Concept: Our game fits within the beat-em-up genre, more specifically the hack and slash sub-genre, a type of action game where the player must fight a large number of enemies and proceed through scrolling levels when they beat the enemies. Yes, our game does meet this expectation, the player will be put against a group of enemies in a room and proceed to the next level or floor once the enemies are defeated. Our concept is novel because our game is high score-based instead of story-based like most other games in the genre. Our theming is also novel, as there are not many medieval-themed games in the 3D beat-em-up genre. The game does not add anything new to the genre.

Visual and Artistic Element: Yes, the game is visually appealing in terms of balance in artistic elements, models, and how they move. The theme of our game is medieval, both character models and the environment will match the theme with visual colors that are appealing but not bland or distracting. The movement is also meant to be snappy and quick, with a wide over the shoulder 3rd person camera for a wide view of the playing field. The animations are meant to convey this snappy feeling by looping through each animation very quickly, and being programmatically controlled as opposed to being controlled by the Animator, since the animator chaining has known issues with playing multiple animation states at once.

Do the sound elements (effects and background) enhance the experience?

Sound has not been implemented yet, but for the core experience, sound design is one of the most important aspects of action games, especially high octane genres like beat-em ups. A moody, subdued soundscape was planned but was not implemented yet.

Does the game incorporate complex visuals (object animation, visual effects, movement in lighting and shading, significant use of parent/child object relations)?

No except for animations, since this is a gameplay focused genre, so emphasis was placed in designing the movement, animation and combat.

Algorithmic complexity: The enemy AIs uses Unity's AI engine for chasing the player, and the Unity physics engine for physics interactions. The combat is based on original, nontrivial code. The character and its every interaction with the world is handled by a State Machine. This state machine script handles the different states of action for the player, like calculating the movement of the player on input, which uses an Input System controller to assign actions to keys and reference these actions in an input handler script referenced by the State machine. An event based system allows inputs to invoke events that are subscribed/unsubscribed actions in the relevant state. For the movement, the input is sent as a Vector2 value to the state machine and handled in the player's relevant state. All inputs have Mouse and Keyboard support as well as Gamepad.

Gameplay: We planned to create the flow effect by upping the difficulty of the game as the player has less health and the enemies move at a good pace towards the player. The flow effect comes from managing both the player's health and enemy numbers to avoid being overwhelmed. We planned to have a state handler in the background that handles spawning the enemy at random intervals and spawn hubs, and also keeping track of how many enemies have been spawned and are alive. As levels progress, thresholds for how many enemies to spawn, the frequency of spawn, the difficulty of the enemies (handled by the Stage State Machine, adjusting movement speed and attack speed and health). For combat, the player is meant to be able to queue a combo string by pressing the attack button closely following the previous press of the button. The player has two enemy detection sphere colliders, one to detect a target to focus on using a targeting mode for the Cinemachine camera, and one to detect the enemy within an actionable distance from the player. Both spheres store an object with the Target script attached to them in their separate lists and the innermost sphere is supposed to destroy the enemy object after accumulating a number of iterations of its WasHit counter, which is incremented every time an attack is initiated by the player. This WasHit counter is meant to represent the health of the enemy.

Completeness: Our game is not in a totally playable place yet. Character and enemy modeling, animation handling, input handling, state handling, enemy AI, and camera has all been mostly completed. Building the tools for the game has taken all our time, so a full game wasn't possible to make in time. There are a few more tools we would like to make before finalizing the tools creation aspect and moving on to game design. We weren't fully able to commit to game design without a clear idea of what tools we would realistically have and can make.

Intangibles: N/A