Instructions Document

Installation Instructions Summary:

NOTE: This summary is also included in the file README.md, which can be accessed and read from the command-line.

1) Extract the submitted zip file to the OSU flip server.
2) To compile, enter the following command from the terminal line: `make all`
3) OSU flip server connection requirements:
    a) Single player mode requires one terminal window (of minimum size 135x32) connected to the OSU flip engineering servers (flip1, flip2, or flip3). Enter the following command from the terminal line: `cubeRunner`
    b) Two player mode requires three terminal windows, connected to the OSU flip engineering servers (flip1, flip2, or flip3). In one console, run command: `cubeRunnerServer AAAAA BBBBB` (where AAAAA and BBBBB are valid port numbers) and in the other two consoles, run command: `cubeRunner`. Besides this startup, no further actions will be taken in the console running cubeRunnerServer. The server requires a SIGINT signal (ctrl-c) to exit and will otherwise run continuously, restarting itself after each 2 player game.
    NOTE: The multiplayer version of the game renders best when both player terminal windows are the same size.
4) To navigate the game menus, use the up/down arrow keys to change the selected option and press Enter to confirm selection. Follow instructions on the displayed menus for further menu navigation. For menus requesting text information, press enter to submit/continue or move to the next line.
    a) For two player mode, you will be asked to enter the hostname/IP address of the server (flip1, flip2, or flip3) and data port number (AAAAA for both clients). ***NOTE: Both players will enter this information to connect to the game server, see Detailed Setup and Usage section for details on how this is handled.***
    b) (recommended) Visit the Instructions page from the Main Menu for game objectives, game controls, and scoring.
5) Game play will begin, command basic player token movement using the arrow keys or wasd keys. ***There are many more game controls, so it is recommend to review the Detailed Setup and Usage > Game Controls sub-section for all game controls. Or visit the Instructions page from the Main Game Menu.***
    a) For two player mode, by default, Player 1 controls only vertical movement and Player 2 controls only horizontal movement. All other game controls are available to both players. (To enable dual axis control for both players in multiplayer mode, set the constant MULTIPLAYER_DUAL_AXIS_CONTROL in constants.hpp to a non-zero value and recompile [see step 2 above].)
    b) Follow instructions on any transitional menus that appear or at the very bottom of the console during game play, such as “Press enter to continue.”

   c) Once the game has started, press any of the ESC or END or q or Q keys to end the game and return to the main game menu.
6) When game play is over, the main game menu will be presented.
7) (optional) To delete directories, executables, and textfiles created during compilation of the program run the following command from the command-line: `make clean`

***** See the next section, Detailed Setup and Usage, for in depth instructions and screenshots*****

Detailed Setup and Usage

Compilation Instructions

The game is designed to run on the OSU 'flip' engineering servers. It requires two executable files: one for the main game software and one for the server software (to enable multiplayer play). Extract the contents from the zip file on the engineering servers. Type 'make all' in the command line terminal to compile both executables with the supplied makefile (included in the zip). When you are finished with the game, you can enter the command 'make clean' in the terminal to delete the OBJECT folder, both executable files, and the gameHighScores.txt files the game creates when a user achieves a high score.

Running the Game

Single Player

For single player mode, only the executable cubeRunner is required. You can run this executable by entering 'cubeRunner' at the command line after the program is compiled. The minimum terminal window size for the game is 135x32 (widthxheight). An error message will display to the user if the terminal dimensions do not meet the required minimums, and the program will close following the next character entered by the user.

Multiplayer

For multiplayer mode, two executables, cubeRunner and cubeRunnerServer, are required. For this mode, open three terminal windows, all connected to the OSU 'flip' engineering servers and open to the directory where the supplied game files are located (extracted from the zip earlier). In one of the terminal windows, type 'cubeRunnerServer' followed by two different valid port numbers (numbers between 1-65535), as seen in the example screenshot below in image 1. The server will then listen for and accept new

connections from different players. (When finished with the server, use Ctrl-C to quit.) The first entered port will be used for the initial client connections to the server and subsequent game data transmissions, while the second entered port will be used by the server to establish additional connections to the clients strictly for receiving user input. The first command-line supplied port number is the port number that clients must connect to when starting the multiplayer game. The host name is the 'flip' engineering server that the cubeRunnerServer program is running on (either flip1, flip2, or flip3).

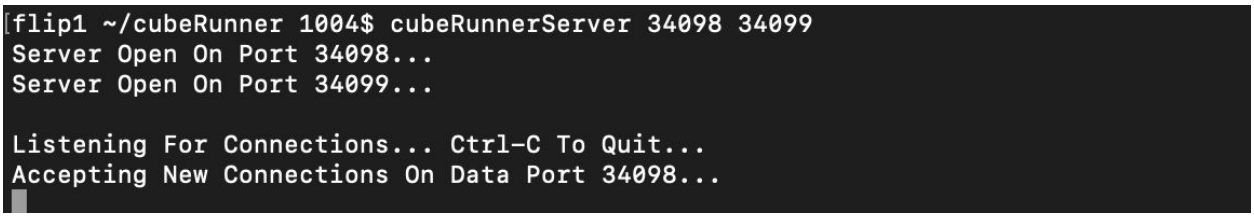

Image 1. Running the cubeRunnerServer program

In the two additional terminal windows, that are at least size 135x32, type 'cubeRunner' in one window at the command line to start the game for both clients. ***Multiplayer mode renders best when both client windows/terminals are the same size***, but the smaller width and height of the two terminal window sizes will be used by default if players have different terminal window sizes.

Game Controls

NOTE: THE GAME IS NOT COMPATIBLE WITH NUMPAD KEYS, AND THEY MAY CAUSE UNDEFINED BEHAVIOR

- w or up arrow key - move up
- s or down arrow key- move down
- a or left arrow key- move left
- d or right arrow key- move right
- e - left up diagonal (single player only)
- r - right up diagonal (single player only)
- c - left down diagonal (single player only)
- v - right down diagonal (single player only)
- spacebar - fire shot (see subsection on Shooting for more information)
- l - lock orientation (Space world only)
- u - unlock orientation (Space world only)
- p - pause/unpause game
- Q or q or END or ESC key- quit the game and return to main menu

- t - transition to next world (testing/grading purposes only - disable by setting TRANSITION_WITH_INPUT to 0 in constants.hpp and recompiling [see step 2 above])

Additional Controls for Testing

For convenience, we have included toggles in constants.hpp to make the game easier to test. The MULTIPLAYER_DUAL_AXIS_CONTROL variable controls whether or not both players (in multiplayer mode) can move in all directions. By default, this is set to 0 and player 1 controls vertical movement while player 2 controls horizontal movement. If this value is set to 1, both players can control movement in any direction. The TRANSITION_WITH_INPUT variable controls whether or not players can trigger a world transition using the key ‘t’. By default, this is set to 1 for grader convenience so that the player(s) can trigger a world transition by pressing ‘t’.

Navigating the Game

Game Menu

The game opens up with a starting animation where you can press any key in order to continue at anytime during its playback. The main game menu (seen in image 2 below) is then loaded after a transitional animation. (Players can also skip the remainder of these animations by pressing any key at any point while they are playing.) The player can then choose if they want to play a single player or multiplayer game, view high scores, view game instructions, or exit.

If the client chooses 1 player or 2 player, they will be brought to a second menu (seen in image 3 below) where they can choose their initial game difficulty mode: easy, normal, or hard. For player 1, once they have chosen their difficulty, they are prompted for a username. After entering their username (19 characters max) the player can begin the game.

Image 2. Main game menu

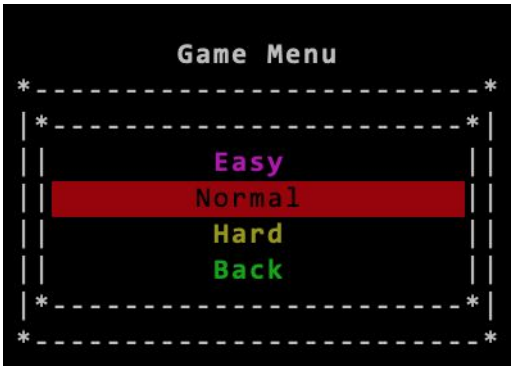

Image 3. Choosing game difficulty

After choosing their difficulty level, players that have chosen the 2 player game mode are then prompted for a hostname and port number to connect to the game server as seen in image 4, where the hostname is the flip server that the cubeRunnerServer program is running on (e.g. ‘flip1’ or ‘flip1.engr.oregonstate.edu’). Both players are then prompted for their usernames (19 characters max). After a player enters their username, the game client attempts to connect to the game server for each player. If the player has entered incorrect server info that does not match a currently running game server, they will see an error message that there is no available server at the entered hostname/port and the game will return back to the main game menu.

The first player to connect to the game server is player 1, and the second player to connect to the server is player 2. As explained in the additional controls section, the default controls for multiplayer allow player 1 to control the vertical axis and player 2 to control the horizontal axis. Both players can use the other available controls. While player 1 is connected and waiting for another player, they will see a waiting screen that tells them to “please wait for another player to connect”. It is not intended for the player to exit during this waiting screen and as such, if the player wants to quit at this point they must enter CTRL-C from the keyboard (the server at that point must also be restarted).

Once both players are connected, the server will check the game difficulty level chosen by both players. If these game difficulties match, the game will start at that difficulty level after a 5s countdown that both players can see. If the game difficulties do not match, the server will start the game at the easier of the two chosen difficulty levels after informing both players that the game will start at the easier difficulty level. In this case, there is a 10s countdown before the game starts to allow both players to read the error message.

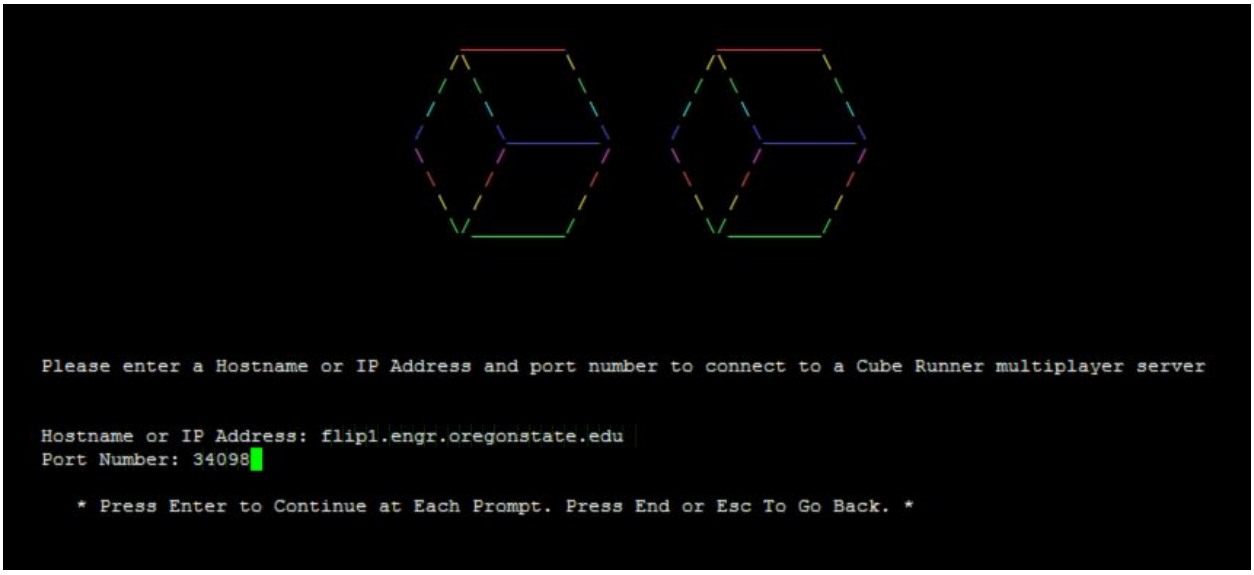

Image 4. Entering server hostname and port number before attempting to connect for multiplayer game. (Both clients enter the same port number at this step, which is the first of the two port numbers specified on the command prompt when launching cubeRunnerServer, the connection/data port. See image 1.)

If the player(s) chooses the high scores option from the main game menu, they will see the top 10 highest scores for all game modes along with the time, difficulty, and username(s) for that play. If there are no high scores, then a message displays informing the user that there are no high scores on file yet and to play the game to generate some.

Choosing the instructions option from the main game menu allows the player to see game information including user controls, game objectives, and other game play information. Players can exit the game by choosing the exit option from the first main menu.

Quitting the Game

When a player presses the q (or Q or ESC or END) key after the game has started, they will quit the game and return to the main game menu. In multiplayer mode, the player that did not quit the game will see a message that the other player has quit the Once the player confirms the message, they will also return to the main game menu. Players can also quit during death animations and transition screens to the next world (other than the first world transition screen which allows users to press any key to continue). In both multiplayer and single player game modes, the score for the game will not be saved as a high score when the game is terminated early (compared to the game ending when the player(s) loses all of their lives).

Pausing the Game

When a player presses the p key after the game has started, the game will pause. In multiplayer mode, there will be a message at the bottom of the screen saying that the other player has paused the game. The player that paused can then press p again to resume the game. (Due to synchronization issues, some Obstacle movement will proceed following a user's command to pause in multiplayer mode. Therefore, the cube will be reset in the starting 'safety zone' when gameplay is resumed after it is paused [multiplayer mode only].)

Game Play

Initial State

The player(s) starts with either 3, 4, or 5 lives (depending on the difficulty level chosen at the start of the game, with easy difficulty having the highest number of lives). The game's scroll rate, the rate at which dynamic obstacles move, and the quantity of obstacles are highest for hard mode and lowest for easy mode (with obstacle counts and rates in between for normal mode).

Minicubes

Minicubes, represented in game as a single unicode square, randomly appear in the game environment throughout each level. When the player(s) collects these minicubes, their score increases by 10 points.

Player Cube

Image 5 shows the player token that the player(s) can move around the game screen to progress through the environments. One edge of the cube changes to a magenta color, depending on the last player token direction commanded by the player(s).

Image 5. Player cube which has most recently moved to the right

Shooting

The player(s) can destroy obstacles in their path using the spacebar. The number of hits required to do so varies by Obstacle type and game difficulty (explained further in the

Water Level, Land Level, and Space Level sections). Points obtained by shooting obstacles equals the number of hits required x 10. When an Obstacle has been hit and is ‘damaged’, it will turn red and an indicator at its top-left corner will display the number of hits remaining before it is destroyed. If the player(s) does not keep shooting the Obstacle in quick enough succession, the Obstacle will begin to regenerate. Only one shot can be ‘in the air’ at a time.

Water Level

With no user input, the player cube will remain in the same location relative to the console boundaries. The water level has four different obstacles: seaweed, coral, sharks, and octopi. Seaweed and coral are stationary obstacles, while shark and octopus are mobile obstacles. Depending on game difficulty, a shark will be destroyed with 3, 4, or 5 player shots (from easy to hard mode, respectively). An octopus will be destroyed with 2, 3, or 4 player shots. Image 6 is a screenshot of the water level.

Image 6. Screenshot of the water game level

Land Level

With no user input or only horizontal input, the player cube will “fall” to the bottom of the console. The land level has four different obstacles: rocks, trees, birds, and bats. Rocks and trees are stationary obstacles, while birds and bats are mobile obstacles. Depending on game difficulty, a bird and a bat will be destroyed with 1 or 2 player shots (1 shot for easy and 2 shots for normal and hard modes). Below is a screenshot of the land level.

Image 7. Screenshot of the land game level

Space Level

The background will scroll in the direction of the last user commanded direction, unless locked by the user(s) by pressing the l key. User(s) can return to the default scroll method by pressing the u key to unlock (see the Game Controls section). There are four obstacles in the space level: asteroids, planets, comets, and spaceships. Asteroids and planets are stationary obstacles, while comets and spaceships are mobile. Depending on game difficulty, a comet will be destroyed with 1, 2, or 3 player shots (from easy to hard mode, respectively). A spaceship will be destroyed with 1 or 2 player shots (1 for easy and 2 for normal and hard modes). Below is a screenshot of the space level.

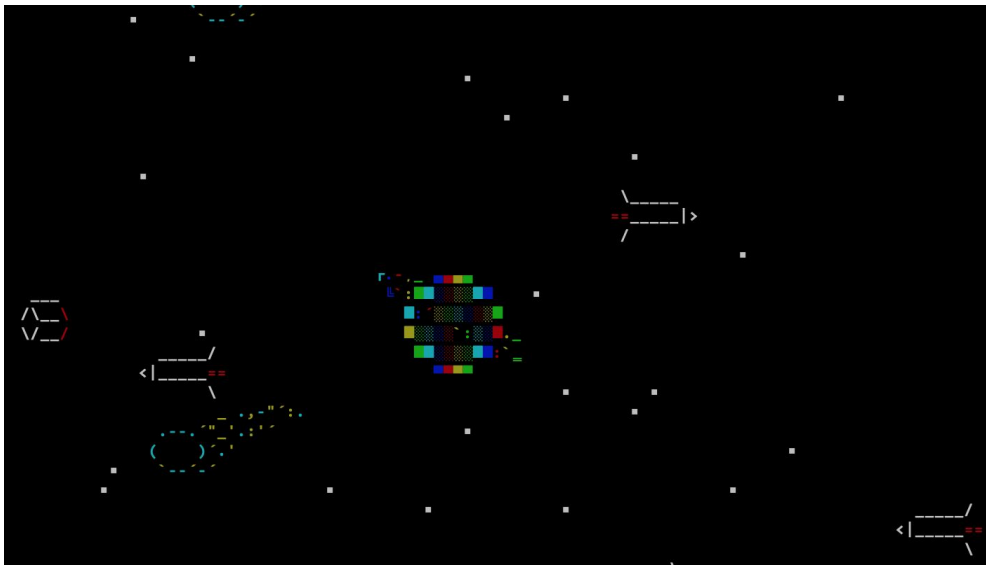

Image 8. Screenshot of the space game level

Transitions Between Worlds

Players start in the water level. When a player reaches 500 points (or when the t key is pressed and TRANSITION_WITH_INPUT constant is equal to 1, see the section on Additional Controls for Testing), they will transition to the next level until game over. In between each world, there is a transitional screen (such as the one seen in the screenshot in image 9). Players transition from water to land, and then from land to space, and from space back to water again. This cycle repeats until game over. Each world transition increases the game difficulty by introducing a 10% increase in scroll and obstacle movement rates, up to the sixth world transition (at which point rates are capped). In multiplayer mode, both players must confirm the world transition before the transition will actually occur. (The initial transition animation can be confirmed or bypassed with any key, but all subsequent transitions must be confirmed or bypassed with the 'Enter' key.)

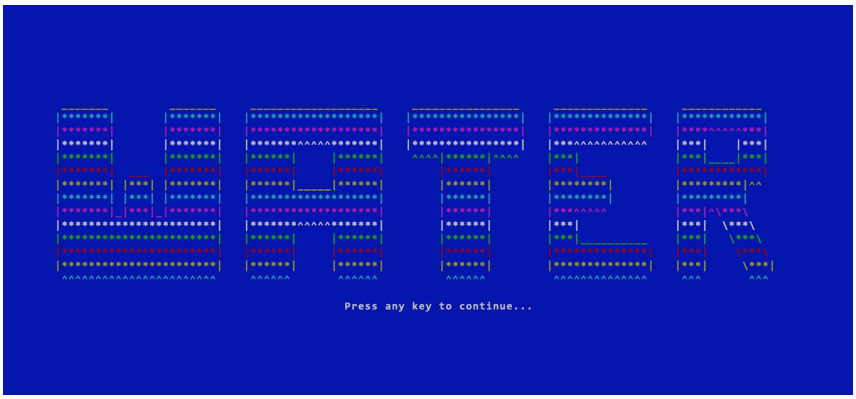

Image 9. Transition screen before water level

Injury and Death

Every time the player(s) hits an obstacle, they lose one life and a death animation appears along with a message at the bottom that tells the player(s) how they died. Each player must press enter to confirm the death (as seen in image 10). When the player(s) dies but the game is not yet over (they still have lives left), the player cube resets to the left of the screen after the player(s) confirms the death in a 'safety zone'[1]. In the case of multiplayer, both players must confirm the death before the game continues.

[1] The width/size of the which is determined by the constant RESET_WIDTH in constants.hpp.

Image 10. Death animation and confirmation

Game Over and High Scores

When the player(s) loses all of their lives, the game will end and they will see a transitional game over screen and can press any key to continue. After the transitional game over animation, they will see a screen showing the stats of their game (player name, other player name, player number, time, final score) and a message that says their score is a high score to be saved to the high scores text file gameHighScores.txt. The high score criteria is that the score is higher than the lowest of 10 scores saved in the file gameHighScores.txt. Only the top 10 scores that are greater than zero are saved as high scores and can be viewed by selecting the main game menu High Scores option. Even if there are currently no high scores when the game finishes, a score of 0 will not be saved as a high score. In the case of multiplayer mode, only the client used by player 1 will save the high score for the game to prevent duplicate high scores from both players. After viewing their game stats, the player(s) can press any key to continue back to the main game menu.