

HavocTown

HavocTown is a multiplayer game in which players are tasked with knocking the enemies off the map via various power ups and weapons, while ensuring that the same does not happen to them. This project will need to implement proper resource management, the use of various OOP features such as abstract classes (i.e. weapons) and varying field scopes. The program flow must keep scalability in mind to ensure high levels of productivity, and proper coding practices. For example, when switching between game screens, or the ease of creating/loading a new map for the game. This program will take advantage of file serialization to load premade configurations for game maps, such as platform position and texture offsets. It will also implement a search bar to help users refine their search when equipping various cosmetic items.

Learning Objectives:

- Resource management and garbage collection to prevent memory leaks
- Effective use of classes to avoid repetition in code and assure productivity
- Efficient program flow to allow for simplistic scalability in the future (for example adding new weapons)
- Use of object serialization to hold static and dynamic data for the program



General running of program:

The program starts at the MainMenu screen. Once in this screen, the user has a choice between two options. They can either choose to see the tutorial, or start a game. If they choose to start a game, they will be presented with a skin select screen. This screen allows each of the two players to select their respective cosmetics. This cosmetic selection features a search bar to help filter their result. Each cosmetic has predefined keywords used for this search. For example, searching “sombrero”, or “hat” will both yield the same cosmetic. If no text is there, it will sort the items alphabetically.

Once cosmetic selection is done. The user can choose one of the three maps on the right. The selected map will be loaded, and both players will be loaded onto this map. Each player starts with a basic pistol and continues to reoccur once ammo is depleted. Supply drops spawn randomly on each map. If a user collides with a supply drop, they will be randomly given another more powerful weapon. The objective of the game is to deplete your enemies lives by knocking them off the map.

Restrictions of the program:

Currently, the program is only limited by its content. Due to its scalable design, it is very easy to implement new weapons or maps for the players. An option to play against the computer would allow users to play without the need for someone else.

Flaws:

The program does not have any known flaws or bugs. Memory management was a great challenge as the graphics library allocates RAM on its own, bypassing the garbage collector that we usually rely on. If assets are not deallocated properly, the program has a “memory leak” and continues to hog memory until it crashes. This issue was resolved.

Things to add/change:

Something I would have liked to add was more weapons and maps. This way the content is more diverse and the user would generally have a more enjoyable experience. In addition, I would have liked to add a campaign gamemode where users can work on levels towards an and goal, giving the game a sense of direction and completion.

PROJECT TITLE		COMPANY NAME	Popal Inc.
PROJECT MANAGER		DATE	2022-12-20

TASK TITLE	PCT OF TASK COMPLETE	WEEK 1					WEEK 2					WEEK 3					WEEK 4					WEEK 5				
		M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F
Project Conception and Initiation																										
Create MainMenu Screen	100%																									
Create Player Sprite	100%																									
Implement World Physics	100%																									
User Accessibility																										
Skin Select	100%																									
Join games	100%																									
User lobby	100%																									
Details																										
Player animations	100%																									
Weapons	100%																									
Weapon animations	100%																									
Supply drops for game	100%																									



