



InCade User Guide

Product Name: InCade

Team Name: Banana Pancakes

Team Members: Rishi Basdeo, Milan Copic, Karanveer Singh, Shri Kiran Srinivasan, Aditya Ramakishran

3D Environment:

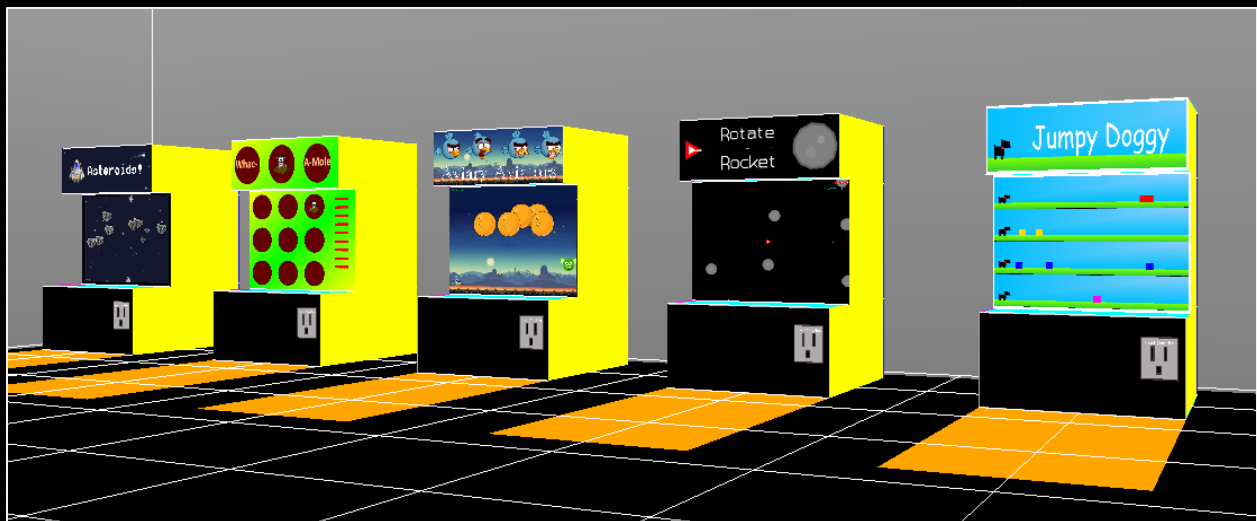


Figure 1: 3D Environment

Controls: W,A,S,D for planar motion, arrow keys for

The 3D environment (where the user is placed at the start of the program) is a 3D GUI used to select the game you would like to play. This was built to be scalable, making it easy to add new games! The user can move in plane with W,A,S,D (as in a traditional game) and rotate the camera with the arrow keys. Once the character "steps" onto an orange activation region in front of an arcade machine, the space bar can be pressed to enter that game. Upon the end of the game, the system will return to rendering the 3D environment. The escape key can be pressed at any time to quit the overall program. While inside any of the games, the 'Q' button can be used to quit and return to the 3D environment.



Rotate Rocket:

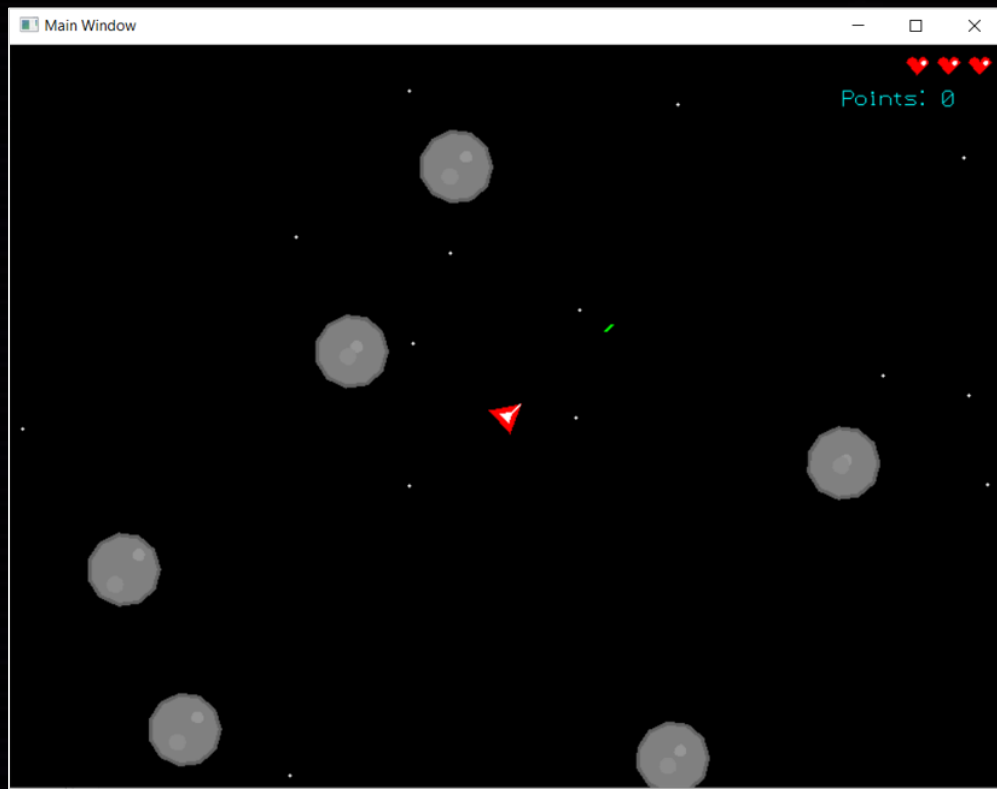


Figure 2: Rotate Rocket Gameplay
Controls: Arrow Keys for planar motion,

- Rotate Rocket is a 2-dimensional multi-directional shooting game. The game starts with the player's space-ship fixed at the center of the screen, along with some space-debris in random directions. The player rotates the ship, firing lasers at the debris.
- In the game, the player starts off with 3 lives. The space-debris start off moving at a random speed in a random direction. When a piece of debris is hit, it disappears from the screen.
- The player controls the space-ship at different speeds using the "Up + Down + Left + Right" arrow keys to push it in a desired direction. When the player fires, the lasers come out of the front of the space-ship and travel in the direction the ship is pointing in.
- The space-debris continuously bounces around the screen. If a piece of debris hits the player's ship, the player loses a life. If all 3 lives are lost, the player loses the game. If the player manages to shoot all the debris on the screen, the game resets.
- The game ends when the player runs out of lives. You can press 'Q' at any time to quit the game.



Asteroids:



Figure 3: Asteroids Gameplay

Controls: Left + Right arrow keys for planar motion, Spacebar

Asteroids is a game comprised of 4 levels. To beat each level, the player must shoot and destroy each enemy ship, located at the opposite side of the screen, before they run out of ammo. Asteroids interfere with this process, and can be destroyed at the cost of wasting ammo.

When a missile makes contact with an asteroid, or a ship, it explodes along with whatever target it meets. These targets are randomly placed and are assigned random velocities at the start of the game, and continually bounce around the screen as the user plays. When a missile is launched, the total ammo inventory will decrease.

Whac-A-Mole:

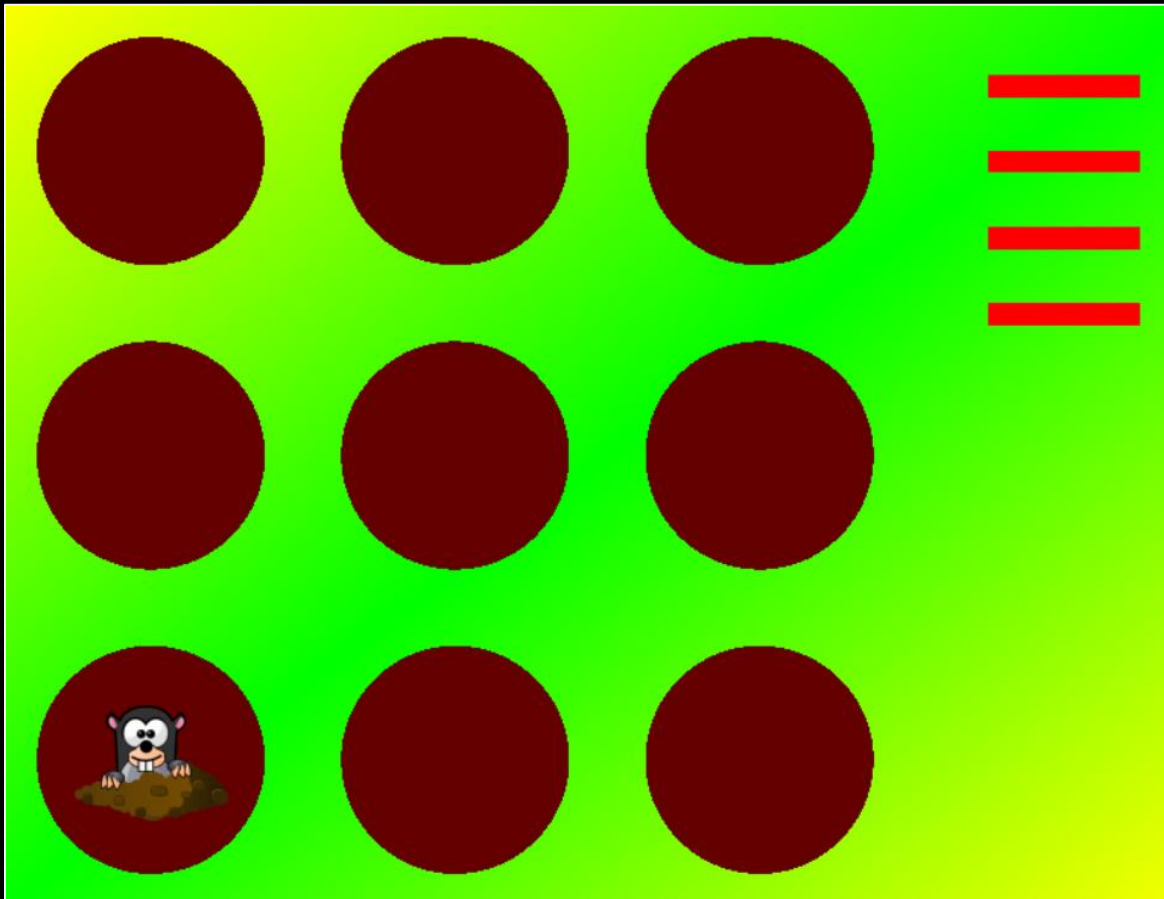


Figure 4: Whac-A-Mole Gameplay
Controls: Mouse Input

Whac-A-Mole is a fun game inspired from the classic Whac-A-Mole arcade game. The game board consists of 9 holes. A mole pops out randomly from one of the holes. The goal is to hit the mole before it disappears. If it disappears without you whacking it, you lose a life. You have 10 lives.

The controls for this game are very simple. You point the cursor over the mole and click on it to whack it. Confirmation of a whack is represented by a sound and a small explosion. Once the game is over, you can either play the game again by pressing "Y" or quit the game by pressing "N".

Aviary Aviators:



Figure 5: Aviary Aviators Gameplay

The player aims for the Boss Pig and tries to hit the Boss Pig, but there are enemies in the way, you can either try to avoid the enemies and directly hit the boss, or you can hit the enemies but the aviator dies with the enemy, and you have to shoot another aviator. You repeat the process until the player hits the boss, upon which you move on to the next level.

Controls:

1. The Aviator can be launched using the SpaceBar.
2. The direction of the launch can be controlled by changing the location of the mouse pointer, the aviator is launched towards the mouse pointer.
3. The velocity at launch can be controlled using the Up and Down arrow keys.
4. The user can press the left mouse button to do an air jump.
5. The user can press the right mouse button to do a turbo boost.



Jumpy Doggy:

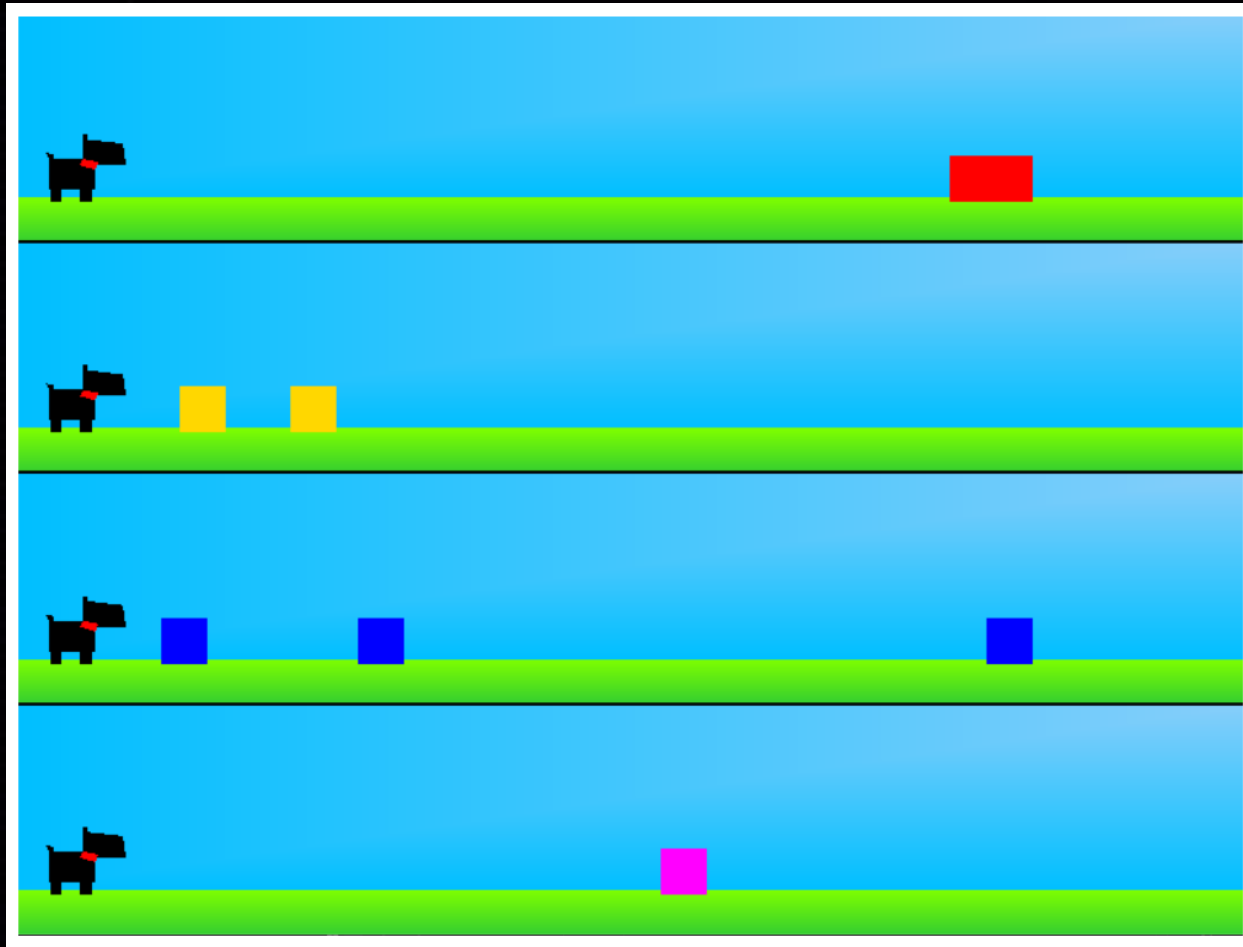
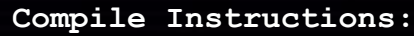


Figure 6: Screenshot of Jumpy Doggy
Controls: Space bar to make all characters jump.

Jumpy Doggy is an endless runner where you play as four characters simultaneously. You must time your jumps in order for all characters to jump over the oncoming obstacle. If a dog makes contact with an obstacle, that lane is eliminated from the game. The game ends once all levels have been eliminated. You can press Q at any time to quit the game.





3. Lastly, navigate to the "Release Folder" under the parent directory. The executable file will be named "FinalAssembly"

« FinalAssembly > FinalAssembly > Release					Search Release
^	Name	Date	Type ^	Size	Tags
	FinalAssembly	11/30/2021 1:38 PM	Application	275 KB	
	asteroidLarge	11/13/2021 5:28 PM	PNG File	8 KB	
	asteroidMedium	11/13/2021 5:26 PM	PNG File	4 KB	
	AsteroidsBanner	11/29/2021 6:32 PM	PNG File	10 KB	
	asteroidSmall	11/13/2021 5:24 PM	PNG File	2 KB	
	AsteroidsScreenShot	11/29/2021 2:09 PM	PNG File	130 KB	
	AsteroidsTitle	11/29/2021 2:20 PM	PNG File	43 KB	
	Aviary Aviators Ban...	11/29/2021 6:33 PM	PNG File	898 KB	
	Aviator 1	11/17/2021 12:56 PM	PNG File	8 KB	

- 4.

« FinalAssembly > FinalAssembly > FinalAssembly >					Search FinalAssem
^	Name	Date	Type ^	Size	Tags
	Debug	11/28/2021 8:26 PM	File folder		
	Release	11/29/2021 4:06 PM	File folder		
	ysglfontdata.c	10/31/2021 10:21 PM	C Source	2,116 KB	
	AsteroidsForSharing...	11/28/2021 12:44 AM	C/C++ Header	3 KB	
	AviatorsForSharing.h	11/30/2021 11:16 AM	C/C++ Header	3 KB	
	fssimplewindow.h	9/19/2021 10:05 PM	C/C++ Header	14 KB	
	JumpyDoggy_for_S...	11/29/2021 6:04 PM	C/C++ Header	3 KB	
	rotaterocket.h	11/29/2021 11:31 AM	C/C++ Header	21 KB	
	whacamole.h	11/30/2021 1:36 PM	C/C++ Header	10 KB	
	ysglfontdata.h	10/31/2021 10:20 PM	C/C++ Header	6 KB	
	yspng.h	11/1/2021 10:09 AM	C/C++ Header	8 KB	
	yspngenc.h	11/1/2021 10:09 AM	C/C++ Header	7 KB	
	yssimplesound.h	10/30/2021 12:26 PM	C/C++ Header	13 KB	
	AsteroidsForSharing...	11/28/2021 12:44 AM	C++ Source	30 KB	
	AviatorsForSharing....	11/30/2021 11:17 AM	C++ Source	11 KB	



Required Standard Libraries

- `stdio`
- `math`
- `string`
- `stdlib`
- `time`
- `chrono`
- `thread`
- `iostream`
- `stddef`
- `vector`

Required YS Libraries:

- `fssimplewindow`
- `Ysglfontdata`
- `yssimplesound`
- `Yspng`

Required Custom Libraries:

- `rotaterocket`
- `AviatorsForSharing`
- `JumpyDoggy_for_Sharing`
- `AsteroidsForSharing`
- `Whacamole`

