

Super Hotline Miami Maker: Video Games Practical 3

fsm

December 2016

1 Introduction

In this report I would like to present the game I have created, Super Hotline Miami Maker.



Figure 1: Title

SHHM is a top down shooter puzzle game with a level creator included. The title is a mix up of three games which this game is a mash up of: Super Hot, Hotline Miami, and Mario Maker. The Player takes controller of the character while they are under attack but times only goes as fast as the character moves. The Player must use the extra time afforded to them to dodge bullets and kill the agents attempting to kill them. The time manipulation idea comes from Super Hot. It's similar to hotline Miami in the sense that it's a topdown shooter and the level maker is how Mario Maker is included. Originally the game was going to include a server so players could share their levels, like Mario Maker, but this wasn't implement due to time constraints.

2 Running the code

The code should be runnable straight from the processing IDE but you may need to install the library Minim. To do this go to the 'Sketch' tab at the top, go to 'import library' then 'add library...'. From there search for Minim then select install. See figure 2 and 3 for an example.

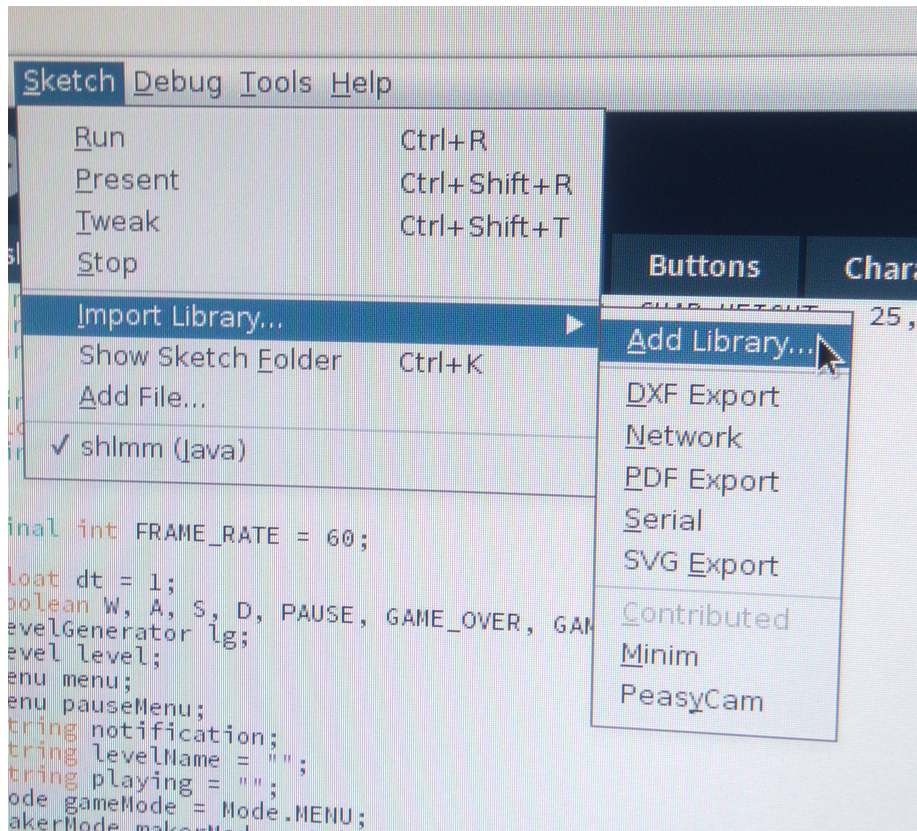


Figure 2: finding libraries

3 Game Goal

The goal is to create a challenging level then complete it. To complete a level you must kill all the enemy character in the level. To achieve the goal you can either shoot the enemies or hit them with close range melee but they will be trying to kill you at the same time.

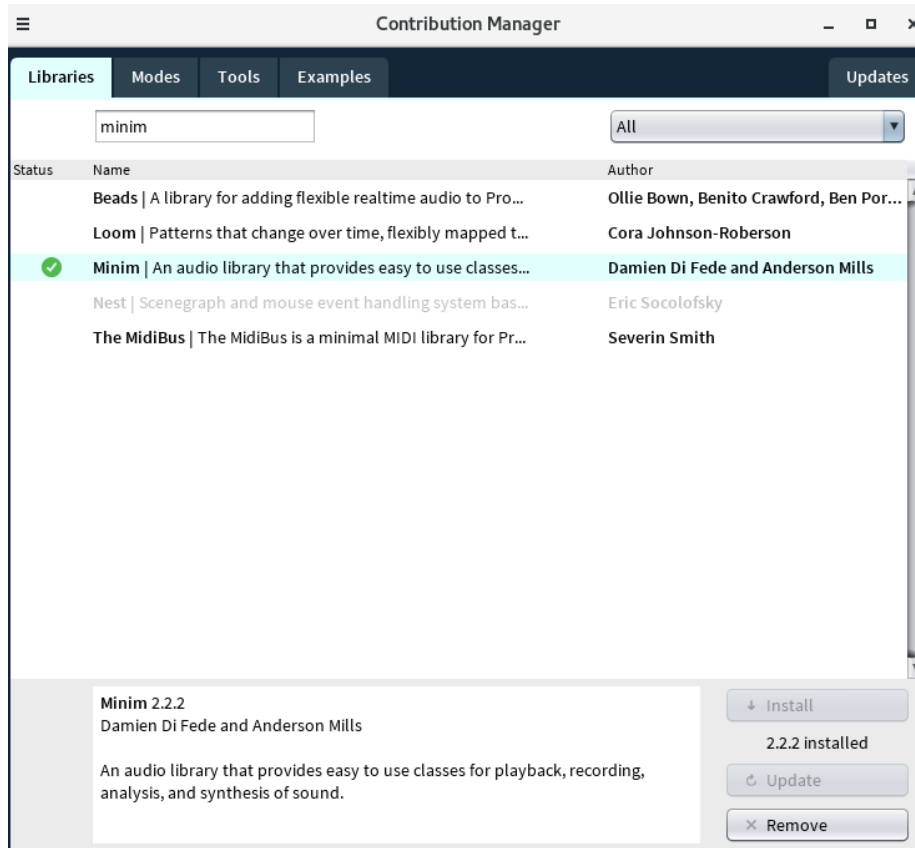


Figure 3: Minim installed

4 Player

The player is represented by a magenta character. The player may move in eight directions, all 45° apart, like a compass. The player will face towards the mouse and that is the way all attacks will go. If the player has a gun the attack will cause a bullet to be fired, if not then a melee attack will be preformed. Melee attacks can also be used to pick up guns. If you kill an agent with a gun while you don't you shall then be given their gun. When a gun runs out of ammo the screen will be given a red outline. At any time the gun can be dropped with the space bar.

5 Enemy

The enemy is represented by a red character. There are two types of enemies: agents and agents with guns. When an agent has a gun they will do their best

to get to a position where they can see you and then take a shot. This shot can sometimes kill their allies and while this was originally unintended I thought it was quite funny so decided to leave it. From what I've seen players are always quite happy when one of the agents shoots another. If an agent has no gun then they will run immediately towards you and use melee to kill you. Both types of agents use the A* algorithm for path finding. The level is created with a graph of multidirectionally connected nodes that can be traversed to find the player. See figure 4 for a visualisation of the graph. To decide if the agent can see if the player a line is created between the agent and the player it is tested if the line intersects any of the walls. To reduce the number of team kills I could also check that no allies are in the way.

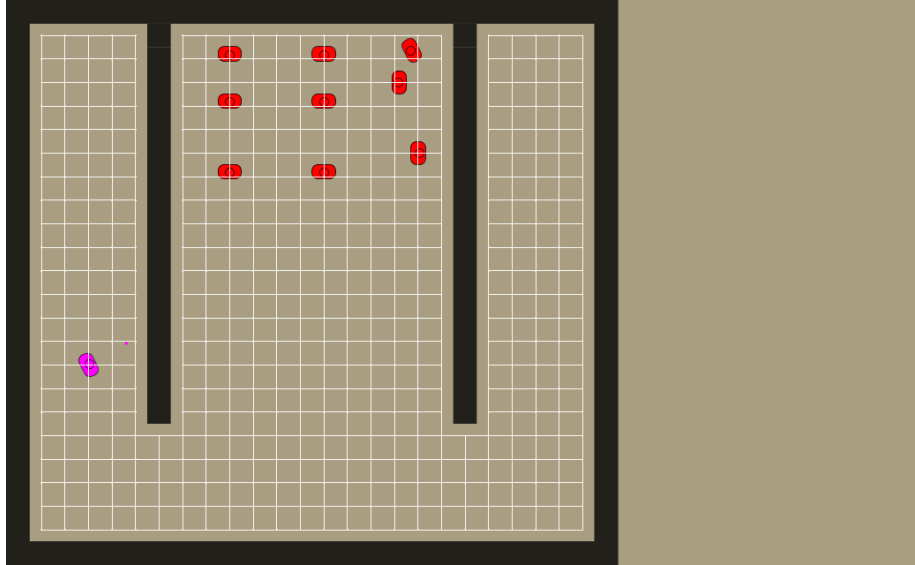


Figure 4: A* grid

6 Collision Detection

Since the character can rotate extra work was required for collision detection. For a given rectangle the angle of rotation around $(0,0)$ to make the rectangle flat, α , is found. To check if a point is within the rectangle the point is rotated around $(0,0)$ by α and then a simple check can occur because it's as if the rectangle was flat. An example is given in figure 5. The real rectangle is the magenta one and the projected rectangle is in cyan. The green square represents there is a hit.

An example of this I put together can be found here [Github Link](#). Note the squares are highly over engineer but the basic concept is still there.

The collision with the walls is detected with a square around the player because if they get too close they the player can turn and get themselves stuck

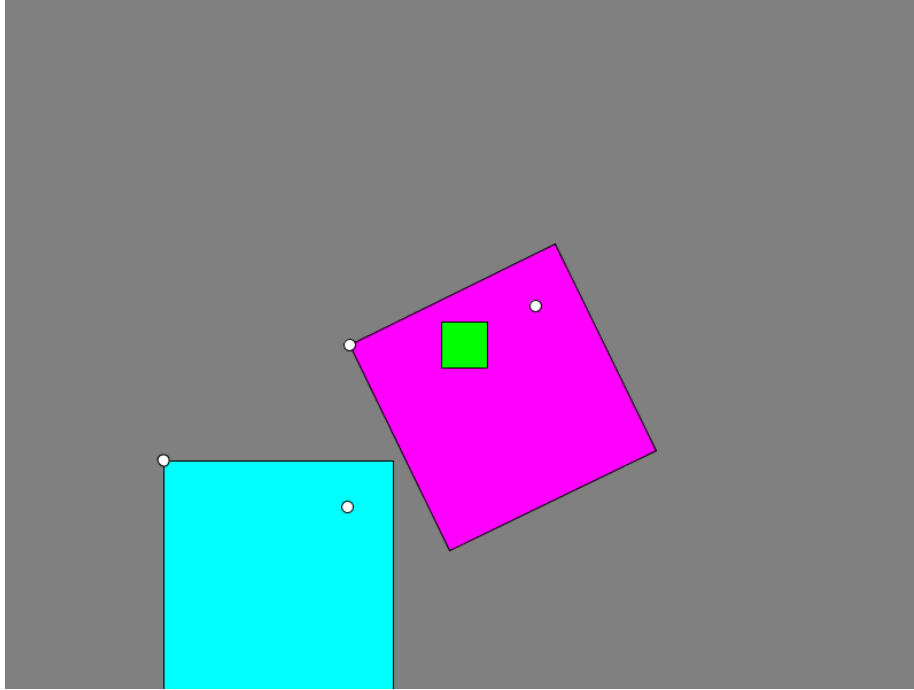


Figure 5: collision detection of a rotated rectangle

in the wall.

When bullets move their collisions are detected with a line from where they were to where they are going, This is because they can move quickly so I didn't want them to be able to go through objects.

7 Time Dilation

The time slowing up and down was quite easy to implement. The rate of time, or δt was calculated by taking the players current velocity over their maximum potential velocity. $\delta t = \frac{v}{potentialv}$ All Velocity are now multiplied by δt before they are applied and the time dilation is sorted.

8 Level Maker

Originally I wanted the level maker so that user could create levels are share them with their friends, increasing the replayability of the game greatly. This can be achieved but would require the user sharing the json files used to represent the level themselves. Ideally I would have like to have implemented the a server for sharing the levels but I felt the game itself needed more work before the

server became a priority. The full controls for the level maker can be found in the player guide. To place an object you go into the appropriate mode with the corresponding key then click where you want it to go. A potential object will be shown on the map and on release the object is placed. If the mouse is dragged around while clicked then extra results can be create, such as long walls or choosing the direction the character faces.

9 Platform

The platform of desktop because of the superior control it gives you with access to a keyboard. I don't think this game could be ported to tablet because the accuracy required would make the game unplayable.

10 Evaluation

I would say the actual game play works well works but one player did say that they would prefer if the controls were more like a tank steering game, I don't personally like this and none of the other testers mentioned it. Obviously the graphics are quite bad but that really isn't a strong suit of mine and would have taken time I didn't have to spare. Testers mentioned that the bullets went too fast so I did reduce the speed but it could benefit from further tuning. For the level maker the controls aren't very intuitive but they are simple and I think that is important.

11 Future work

I think one of the most important future extensions to the game would be a graphical overhaul to the menu system, it is clear but quite ugly. Other than that a replay of the game in 'real time' would be very cool to see, I think this could be achieved using the saveframe() function of processing. I would like to see more guns added, glass walls that can shatter and I think over the top blood effects would add to the game quite a lot.

12 Credit

All Music was created by Callum Fare.