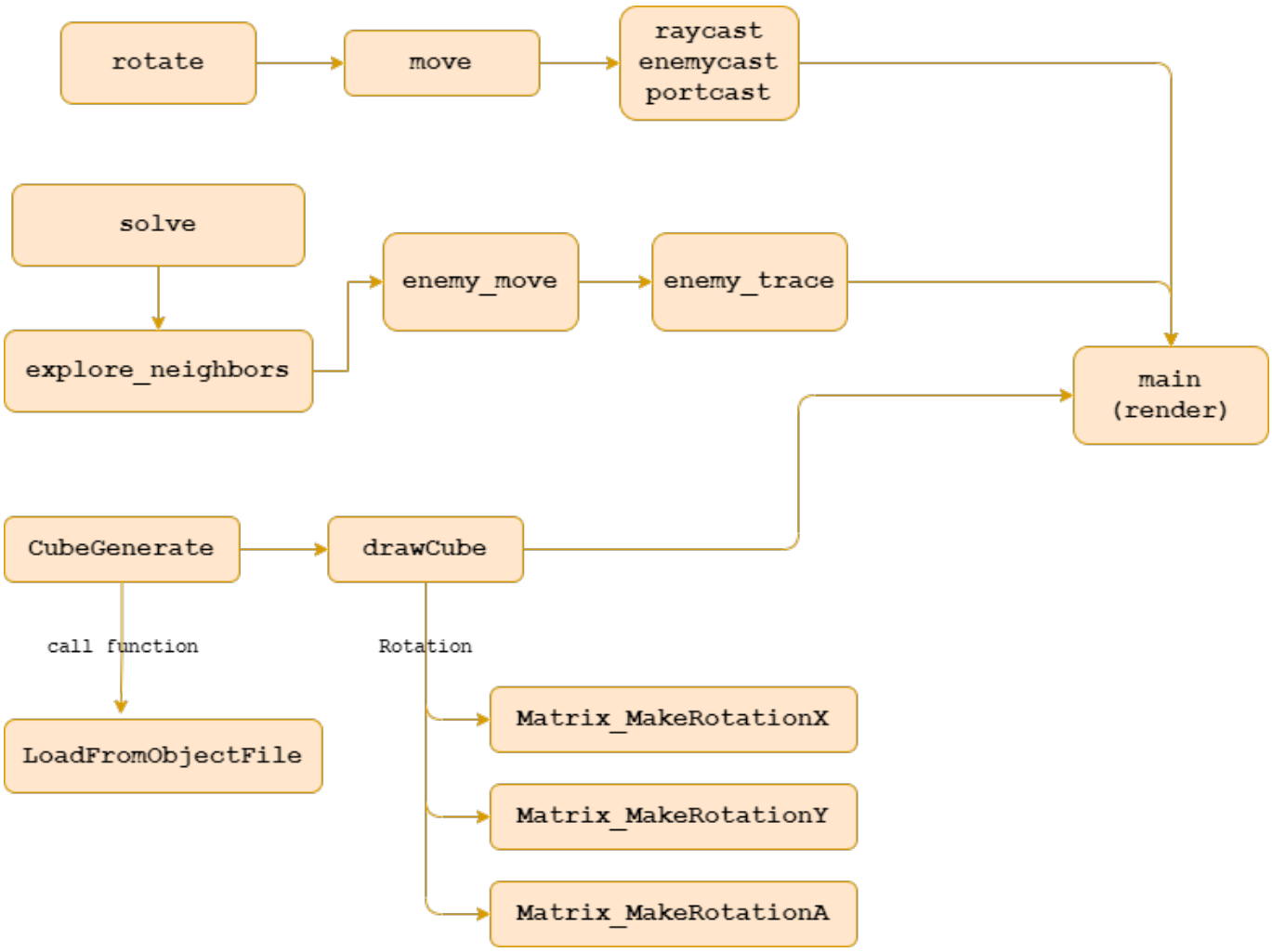
(B) 遊戲操作說明：

1. 首頁
   (1) 按 SPACE 進入主選單。
2. 主選單
   (1) 滑鼠點擊「HOME」回首頁。
   (2) 滑鼠點擊「EXIT」或按 ESC 離開遊戲。
   (3) 滑鼠點擊「TIMER / SURVIVAL / INVERSION」選擇遊戲模式。
   (4) 滑鼠點擊「SIMPLE / CLASSIC」選擇地圖樣式。
   (5) 滑鼠點擊「ON / OFF」選擇開啟或關閉音樂及音效。
   (6) 滑鼠點擊「PLAY」以開始遊戲。
3. Loading 加載頁
   (1) 按 SPACE 跳過簡易操作說明，進入遊戲畫面。
4. 第一（人稱）視角
   (1) 按 W / S / A / D 向前進 / 向後退 / 向左平移 / 向右平移。
   (2) 快速連按 W / S / A / D 可加速（連按兩下，第二下按住不放，可見明顯效果）。
   (3) 移動滑鼠改變視角(置中：向前看 / 置左：向左轉 / 置右：向右轉)。
   (4) 點擊滑鼠右鍵向後轉。
   (5) 滑鼠點擊「▶️‖」或按 P 暫停遊戲，再按一次可繼續遊戲。
   (6) 按 TAB 切換視角。
5. 上帝視角（第三人稱視角）
   (1) 移動滑鼠改變 3D 地圖旋轉方向(越遠離中心，旋轉越快，可置中以停止轉動)。
   (2) 按 O 回到所在位置的地圖面向。
   (3) 滑鼠滾輪向上可放大地圖，向下則可縮小地圖。
   (4) 滑鼠點擊「▶️‖」或按 P 暫停遊戲，再按一次可繼續遊戲。
   (5) 按 TAB 切換視角。

6. 暫停頁面
   (1) 滑鼠點擊「CONTINUE / ▶️‖」或按 P 繼續遊戲。
   (2) 滑鼠點擊「QUIT」結束遊戲。
7. SUCCESS / FAIL 遊戲結果
   (1) 滑鼠點擊「TRY AGAIN」重新開始遊戲。
   (2) 滑鼠點擊「BACK TO MENU」回主選單。

B. 程式架構

(A) 程式架構圖：



(B) 上帝視角：

1. void CubeGenerate()：將三維座標點匯入到程式內
2. bool LoadFromObjectFile(const char* sFilename)：CubeGenerate 呼叫的方程式，在方程式內打開.obj 檔並將座標點存到一個 vector 內
3. void drawCube()：將原先的三維座標點投影到二維的座標點，並存起來回到 main 印出來
4. 投影矩陣運算：
   (1) mat4x4 Matrix_MakeTranslation(double x, double y, double z)：可以將向量平移的 function
   (2) int Triangle_ClipAgainstPlane(vec3d plane_p, vec3d plane_n, triangle& in_tri, triangle& out_tri1, triangle& out_tri2)：可以將超出螢幕介面的三角形切割掉
5. 矩陣/向量運算：
   (1) mat4x4 Matrix_MultiplyMatrix(mat4x4& m1, mat4x4& m2)：矩陣乘矩陣
   (2) vec3d MultiplyMatrixVector(mat4x4& m, vec3d& i)：矩陣乘以向量
   (3) vec3d Vector_Add(vec3d& v1, vec3d& v2)：向量加向量
   (4) vec3d Vector_Sub(vec3d& v1, vec3d& v2)：向量減向量
   (5) vec3d Vector_Mul(vec3d& v1, double k)：向量乘以某值
   (6) vec3d Vector_Div(vec3d& v1, double k)：向量除以某值
   (7) double Vector_DotProduct(vec3d& v1, vec3d& v2)：向量內積
   (8) double Vector_Length(vec3d& v)：向量長度
   (9) vec3d Vector_Normalise(vec3d& v)：向量單位化(長度變 1)
   (10) vec3d Vector_CrossProduct(vec3d& v1, vec3d& v2)：向量外積
   (11) vec3d Vector_IntersectPlane(vec3d& plane_p, vec3d& plane_n, vec3d& lineStart, vec3d& lineEnd)：可以運算出線和面相交於哪一點
6. 旋轉矩陣：
   (1) mat4x4 Matrix_MakeRotationX(double fAngleRad)：沿 x 軸旋轉
   (2) mat4x4 Matrix_MakeRotationY(double fAngleRad)：沿 y 軸旋轉
   (3) mat4x4 Matrix_MakeRotationZ(double fAngleRad)：沿 z 軸旋轉
   (4) mat4x4 Matrix_MakeRotationA(double fAngleRad, vec3d a)：沿任意軸旋轉

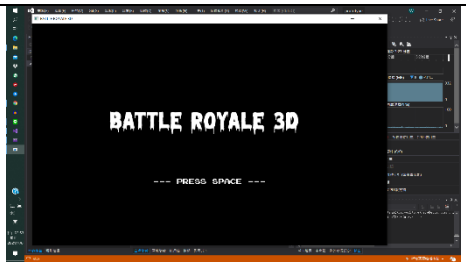
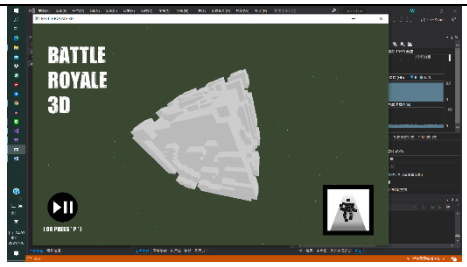
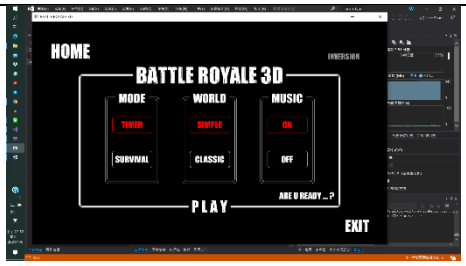
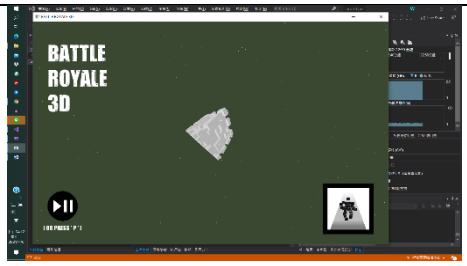
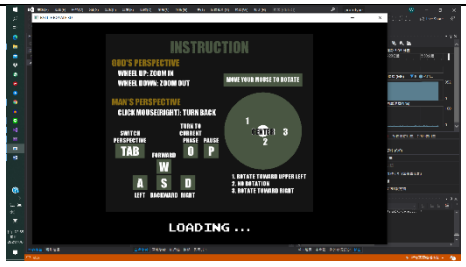
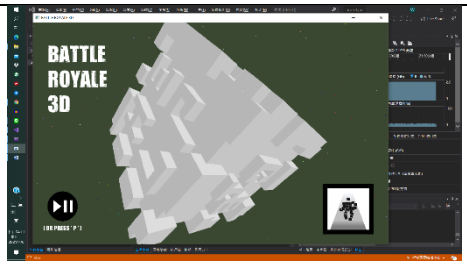
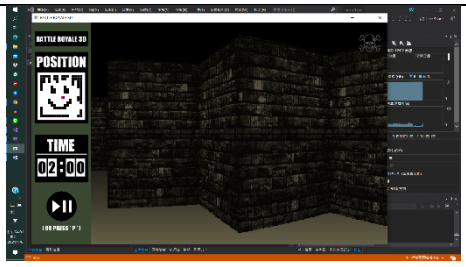
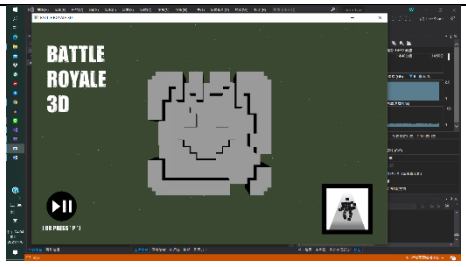
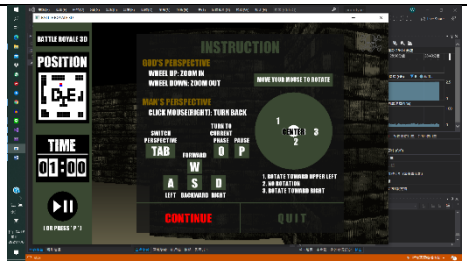
(C) 第一人稱：

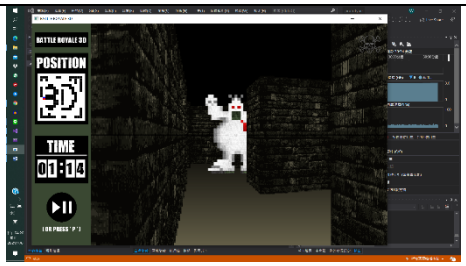
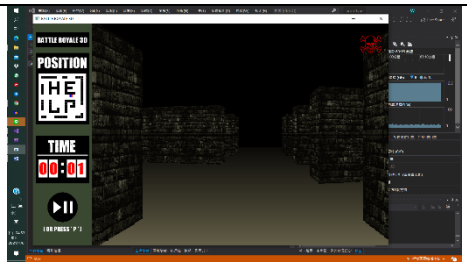
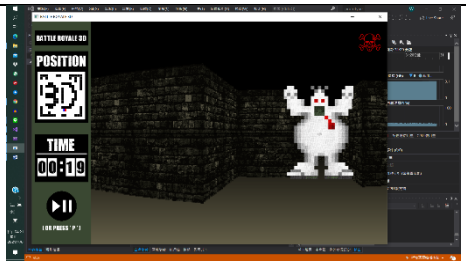
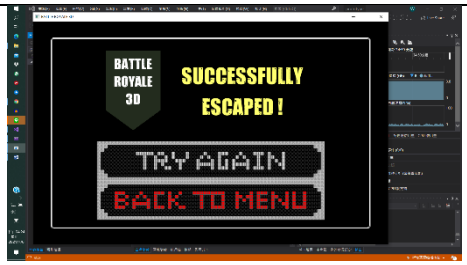
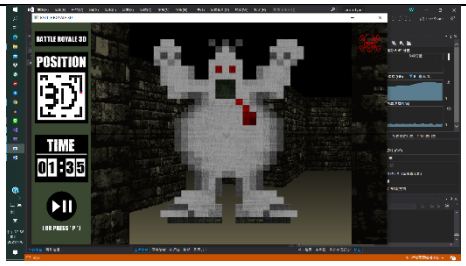
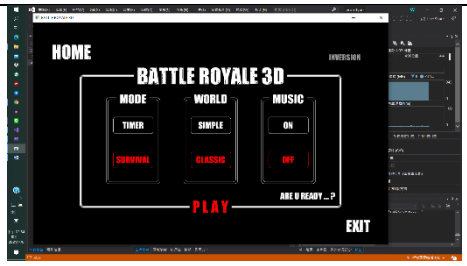
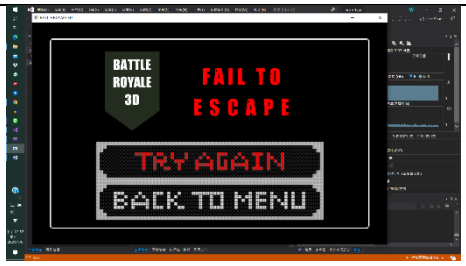
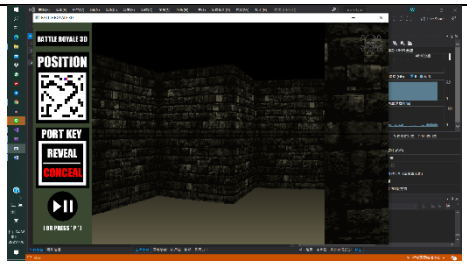
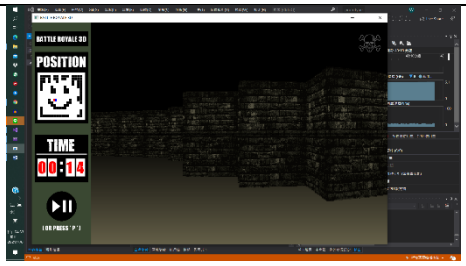
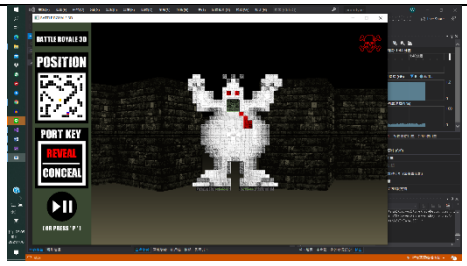
1. void explore_neighbors(int r, int c)：BFS 擴大搜尋的 function
2. int solve(int& move_count, int sR, int sC, int eR, int eC)：判斷 BFS 是否有解
3. void enemy_move(void)：尋找最短路徑
4. void enemy_trace(void)：根據路徑讓敵人移動
5. int rotate(void)：判斷使用者是否是要到下一面並做對應的面切換
6. void move()：處理使用者移動
7. void raycast(void)：處理視野內建築物與使用者之距離並計算該物件需 render 的位置
8. void enemycast(void)：計算敵人需 render 的位置
9. void portcast(int num)：計算 survival mode 傳送門需 render 的位置

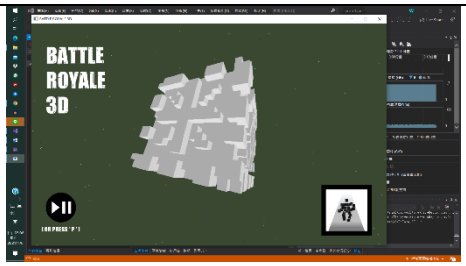
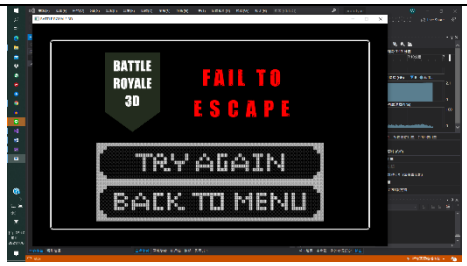
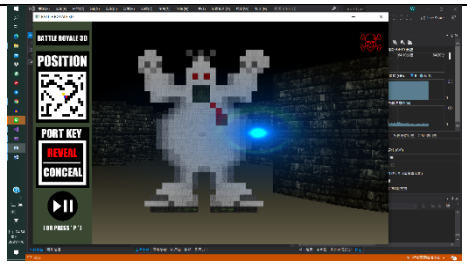
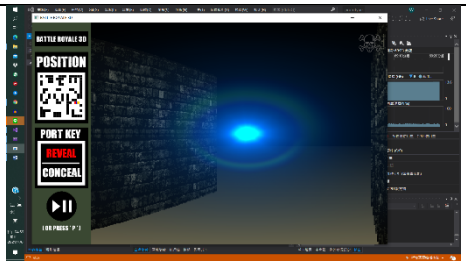
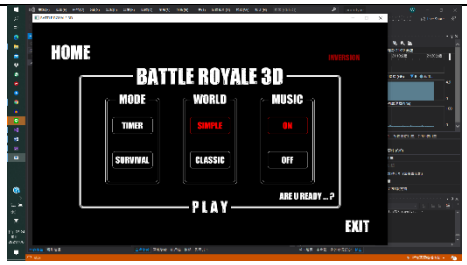
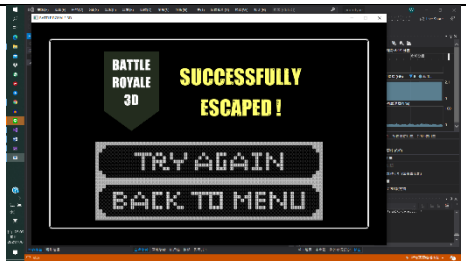
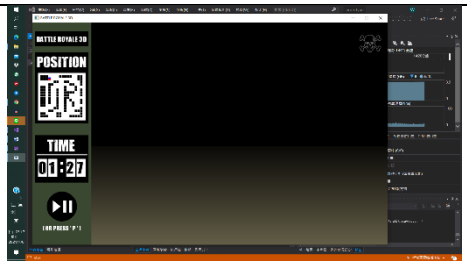
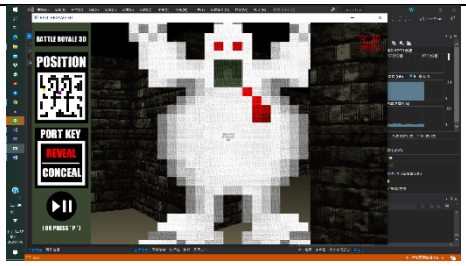
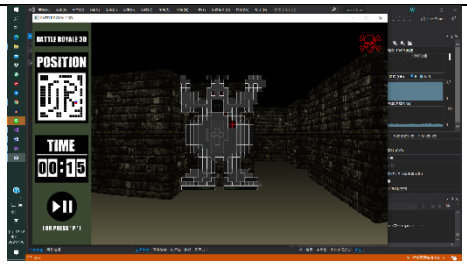
(D) Beta Version (反轉大逃殺)：

1. void enemy_check(void)：360 度檢查視線範圍內有無使用者出現
2. void enemy_run(void)：逃跑移動

C. 執行結果

1. 首頁	6. 計時賽_上帝視角_旋轉
	
2. 主選單_計時賽_簡易地圖	7. 計時賽_上帝視角_縮小
	
3. Loading 加載頁	8. 計時賽_上帝視角_放大
	
4. 計時賽_第一視角_2 分鐘挑戰	9. 計時賽_上帝視角_暫停
	
5. 計時賽_上帝視角_歸位	10. 計時賽_第一視角_暫停
	

11. 計時賽_第一視角_安全距離	16. 計時賽_第一視角_最後一刻
	
12. 計時賽_第一視角_危險警告	17. 計時賽_挑戰成功：D
	
13. 計時賽_第一視角_敵人逼近	18. 生存賽_經典地圖
	
14. 計時賽_挑戰失敗：C	19. 生存賽_第一視角_金鑰隱藏
	
15. 計時賽__第一視角_開始倒數	20. 生存賽_第一視角_金鑰出現
	

21. 生存賽_上帝視角	26. 生存賽_挑戰失敗：C
	
22. 生存賽_第一視角_暫停	27. 生存賽_兩難_前進或後退？
	
23. 生存賽_第一視角_找到金鑰	28. Beta Version_反轉大逃殺
	
24. 生存賽_挑戰成功：D	29. 反轉大逃殺_邊界風景
	
25. 生存賽_第一視角_敵人逼近	30. 反轉大逃殺_抓到敵人即成功
	

D. 專題關鍵

(A) 上帝視角：

1. 座標點匯入：為了能夠在電腦上呈現出 3D 世界的樣子，我們利用 blender 先繪出世界的樣子，再將三維的點座標匯出成 obj 檔，因為匯出的檔案可以以文件檔的方式打開，所以我們就想到利用老師先前教過的讀寫檔案的 function 將那些座標點從文件裡取出存到程式需要的 vector 當中。
2. 3D 投影公式：將三維座標點與投影矩陣相乘，算出二維座標點。
3. 任意軸旋轉：原先我們的 3D 世界只能沿著 x 軸和 y 軸旋轉，但我們希望能夠加強它旋轉時的流暢度，所以希望滑鼠移到哪，世界就旋轉到哪，因此我們從網路上找到任意點的旋轉公式，將螢幕上的滑鼠座標換算到三維的空間中，再取一條垂直於此點到世界中心連線的軸線，當作是旋轉軸。

(B) 第一視角：

1. Raycasting：Raycast 顧名思義就是利用掃描的方式，我們先將視線範圍內的東西沿 x 軸切成很多等份，再一一去掃描使用者到牆壁的距離，距離越遠的在視覺上天花板越矮、地面越高，因此牆壁高度越短，相反來說，越近的牆壁高度越長，依照此算法，我們就可以將一個二維的陣列呈現出三維的感覺。
2. 牆壁貼圖：因為 SDL 無法將圖片壓縮變形成我們想要的形狀，所以我們只能將圖片分割成細碎的長方形，再用 SDL 內建的圖片縮放方程式將圖片在螢幕上呈現。越遠的牆壁，因為較不明顯，所以圖片可以切的不那麼細，但越靠近牆壁，尤其是貼近牆壁時，螢幕上被分割的每一格就需要將圖片局部切的越細，被分配到的圖片大小就越小。
3. 敵人追蹤：因敵人追蹤需採取最佳路徑，故採用 BFS 演算法，由於每一面的地圖皆是由陣列製作，故可明確知道牆及路的位置，藉由 BFS 產生敵人追蹤 user 的路徑，並將其移動的方式修正為連續移動，使整體遊戲較有真實感。

(C) Beta Version (反轉大逃殺)：

這個模式將敵人模擬為另一個 user，賦予視角角度（也就是其前進方向），使其具有觀察四周的能力，其觀察功能是從 raycasting 方法延伸，若在他視線範圍固定距離內偵測到敵人，會先從與 user 同向正負 90 度範圍內篩選逃跑路線，但以上方法由於會有死角問題，故又植入象限與相對位置分析，若是遇到死角，會再擴大角度篩選範圍，並判斷最佳移動方向。剩餘的待處理問題：敵人在死角會因為角度問題而無法逃脫，需給予更多情境判斷，例如：象限、座標。

(D) 其他：

遊戲角色、配件跟版面設計以及音樂與音效的搭配，也是我們在此專題中花費了許多心力的部分。另外，如何透過多次且全方位的測試，以達到最棒的使用者體驗，也讓我們費盡苦心。

E. 心得感想

(A) 羅文昕：

我覺得自己非常幸運，能在做專題的路上遇到兩位願意一起做大夢，然後再一起實踐夢想的組員。從第一次約討論，到最後一次整理專題報告，我們幾乎都是一起在做事。雖然大家忙的事情不相同，但過程中能隨時有人給予回饋真的是非常棒的經驗。也因此，即便我們常常在 Demo 前大改內容，也總是能咬著牙撐過去。回想一開始因為 3D 建模，進度相對其他組落後許多，當時真的超級緊張，幸好後來我們靠著日夜接力，總算完成相當滿意的成果。未來回想起這段時光，肯定是充實快樂且幸福的吧！

(B) 黃羿寧：

我覺得這次專題就是一個不斷在挑戰自己能力極限的過程，當初看到學長姐專題範例時就覺得做 3D 的組別很厲害。但助教在教 SDL 時不會帶到那麼難的部分，所以在一開始光是要在電腦螢幕上呈現出 3D 的樣子就花了很多時間和精力在搞懂其中的原理，所幸沒有在一開始就放棄 3D 的想法換成別的主題，而是撐到最後完成了現在大家看到的遊戲，且竟然能夠拿到第二名，真的很開心。這堂課上我獲得了很大的成就感，且在這個過程，我學到了很多的程式技巧，相信我以後再接觸到三維的程式設計可以更得心應手。我覺得有時候題目定的高一點可以強迫自己要學更多的東西，在這個過程其實也很充實且有趣，隊友選得好也很重要，這個專題的完成是我們三個人的力量，真的很謝謝他們！

(C) 王綉雯：

這次的專題製作非常有挑戰性，接觸了許多以往不知道的領域，在找資料的過程學習到非常多東西。雖然整體專題還不致完美，但對於能在這麼短時間內完成這些目標感到非常感動；另外，非常感謝我的組員們都有盡責地完成彼此該做的事，讓分工流程能順利進行，也非常謝謝他們對我的包容，總的來說，能完成此次專題並獲得大家的賞識我感到很榮幸。

(D) 總結：

對於大逃殺 3D 未來的升級與擴充，我們還有幾個想法，例如：角色選擇與互換、雙人互動、連線遊戲等。我們想闡明的是我們創作的這個遊戲並不是死的，還有未來發展的可能性與實際達成的可行性，只是現階段我們只能測試至反轉大逃殺而已。總而言之，我們希望賦予這個專題一個未來發展的可能，並非一個做完就「徹底結束」的遊戲。

F. 參考資料

(A) YouTube 頻道：

1. 3D：
   (1) https://reurl.cc/dGab9V
   (2) https://reurl.cc/MAzm9L
   (3) https://reurl.cc/EnQNW0
2. 第一人稱：
   (1) https://reurl.cc/yEAbKa
   (2) https://reurl.cc/vqLbEy
3. BFS：
   (1) https://reurl.cc/ze5G7y
   (2) https://reurl.cc/4aNDGD
4. 任意軸旋轉：
   (1) https://reurl.cc/LbQgGx
5. 總之，大推特推 javidx9！

(B) 音樂與音效：

1. https://mixkit.co/free-sound-effects/game/
2. (menu) Two Steps from Hell_Final Days of Rome feat Felicia Farerre
3. (timer-man) Two Steps from Hell_Birth of a Hero Classics
4. (survival-man) Epic Dark Battle Music – Escape
5. (god) Two Steps from Hell_Victory
6. (alert) Siren Alert Alarm sound in the WAR
7. (pause) the Black Waltz
8. (success) Good Luck in the Coming Wars
9. (fail) 007 James Bond Theme