Name:

Group 6 (on Canvas)

Team Members:

List all members and their student IDs:

- Carson Bennett (18726513)
- Aakash Tirathdas (19000413)
- Tawana Ndlovu (78438512)
- Jan Suratos (62384391)

GitHub Project Repo: https://github.com/sjanr/final_project_group6

Original Concepts, Ideas, and Milestones (Updated Version Below)

Core Concept

Briefly describe the game.

- We decided to use ice climbers as a base game. The game objective is to climb up a mountain with a hammer. The hammer is used to assist movement upwards.
- Our unique twist. We added 3 different weapons that can be rotated and used for different purposes. Other than the general hammer each weapon has a cooldown. Another small twist is that instead of a given timer we will be using water rising from the bottom.

Example: Our game is inspired by Breakout, but instead of a paddle at the bottom, the player controls a magnet that can attract or repel the ball, adding a physics-based challenge.

Core Gameplay

Game Loop:

Describe the basic actions the player takes and how they interact with the game.

- The goal is to reach the top of the mountain using the items present. Defeating enemies will give bonus time to get up the level. The core action is jumping up on platforms, hitting enemies with your hammer and breaking blocks above you.
- If an enemy successfully hits you, you will be knocked down 1 platform.
- The game provides feedback through visual and sound effects. Each action has a unique visual and sound effect. As time reduces water rises from the bottom. You lose the game if your player touches the water. When defeating an enemy, the water will stop moving for a given time.

Example: The player moves a spaceship left and right to avoid asteroids and shoot enemies. Destroying enemies grants points and occasional power-ups, making the ship stronger.

Player Controls

List the controls clearly and simply.

- Keyboard (or Controller) Inputs:
  - Move: Arrow Keys / A & D
  - Jump: Spacebar
  - Attack/Shoot: Left Mouse Button
  - Change item: E
  - Pause: ESC

Level & Progression

Game Structure:

- There will be 10 levels. The first 3 levels introduce the weapons and its uses. All 10 levels must be completed with the 3 lives given at the start. Loss of a life send you back to the start of that level.
- Levels will be hand crafted (If we find a way to procedurally generate in the style we want we will switch to it.)

Progression System:

- More obstacles and faster enemies. The initial time for water to increase will stay the same. But there will be more enemies to stall the water.
- We plan to add 2 new weapons and tweak the existing weapon from the original game.

Example: Each level introduces new enemy types and increases their speed. The player can collect shield power-ups for temporary invincibility. There will be 3 levels in total and the player has to win in all 3 to finish the game

---

Scoring & Win/Loss Conditions

Winning:

- Beat the final level without losing all their lives

Losing:

- When the player falls into the water (touches the water) 3 times. It is a time and life-based game.

Score System:

- This is based on how low the water is when a level is complete.
- There is a bonus for completing each level without losing a life.

Example: The player earns points for defeating enemies. A combo multiplier increases score for consecutive hits. The game ends when the player loses all three lives.

Old Timeline & Milestones

Week 1: Core Mechanics & Gameplay Elements I ✔️

- Set up basic Unity project ✔️
- Implement player movement (walking, jumping, climbing mechanics) ✔️
- Implement weapon mechanics (hammer, switching weapons, cooldown system) ✔️
- Implement enemy behavior (movement, attack, knockback system) ✔️

Week 2: Core Mechanics & Gameplay Elements II

- Create water rising mechanic and loss condition ❌
- Implement basic scoring system (water height bonus, enemy defeat bonus) (implemented differently)✔️
- Create rough UI (score display, health, weapon cooldown indicators) ❌
- Add basic sound effects for movement, attacks, and enemy interactions ✔️

Week 3: Level Design & Progression

- Design and implement 10 levels with increasing difficulty (only did 2)✔️
- Introduce tutorial levels for each weapon mechanic ❌
- Balance enemy placement, attack patterns, and movement speeds ✔️
- Refine water rising mechanic and add pauses when enemies are defeated ❌
- Test and tweak difficulty progression across levels ✔️

Week 4: Polish & Analyze ✔️

- Add particle effects for attacks, enemy defeat, and environmental interactions ✔️
- Improve animations for the player, enemies, and environment ✔️
- Enhance UI with improved visuals and animations ✔️
- Add background music and finalize sound effects ✔️
- Implement main menu, pause menu, and game over screen ✔️
- Conduct final playtesting and bug fixes ✔️
- Package the final build for submission ✔️

Current Features and Changes Made to the Game

Even Though we were supposed to follow the 2 week timeline as a group in the 2 weeks we had we structured our progress similar to the initial 4 week plan just more intense. Because of that we used it as a checklist in a way instead of our 2 week plan.

New Timeline & Milestones Week 1:

Core Mechanics & Gameplay Elements

- Set up basic Unity project
- Implement player movement (walking, jumping, climbing mechanics)
- Implement weapon mechanics (hammer, switching weapons, cooldown system)
- Implement enemy behavior (movement, attack, knockback system)
- Create water rising mechanic and loss condition
- Implement basic scoring system (water height bonus, enemy defeat bonus)
- Create rough UI (score display, health, weapon cooldown indicators)
- Add basic sound effects for movement, attacks, and enemy interactions

Week 2:

Level Design, Progression, Polish & Analyze

- Design and implement 10 levels with increasing difficulty
- Introduce tutorial levels for each weapon mechanic
- Balance enemy placement, attack patterns, and movement speeds
- Refine water rising mechanic and add pauses when enemies are defeated
- Test and tweak difficulty progression across levels
- Add particle effects for attacks, enemy defeat, and environmental interactions
- Improve animations for the player, enemies, and environment
- Enhance UI with improved visuals and animations
- Add background music and finalize sound effects
- Implement main menu, pause menu, and game over screen
- Conduct final playtesting and bug fixes
- Package the final build for submission

What we implemented

Due to the time we expected to have and what we do have (4 weeks to 2 weeks) we focused on having a working game loop with minimal features that work properly.

Weapons

Hammer- This is used to break blocks directly above the user. It can only be used to break blocks and not damage enemies.

Launcher- This is the weapon used to damage enemies. It shoots out a projectile that travels over a set distance.

Enemy types

Standard enemy - He moves back and forth over the platform. He has 3 health. If he runs into contact with the player the player dies.

Long range enemy- He shoots projectiles which deal damage to the player. The enemy it-self does not move. He has 5 health.

Checkpoint enemy- He guards the entrance to the next level.

Win case and loss case

Due to the lack of time we simplified the score process and death conditions.

The main goal now is to make it to the checkpoint before time runs out.

You can die either by time running out or by getting hit by an enemy.

The score is calculated by the sum of time remaining at the end of each level

Game Flow

There are currently only 2 levels in the game.

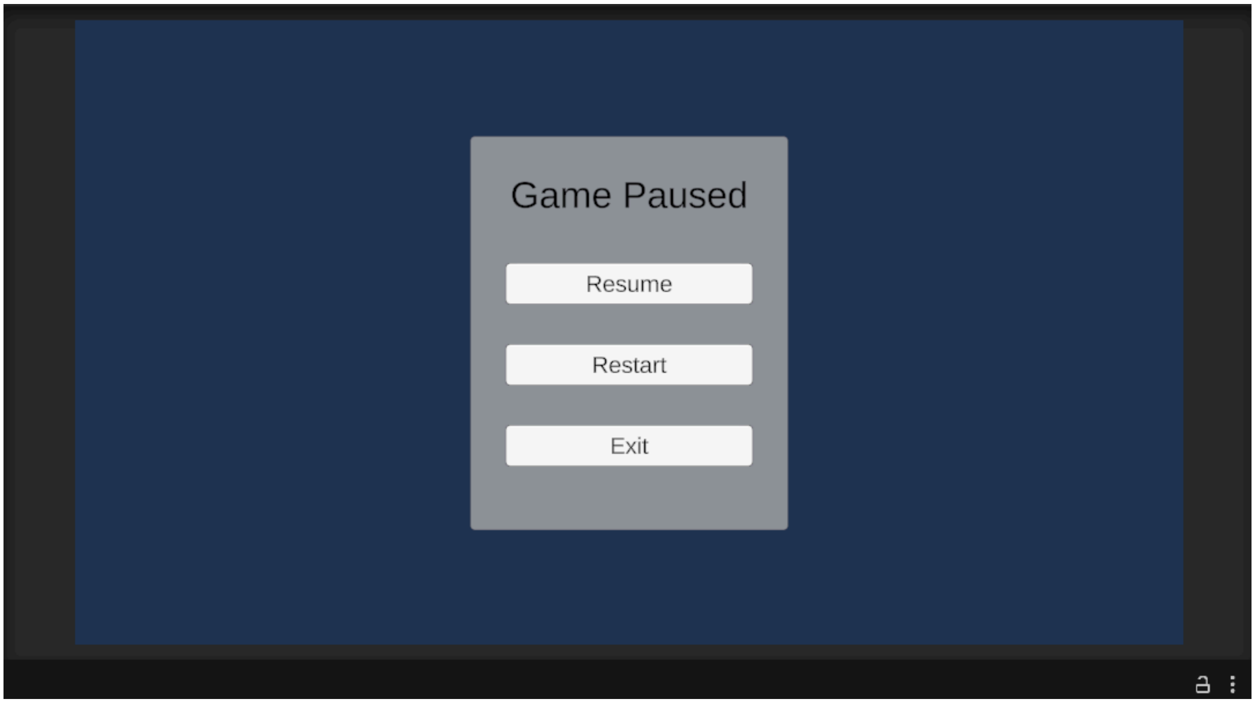
Main menu-> level 1-> level 2-> game over/ end screen

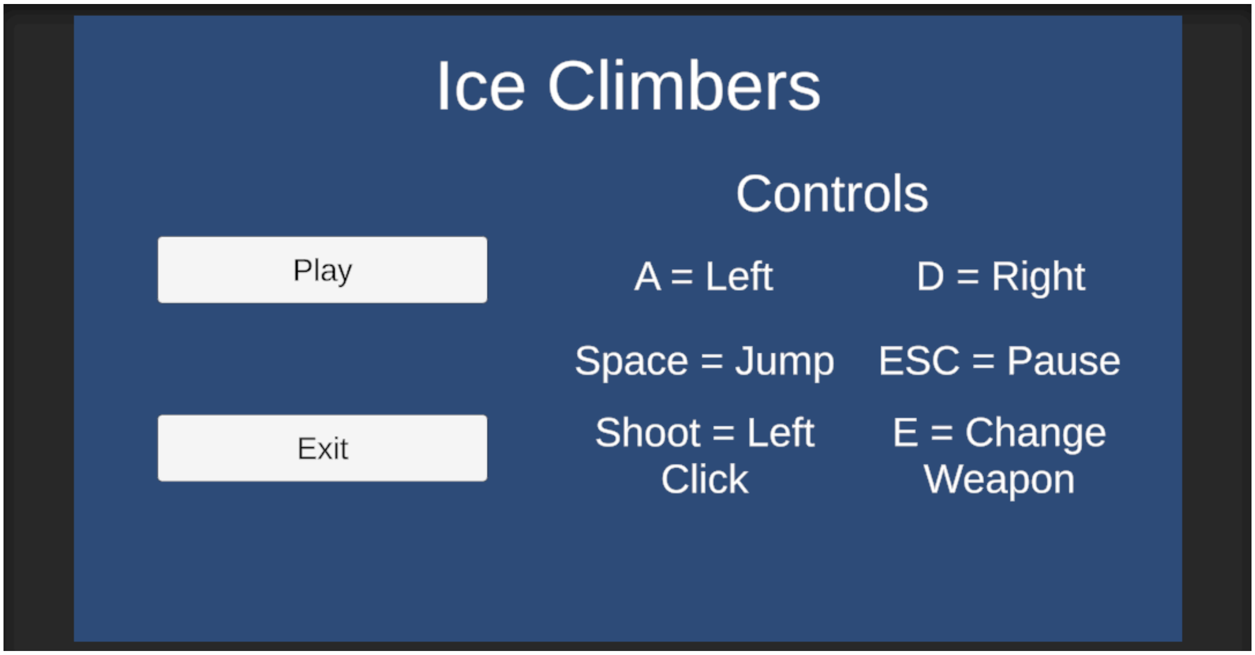
There is also a pause menu that can be accessed at any time.

What Everyone Contributed

Carson Branches Worked on: Level2Merge, BlockBreakingFix, IceWeapon, MovementAndParticle, MenuUI, Movement

- **Implemented the UI** → Main menu screen that shows at the start of the game or exit the whole game entirely. Main menu also shows the controls for the game. Implemented Pause Menu for both levels that is opened on ‘ESC’.





Controls

A = Left
D = Right
Space = Jump
ESC = Pause
Shoot = Left Click
E = Change Weapon

- **Player movement** -> ‘A’ to move left, ‘D’ to move right, ‘SPACE’ to jump. Jumps can only be triggered if the player is standing on the ground. Added particle effects to the movement.
https://drive.google.com/file/d/1K-YVV9bYHAKy26-WoAr-WXiDP7btkz6C/view?usp=drive_link
- **Rocket launcher** → attached the launcher sprite and added animation and knockback to it.
https://drive.google.com/file/d/15Yc1HyHTZBBHiu-4O-UL1qSFHEctDbsb/view?usp=drive_link
- **Block Break** → helped with the block break with a hammer when combining the level design platforms and weapons code.
- **Checkpoint Ghost** → Created both ghost prefabs for level 1 and level 2. Updated checkpoint scripting so the ghosts turn when they reach the end of their designated movement area.

Tawana Ndlovu branches worked on: audioManager, timerManager, checkPointManager, Level1v1Audio, gameOver.

Game Over UI

- Implemented a complete **Game Over screen** that activates when the player:
  - Runs out of time
  - Is touched by an enemy
  - Is hit by a snowball
  - Or touches the ghost checkpoint
- The Game Over screen includes **Play Again** and **Exit Game** buttons.
- UI buttons are functional and reset the game or close it accordingly.

Timer System

- Designed a **75-second timer** (adjustable per level).
- Timer UI is on the main game scene and remains fixed regardless of camera movement.
- Timer:
  - Starts when the level begins
  - Resets on new levels
  - Triggers Game Over if time runs out without destroying the checkpoint enemy

Checkpoint Logic

- Created a **checkpoint enemy (ghost)** that blocks level progression.
- The ghosts speed varies across level 1 & 2:
  - Level 1: 5
  - Level 2: 6

- The ghost must be **hit by projectiles**:
  - Level 1: 3 hits
  - Level 2: 4 hits
- If the player destroys the checkpoint, they're taken to the next level or Game Success scene.
- If they touch the ghost, they are sent to Game Over.

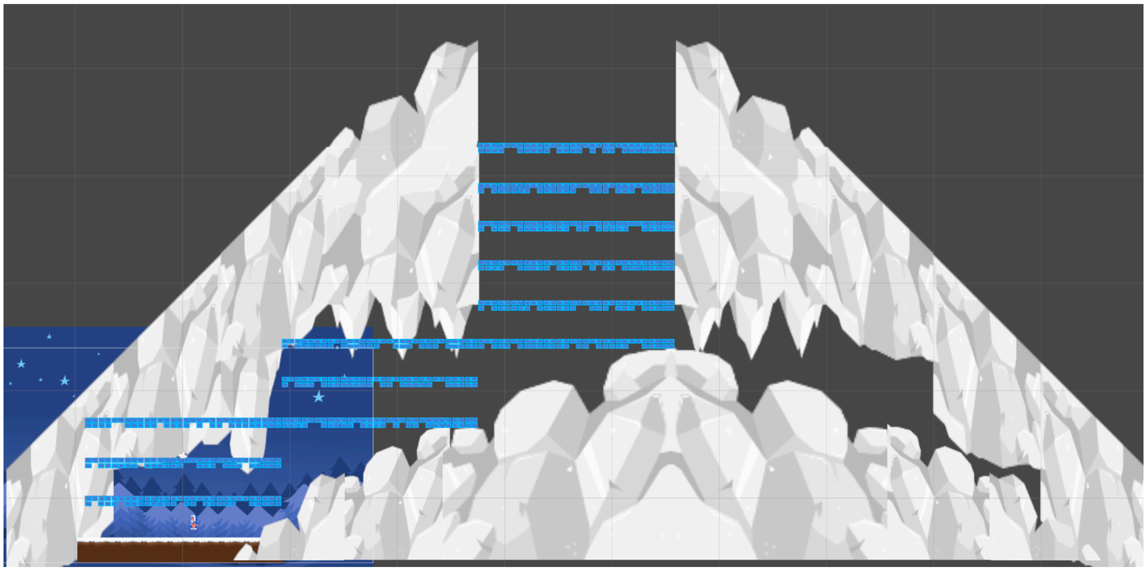
Audio Implementation

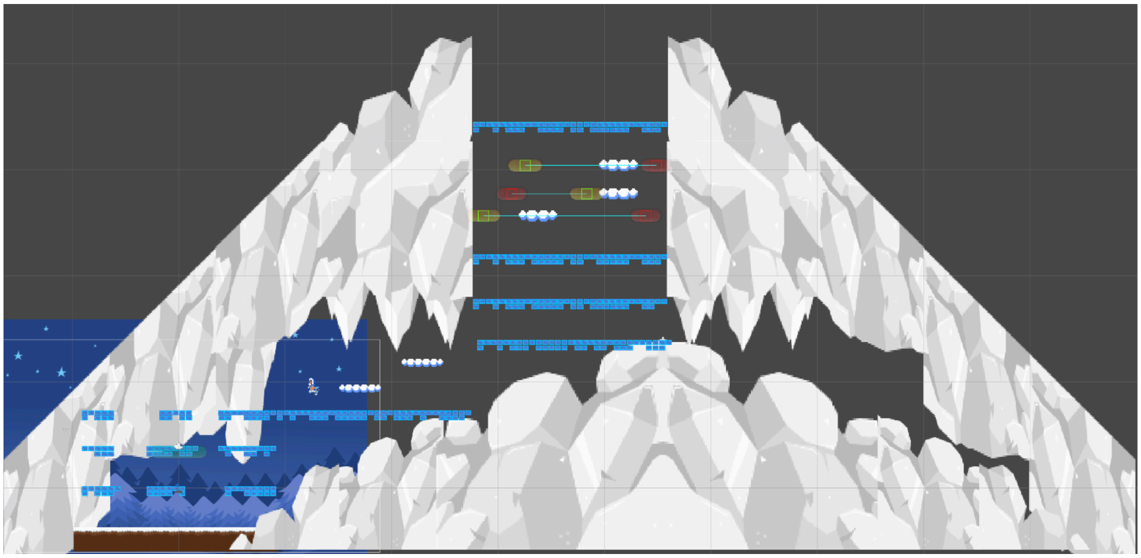
- **Ambient background music** (looping and consistent across scenes)
- **Shoot sound** (when using the rocket launcher)
- **Brick breaking sound** (when hammering blocks)
- **Footstep sounds** (played during player movement)
- **Game Over music** (played when player runs out of time)
- **Game Success music** (played when player completes both levels)
- Used a global AudioManager script with persistent sound effects and one-shots

Jan Branches Worked on: setup-complete, version2-player-animation, leve-two-world-setup

- **Level design 1** → Created the base level, including a background that follows the camera, a mountain formation sprite to serve as the main structure, and the initial block layout. This level was designed to be easy to help players get familiar with the game mechanics.

- **Level design 2** → Designed a more challenging level featuring a wind zone, disappearing platforms, and moving platforms to introduce mechanics and increase the difficulty.





https://drive.google.com/file/d/1BUcs2tPWd69EdHsfq1bCsCG2eAlSiHy7/view?usp=drive_link

https://drive.google.com/file/d/1Q6LhL11H2nbm14Vfe5aUfF6qsDMMscJm/view?usp=drive_link

https://drive.google.com/file/d/1jIhoGMt4Aj53qQ5QR1s14Nr6LPuDtD6d/view?usp=drive_link

- **Player animation** → Created the player idle, run and jump animation of the game.

https://drive.google.com/file/d/1PiC_b2rn5WMATfepMUbFA5Cr_ZOd7r3J/view?usp=drive_link

Aakash Branches Worked on: weapons, enemy-setup, enemyInLvl2, level1Enemy

- **Weapon creation 1 & 2** → implemented the base hammer function to break blocks above using raycast. Helped merge the code between weapons and level design. Created the projectile system for the launcher. Implemented the weapon switching on ‘E’.

https://drive.google.com/file/d/1OnV3AbWCtxxiUAtn43FAUOuqfexB_4gL/view?usp=drive_link

- **Enemy types 1 & 2** → Implemented the standard enemy that walks across platforms. Implemented the long range enemy that shoots projectiles.

https://drive.google.com/file/d/1RaLe9n7WiQMC8IFkVbbFCfyvu4Rup7qR/view?usp=drive_link

Conclusion of game jam

Overall as a group we are happy with our final game. We felt the work was split evenly and given 2 weeks we had learnt and accomplished a lot. Majority of our initial goals that we set when we thought we had 4 weeks were complete. The only feature that we had suggested as a bonus but we could not touch was the water level rising to show urgency of time) which was more of a visual feature than a functional feature. If we were to improve this game in the future we would add animations for the enemies, implement the water level rising, add more levels and enemy types and better balance the game as we felt in certain areas there was improvement that could be made.

Assets

Models & Art:

Provide links or sources for sprites, models, and animations.

NES Ice Climber Sprites: (https://www.spriters-resource.com/nes/iceclimbers/)

Sprite Database – Ice Climber: (https://spritedatabase.net/game/225)

Sound & Music:

NES – Ice Climber Soundtrack (with download link):
(https://downloads.khinsider.com/game-soundtracks/album/ice-climber)

2nd Website for soundtrack download link:
(https://www.zophar.net/music/nintendo-nes-nsf/ice-climber)

UI:

NES Ice Climber Sprites: (https://www.spriters-resource.com/nes/iceclimbers/)