Markus Takei

Byron’s Zombie Game Tech Specs

Game should require 1-2 weeks of programming to complete

There are no necessary pre-written pieces of code or libraries

Design assumptions, like the map, will be features of the game

Basically everything is a feature that will require some coding

Issues are listed in the table below where they appear

Difficult functions/items to code appear in the table

Shooting and killing zombies are linked functions. Moving around and running into environmental elements are also linked.

No features will be that complex or slow the game down too much; the sheer number of zombies spawned if they aren’t killed off quickly enough may slow the game down and cause some lag due to complexity

All variables will be local except for score, which will be like a private instance variable, so a global variable that is accessible throughout the program

Experimental feature will be linked to spawnZombie(), more zombies will be spawned during the 20 s interval (see table)

|  |  |  |  |
| --- | --- | --- | --- |
| Game Feature (Listen in order of priority (High-->Low)) | Requirements/Programming Difficulties | Pseudo-Code/Core Functions | Performance Impact |
| Moving Around | Keeping the speed of movement to a point where players can’t just run from zombies forever  If a player comes into contact with a wall or other landscape element, can’t be able to just go through it, similarly with zombies, can’t just go through them: will require knowledge of collisions | Functions: moveUp()  moveDown()  moveRight()  moveLeft()  (↑←↓→ , W A S D)  For player one, two respectively | Should not take long, simple movement functions dictated by arrow keys  Won’t affect performance |
| Players (sprites) - Most difficult element to program | Must have 2 separate characters, represented by two separate variables.  Two separate characters on the same map, controlled by different people - may be difficult to program  With another player comes another score count, and the need to compare  Collisions with environment, zombies, other players | var playerOne()  var playerTwo() | Will require different creation of two player objects, each with separate controls, movements, and weapons. Lots of different variables and things to keep track of for each player, so time complexity will be high  Won’t affect performance |
| Zombies (sprite) | Generating an endless stream of zombies, creating variation among each zombie’s look  When a zombie touches a player, the player will instantly die - makes it relatively easy b/c there are no lives to track  Understanding of collisions | Functions:  spawnZombie()  -spawns a zombie with one of maybe 3 different looks (all will be the same zombie underneath, with the same noises, speed, and movements, just a textural element)  addTwoZombies()  -adds two more zombies every round  move()  moves zombie towards the player(s), a constant movement; they will gravitate towards players at all time  killPlayer()  When a zombie touches a player, player will die | Zombies will have a very high spawn rate as level increases, and this may cause some lag due to complexity |
| Shooting | Understanding collisions, controlling the rate of bullet firing, aim control  Bullets will fire every 1.5 seconds  Will need to move player’s aim based off mouse control somehow; right now, Byron’s design just includes moving and shooting; there is no real way to aim  Player will have a little pistol in hand from which bullet objects will shoot, and eventually collide with something (background element of zombie)  A bullet that hits a zombie will instantly kill the zombie and remove it from the battlefield. | Functions:  shoot ()  Will need to have a delay between shots (1.5 s)  Aim ()  Use trackpad/mouse to aim | A relatively complex aspect of the game; there will need to be code for the moving bullet, a sprite/graphic for the gun, aiming, setting the 1.5 delay rate between shots, and the aspect of collisions as bullets hit and kill zombies. May impact performance slightly. |
| Landscape and barrier elements | Buildings, walls, and scattered objects like cars will need to prevent the passage of bullets; act as shields and shelter for the players. Will require understanding of immutable objects along with the concept of collisions. Will need to be randomly generated to mimic that eerie post-apocalyptic landscape that Byron wants. | Generate different element textures (buildings, cars, windows, traffic lights, etc)  generateBuilding ()  generateCar ()  generateWall () | Not time-consuming - randomly generated, non-moving objects - aesthetically important to give the game texture and environmental elements |
| Score | Will need to track player score; players have points acquired by staying alive (for every second that you’re alive, you get a point), and points acquired by killing zombies (1 kill = 1 point). In 2 player mode, players will compete to accumulate more points | int oneTimeScore ()  -tracks player one’s time score  int twoTimeScore ()  -tracks player two’s time score  int oneKillScore ()  -tracks player one’s kill score  int twoKillScore ()  -tracks player two’s kill score  int oneTotalScore ()  -a combination of player one’s time and kill score  int twoTotalScore ()  -a combination of player two’s time and kill score  Boolean determineWinner ()  If player 1 has a higher total score, return true. Otherwise, return false. Display a congratulatory message to player 1 if true, display one for player 2 if false | Int values track kill and time points, so they shouldn’t be that time-consuming. Comparing scores will take a bit more time, but overall, computing, displaying, and congratulating the winner shouldn’t be that complex. |
| Game Overlay/Basic escape and pause functions | Will need to display an instructions/resume/end game button  Prompted by the ESC key | displayInstructions ()  When player hits instructions, this function will display the instructions of the game as well as the objectives and controls  resumeGame()  endGame()  -both self-explanatory | A simple overlay function that will not affect game performance |
| Level Increases | Will need to add 2 zombies every time level is increased | increaseLevel ()  Increases level/wave of zombie attack once all zombies in a certain wave are killed | A necessary game element that will not affect speed or performance |
| Experimental Feature | Every 2 mins a player survives, there will be a larger wave of zombies  In multiplayer mode there will be a 20 s period in which players will be vulnerable to not only zombies but also each others’ bullets  Not sure how the extra zombies in this larger wave will be removed from the game if they aren’t killed in time, signalling the end of the larger wave. Maybe have them just despawn, but that adds another elements. This experimental feature needs a bit more thought. If a player is killed by the other player, will the surviving player win, ending the game, or will they continue playing in single player? | Will use spawnZombie()  Call the function more to spawn more zombies  makePlayerVulnerable () makes player vulnerable to each others’ bullets  Time ()  Keeps track of time - if 2 mins pass, commence the big wave and keep it going for 20 s | Seems to be an easy feature to add; just spawn more zombies  However increased player vulnerability may add another dimension to the game that causes lag or affects performance |