

**CS4555 Introduction to 3D Game Programming**  
**Fall 2018**  
**HW2 – NPCs for a FPS (20%)**

- Submission deadline: 10/25 Thursday, 9AM
- What to submit:
  - Submit only script files via CSNS. File names should be self-explanatory. Each script must have a short description.
  - Submit your game document via CSNS. See the [game document template](#) and complete it as requested.

Every team will be peer-evaluated on Thursday, 10/25/2018. Also, several teams will be selected randomly and evaluated by the instructor. Unselected teams for the instructor evaluation must schedule to visit the instructor for grading.

Like HW1, your reference or inspirational game is Goleeneye 007 FPS. Watch a gameplay clip at [https://www.youtube.com/watch?v=dW\\_ElQJ6eRA](https://www.youtube.com/watch?v=dW_ElQJ6eRA). Then, design your game (background story and game mission) and document it using the [template](#).

For this homework, your primary focus is to implement refined player controls and enemy actions. Do pay less attention to visual effects for now. GUI implementation is not required.

- ✓ Enhance your game world.
  - Make your game world visually appealing and suitable for your game theme. Use realistic texture images and lights.
  - Add raised platforms where characters (the player and enemies) can move up (down) and jump.
  - Maintain several portals where the player can transport to another location. An enemy can chase the player very closely but it cannot transport through a portal.
  - Place big and small obstacles (e.g. crates) in your game world. These obstacles will NOT react (moves/rotates) to external forces for now.
  - Place several pick up items (e.g. ammo). Tie these items with the game mission.
- ✓ Player:
  - In addition to looking around, walking and shooting, the player must be able to run and duck (lower head and body quickly to avoid being hit). Provide seamless transitions among these moves.
  - Display the player's gun. The gun should not be warped under a certain view or be clipped by any objects. As requested by HW1, two weapons should be available for the player and the player should be able to choose/toggle a weapon using the right mouse button.
- ✓ NPCs:
  - Use humanoid-like model with animations for all NPCs. You are allowed to use Pathfinding library for NPC's moves. Design well-balanced enemy characters so that the game does not become too boring or impossible to win.
  - Enhance your NPCs as requested below:

- Enemy 1 – Modify your current Enemy 1 to act more intelligently. This character must be able to idle, walk, run, shoot, **jump**, and chase to attack and avoid the player. This enemy can sense the player only when the player is nearby and within his view angle. Define its attack strategy further. Modify fireballs (e.g. use bullets) so that the attack looks more realistic.
- Enemy 2 – Like HW1, this enemy will not chase the player. It always stays around and hides behind an object (e.g. crater). When this enemy detects the player in its vicinity, this enemy must look at (turn to) the player and shoots the player. Also, this enemy will alert other enemies around it to attack the player. Define its attack strategy further.
- Bystander – This character doesn't do anything. This character runs away if it sees the player (i.e the player is in this character's line of sight). Define its escape strategy.

✓ Scene:

- When any NPC gets destroyed, spawn new NPCs around the player. The total number of NPCs in the scene should not exceed 10. Define your spawn strategy (spawn speed, enemy types, and so on).
- Up to 3 enemies can attack the player at a time.
- Handling collision among enemies is not required.
- Provide a keystroke for game "reset". The game will restart with the initial setting for the player, enemies, and the environment. Do NOT use Unity's scene control functions (such as LoadScene) for this.
- Implement the "Game Over" based on the losing condition that you defined in your game doc. The program should stop processing player moves, player shooting, and enemy's attack. Player's rotation and enemy's moves must be allowed.