

Final Project Report- Vampire Rush

Jacob Lin, Suvir Mehrotra, Chang Chun Hwang, Pranav Charkupalli

Team Representative - Jacob Lin

Names/Github Handles/ID's:

Suvir Mehrotra

- Email: suvir.mehrotra@gmail.com
- Github Handle: Suvir-Mehrotra
- ID# 304769726

Jacob Lin

- Email: jacobchenlin@gmail.com
- Github Handle: articbear1999mail
- ID# 605172088

Pranav Charkupalli

- Email: pranavcharku07@gmail.com
- Github Handle: PranavARC
- ID# 905143315

Chang Chun Hwang

- Email: c2yellow@gmail.com
- GitHub Handle: c2yellow
- ID#104171224

Vampire Rush Description:

The premise is that you are a vampire walking through some on a lonesome road attempting to reach your cozy home in a cave. You can't stay in the light for too long, as it slowly burns your to a crisp. You can alleviate your pains by hiding in the shade of buildings, but this shade only lasts temporarily as the shadows do move. The light sucks and the shade can't always help you, so as a vampire, rush through the level and attempt to reach the end before the light gets ya! Also, stay safe and dodge any reckless drivers along your path!

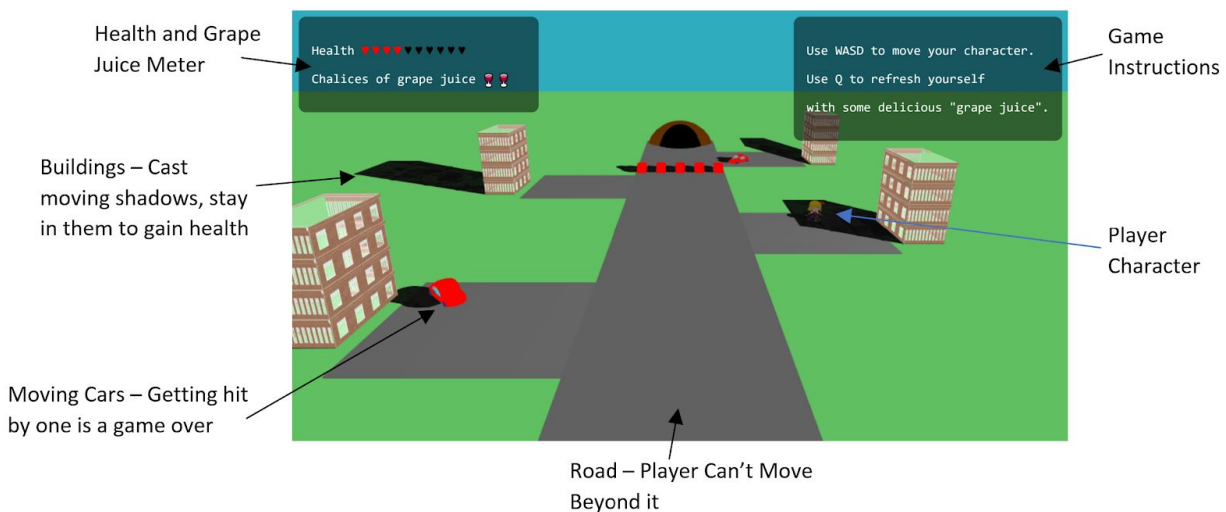


Figure 1: Image of gameplay. The main character will progress through the level to the goal and has to hide in the shadows cast by buildings to avoid the sun. The sunlight reduces the vampire's health, when you lose all your health, you lose the game.

Gameplay:

In our final release, gameplay is focused on a vampire and the player is given a 3D view of the whole world. Our three levels have a cave as the goal and several buildings that cast shadows within the map. Additionally, there are also some cars to serve as obstacles along the way that can instantly kill the vampire. The player must navigate from the starting position to the cave while avoiding being in the sun for too long, or die a fiery death. This is triggered when the health bar completely empties, causing the character to explode. However, hiding in shadow will slowly refill the player's health as long as you remain within its bounds. Reaching the cave will allow the player to proceed to the next level. Lastly, there is a skill that allows the player to consume a potion and refill the vampire's health! Use these potions wisely, else you may not

make it to your final cave, as the stock is limited based on difficulty and does not replenish between levels.

Controls are simple, the character is moved using standard WASD keys and the camera is mapped to IJKL. The Q button is used to activate the potion skill. We have left the camera as 3rd person omniscient for ease of visibility throughout the levels.

How it Works

The program starts out at a title screen, created with CSS and HTML, that lets the player choose between three difficulty levels. Upon picking a difficulty, the associated .js file is then loaded and the game begins. At the start of the game and upon reaching every goal, the .js file checks what level is being played at the moment, and subsequently loads the corresponding map data containing the buildings, obstacles, and paths along with their collision data. As the player moves along the path, their health constantly decreases, and upon reaching zero, triggers a death event that creates an explosion at the player's current location. Conversely, while in shadow, the player slowly gains health back, until they reach max health. After the player reaches the cave, the level variable changes, leading to the next level being drawn and the position of the character reset. Finally, reaching the final cave on level 3 displays the final victory screen.

Advanced Requirements Description/Implementation:

Description:

Shadows

Shadows are the premise of the whole game, as the vampire has to hide in the shadows to avoid the sunlight. The shape of shadows can change throughout the game based on the movement of the sun or an object.

Collision Detection:

This is also an integral part of our game, since we can't have the vampire walking through objects. Additionally, collision detection is necessary for deadly obstacles to work. Particular examples of how collision detection functions in our game can be seen with the cars that kill the vampire upon collision and the vampire being stopped upon reaching the edge of a building.

Implementation:

Collision Detection

The general implementation for collision detection was split into two parts, shadow detection and general obstacle detection. For obstacle detection, the process starts by finding the exact x and z axis locations at the corners of a building. Then, if the character was going to move into the location of a building, its position is instead kept static. The same logic is applied to the sides of the roads, so that the character cannot move outside the boundaries. For the cars, these objects

move based on a period which determines their current x and z positions. If the character is in range of a car, health is set to 0. This is calculated by mapping the period itself with how the cars are positioned in the x-z plane and then using the scaling of the car's position matrix to determine what coordinates should be used to determine whether the character is in contact with a car or not. Lastly, collision detection for the cave is implemented with the same logic, by finding the coordinates of the cave. If the character is in that range, the caveIn function is called to display a "you won" message for that level.

Shadow Detection

In order to implement shadow detection, we employed the use of shadow rays. At first, it was annoying to get started because much of the implementation involved using pen and pencil to figure out the math equations. However, once the math equations were finished, things became easy. Essentially, for all figures on each map, give a list of the faces which lie on a plane. Using some multivariable math equations, figure out the intersection of all the planes with the shadow ray. Then, decide if that intersection was within a valid bound such that the face was on that plane. If it is, it is in shadow, else it is not in shadow. All shadow ray code was placed into the simpleRayCast file. If shadow rays count as another advanced feature, then we did extra.

Models

The model code is imported from the week 7 tiny graphics code. At first it took a while to integrate, but we realized that past projects must have used models somehow, so we looked at them and realized they just copied the Shape_From_File code from the obj_to_file demo. We took their lead, and spent a lot of time looking for compatible obj and png files. Then, we simply applied it in the same way the obj_to_file demo did.

Direction

This was simple to implement, every time a control is pressed, it checks if the angle was within a certain range. If it is within a certain range, it will turn, else it will do a hard turn into the original position. The softer turn of 45 degrees is to deal with the case that people press two adjacent directions at once.

Team member contribution:

Suvir Mehrotra:

Contributions: Character health logic/All forms of collision detection/Glitch-debugging/Initial shadow detection

Github commits:

<https://github.com/intro-graphics/team-project-team-pranav/tree/ff9a6af6a9745917a52d41188c9c087574ff9a9c>

<https://github.com/intro-graphics/team-project-team-pranav/tree/92890b394548204ba437c55891fba8cd04bb8b4e>

<https://github.com/intro-graphics/team-project-team-pranav/tree/3180daa6264464df6653a503d64efab98f780ebe>

<https://github.com/intro-graphics/team-project-team-pranav/tree/ec4a2c685a2f69ddd56db3d44aa5008efe7b3025>
<https://github.com/intro-graphics/team-project-team-pranav/tree/50cccd742491de57486294086242963cf55e4d03>
<https://github.com/intro-graphics/team-project-team-pranav/tree/91bffb7057ae25136fa979718604927807597085>
<https://github.com/intro-graphics/team-project-team-pranav/tree/8168a57b4704ede80b03a7857612b64e81e29e65>
<https://github.com/intro-graphics/team-project-team-pranav/tree/49ce0099b23fa032be17662f8e1f3411697b2210>
<https://github.com/intro-graphics/team-project-team-pranav/tree/9755ace0241f1a78d966bca4db27e93cde34b456>
<https://github.com/intro-graphics/team-project-team-pranav/tree/a598600c26a71d5747c32094a7fb8b1c29d440b0>
<https://github.com/intro-graphics/team-project-team-pranav/tree/1c1ad3979cf90e8cd2b982b96ba63e6e1cf505f0>

Chang Chun Hwang

Contributions: UI design and logic/2d graphics/game flow and scene transition logic/player character model/BGM and SFX

Base for character model (retextured for project): <https://bowlroll.net/file/1347>

<https://github.com/intro-graphics/team-project-team-pranav/tree/50dbd94020e4d221e8496ce208304c8162ede876>
<https://github.com/intro-graphics/team-project-team-pranav/tree/60aea202bc4e6ea1f3061f85291309f5a40fe4e1>
<https://github.com/intro-graphics/team-project-team-pranav/tree/d93429e959023e4298223ea9e780be5c8677e0d4>
<https://github.com/intro-graphics/team-project-team-pranav/tree/483b327a59ec0047c69e3869764e52d1af162974>
<https://github.com/intro-graphics/team-project-team-pranav/tree/29423bf6f1d82f2ea03ff87efa7b41fd3d9774cc>
<https://github.com/intro-graphics/team-project-team-pranav/tree/5d94b284b1afb114805c0de671fd05d347660f>
<https://github.com/intro-graphics/team-project-team-pranav/tree/41d7d0fd001f44aa4ad658ffb6adaa424a22587a>
<https://github.com/intro-graphics/team-project-team-pranav/tree/a56de26200916d64e308edd26f9928d9570bf12a>
<https://github.com/intro-graphics/team-project-team-pranav/tree/1b120d2d785cdc5b3356ca2c1c89d2105c9c5f07>
<https://github.com/intro-graphics/team-project-team-pranav/tree/9483abd9d25409cc6ac2c826eac64a07c3f9825e>
<https://github.com/intro-graphics/team-project-team-pranav/tree/711b306dcbdde35126293f7f1aaed5c8aa8e424c>

Jacob Lin -

Responsible for all Shadow work, and most of the importation of models,direction of character Model for building-

<https://free3d.com/3d-model/brick-building-51863.html>

Shape From File code is directly pulled from week 7 discussion -

https://github.com/JonathanCMitchell/CS174A_Discussion_Sections/blob/master/week_7/tiny-graphics-js/examples/obj-file-demo.js

Shadow_Shader.js is pulled directly from Jonathan's past final project code

https://github.com/JonathanCMitchell/CS174A_DrivingSimulator/blob/master/shadow_shader.js

I committed a lot even when deleting comments, so there's a lot of commits, not all of them were hugely important. Including what I think were the bigger changes.

Oldest to newest:

<https://github.com/intro-graphics/team-project-team-pranav/commit/d44ba62ca3121513a771bacc9621c86ed7d0b6f7>
<https://github.com/intro-graphics/team-project-team-pranav/commit/e31c99a7bf7e93968dd16f11f56c0b7986dc3801>
<https://github.com/intro-graphics/team-project-team-pranav/commit/7778ef9dfe1feb1c7c3996e692f1d637fd644441>
<https://github.com/intro-graphics/team-project-team-pranav/commit/3589446c4403e4f4154f3bce7b7c1aa5140a0b56>
<https://github.com/intro-graphics/team-project-team-pranav/commit/0f0b819427dc24c8a4124095df5b3a976d9c60ee>

<https://github.com/intro-graphics/team-project-team-pranav/commit/6ba9619d5756cb0a763386c60df7b7d8b7951432>
<https://github.com/intro-graphics/team-project-team-pranav/commit/43d1d067e41e3dfcf59cf68b591ae2666df2a32c>
<https://github.com/intro-graphics/team-project-team-pranav/commit/1dfa772f364c40d2ea061ef3f34d4922be8409f>

Pranav Charkupalli: Level creation, design, and collisions, movement, health refills, difficulty levels

<https://github.com/intro-graphics/team-project-team-pranav/tree/b7b396d5e2083e6e531fc8b7422b02a836ab1bc6>
<https://github.com/intro-graphics/team-project-team-pranav/tree/78315d24d356ce88add5c169dae7e329577a4b0a>
<https://github.com/intro-graphics/team-project-team-pranav/tree/3b3c9040b25653edd11e6db1151fa5adb4ca48ac>
<https://github.com/intro-graphics/team-project-team-pranav/tree/da72a196ee507f1a4e4bf25a7778d64595cace50>
<https://github.com/intro-graphics/team-project-team-pranav/tree/7f932ec4fc35c3e54140482beeb0c0816d9e1804>
<https://github.com/intro-graphics/team-project-team-pranav/tree/c115f5ff42546725572b628700a08073d84a090c>
<https://github.com/intro-graphics/team-project-team-pranav/tree/f30038cf9f378dd7514fd67c26d60c398769cff3>
<https://github.com/intro-graphics/team-project-team-pranav/tree/dc9f4cd661efe2fb6c3d14fb40485b73ad94fe35>
<https://github.com/intro-graphics/team-project-team-pranav/tree/f313d2cf3c7be8669067122f6478027024e82cd3>
<https://github.com/intro-graphics/team-project-team-pranav/tree/9092b599044c6ae692bc434d3568b14919db43ff>
<https://github.com/intro-graphics/team-project-team-pranav/tree/a358f2b8ddf33832b660b484b363f77f7d4911b5>
<https://github.com/intro-graphics/team-project-team-pranav/tree/1190118727d6a24c582040a632fda051a1db3325>
<https://github.com/intro-graphics/team-project-team-pranav/tree/be2a15eff91bd8a394901bfbfd148a970a756817e>
<https://github.com/intro-graphics/team-project-team-pranav/tree/a011f416eeaf09fd78bcb116c1671294f0ca105>
<https://github.com/intro-graphics/team-project-team-pranav/tree/b77e6281bf0a9aa4fc2053a9d6269459e3307c01>
<https://github.com/intro-graphics/team-project-team-pranav/tree/dccc668840bf3460d355c4deb752c231a38b340>
<https://github.com/intro-graphics/team-project-team-pranav/tree/3f11015cd71e65318bb63a207b31800ac6490a29>
<https://github.com/intro-graphics/team-project-team-pranav/tree/0b41a1fa774c6b94b8d101a2116a1822ebe818f8>