

Climb of Duty 2: Scrolling Warfare

Authors: Colin Lou, Liam Berman, Sanatan Mishra

Revision: 05/26/23

Introduction:

Our program is a successor to our previous game, Climb of Duty. Although part of the same series, Climb of Duty 2 is a completely new and updated game coded from the ground up that takes the core mechanics of the original Climb of Duty and elevates them to an APCS level. These core mechanics include vertical scrolling, platforms, and enemies.

The game's story picks up on the last game. To win in the previous Climb of Duty, the player needed to reach outer space and go through a portal. This portal teleported the player into a new dimension with a military theme and realistic sprites. Unfortunately, it was not only one player that jumped through this portal as the second player also went through.

In Climb of Duty 2, players once again need to compete to reach the top of the map but now players have weapons that they can shoot at each other to lower their health. In the previous Climb of Duty, there was no such thing as health, however in Climb of Duty 2, each player has a health bar displayed at the top of their model. There will be many enemies that may damage a player as well as opportunities to regain health. Enemies will follow a model which predicts the player's position so that it can position itself and attack accordingly. When a player runs out of health, the player will be taken back to the bottom of the map.

The goal of the game differs with the mode the user chooses. In solo mode, the user competes against itself with the goal being to reach the top of the map in the least amount of time. In the multiplayer modes, the goal is to reach the top before your opponent does. Our vision is for you, the player, to have a fun time whether it is by yourself or with others.

Instructions:

Player 1:

W - Move Up

A - Move Left

S - Fast Fall while Airborne, Dodge while Shooting

D - Move Right

Space - Shoot

Player 2:

Up Arrow - Move Up

Left Arrow - Move Left

Down Arrow - Fast Fall while Airborne, Dodge while Shooting

Right Arrow - Move Right

NumPad 0/Shift - Shoot

There will be a main menu screen where the user can specify the game's difficulty as well as modes (solo, local multiplayer, online multiplayer)

Features List (THE ONLY SECTION THAT CANNOT CHANGE LATER):

Must-have Features:

- As it is a climbing game, Climb of Duty 2: Scrolling Warfare will require vertical scrolling between levels. This means that the field of view of the player changes with respect to the player.
- In our game, we will need two players to create an atmosphere of competition. So, it is necessary that we have a two-player setup.
- To make the game more interesting, we will need antagonists in the game. In this case, we will have enemies. At the lower levels, the enemies will only harm the player if the player touches them. At higher levels, however, the enemies will harm the player by shooting them as well.
- As an improvement to the previous installment, players will have a set amount of health that will drop down as players are shot or touching an enemy. The health will be displayed above the player's model.
- We want Player-vs-Player combat in Climb of Duty 2: Scrolling Warfare, where one player is able to shoot at the other. This makes the game engaging as it increases competition between players.
- Our game will need a scoring mechanism, in which the distance the player has climbed is their score. The score is cumulative until the player beats the game, meaning the more the player falls down to the ground and comes back up, the more points they earn.
- Our game will have an interactive menu with buttons to start the game, choose to play solo, local multiplayer or online multiplayer.

Want-to-have Features:

- While having a two-player game is interesting, it is rather inconvenient to have two players play on the same device. So, it would be nice if the game would have online multiplayer capabilities.
- Another feature we would like is AI enemy behavior. We would like to "predict" where the player will move so that the enemies can figure out where it would be best to shoot and where it would be best to move, depending on the level.
- One feature that would be interesting to have in the game is a count of how many platforms the player has climbed and how much farther they need to go. This count will simply be the number of platforms left.
- It would be nice to have a difficulty slider, which will allow the players to determine how difficult they want the game to be. It would simply change the distance between the levels and the number of obstacles.
- Another feature that would improve the game would be power-ups. They function just like obstacles, but instead of sending the player back to the start, they allow the player to pass through them.



Stretch Features:



- One feature that would be in a fully complete version of this game that we probably will not be able to implement in this time frame is 3-dimensional graphics. Each player, obstacle, and platform must be in 3D instead of 2D.
- Another such feature would be visual post-processing with animations, where each character is animated.
- One feature that would look good but is dependent on having 3D graphics is having ray-traced graphics. This is nearly impossible to implement in time.

Class List:

- Player - Represents a player in the game
- Enemy - Represents all Enemies in the game
- SmartEnemy - Extension of Enemy with intelligent behavior
- PowerUp - Generic powerup that increases health
- TeleportPowerUp - Teleports player up one level when collided, along with increasing the player's health
- Camera - Controls the translation and scaling of the game panel
- GameObject - Represents all objects that live in the Game
- Main - Adds the PanelHandler
- PanelHandler - Handles what panels are drawn to the JFrame
- MenuPanel - JPanel for the Menu
- Bullet - Extension of GameObject that represents all bullets
- Button - Represents the buttons used in the Menu Panel
- CycleButton - Represents buttons that can cycle through
- GamePanel - JPanel for the Game with Loop
- Platform - Represents all platforms in the game

Credits:

- Sanatan Mishra – Enemy, SmartEnemy, Platform, PowerUp, GameObject, and GamePanel classes
- Colin Lou – Camera, Player, TeleportPowerUp, PanelHandler, and Main classes, all artwork
- Liam Berman – Button, MenuPanel, and CycleButton classes
- Jay Kamat – JayLayer library
- Clifford Harris, Lavell Crump – Theme song
- nBeats –  Sound Effects Of Fireworks | 10 MINUTES | High Quality Audio (Fireworks sound effect)
- BBC – <https://ichef.bbci.co.uk/images/ic/480xn/p05lqc4f.jpg> (Fireworks image)
- Vittorio Romeo –  Getting The Game Loop Right (algorithm used in GamePanel class)
- Sprites
 - https://as2.ftcdn.net/v2/jpg/01/36/82/05/1000_F_136820539_COZwuSEMHAQLxAeWKXOd3Kg8coPUkMHm.jpg
 - <https://i.pinimg.com/originals/76/19/b0/7619b0e13e3c1b10dcdd4a13cfd02ee.jpg>

- <https://images.fineartamerica.com/images/artworkimages/mediumlarge/3/pixel-art-moving-flame-andrea.jpg>
- https://encrypted-tbn2.gstatic.com/images?q=tbn:ANd9GcSG8Jlc4pZeXoNjBm_HpmDlS1yWaeKLI_6BPzIT09tSLRz97Ca8
- CPhT -  Player Grunt Sound Effect First Variation
- Gaming Sound R6 -  Sonic Ring - Sound Effect (HD)
- <https://pixabay.com/sound-effects/9mm-pistol-shot-6349/>
-