Ethan Hodess

10/21/19

Technical Specification for Anjee’s Game

|  |  |  |  |
| --- | --- | --- | --- |
| Item Name | Performance Impact | Difficulties | Functions |
| Person sprite | This person is controlled by the player. It is the only controllable sprite in the game. The player controls the jumps to get to all the platforms. WASD to control movement, space for powerups, W to jump. | Dynamics of jumping (hold key to jump higher/farther)  Collisions with coins, powerups, enemies, the ground | Jump  Use powerup  Lose life  Gain life  Die  Pick up coin |
| Flying enemies | They will fly across the screen, and if the person touches them, the person loses a life. | Randomly generating when and where the enemies will fly in | Fly in |
| Standing enemies | They stand on the platforms and if the person touches them they lose a life. | Must follow the path of the platforms because it is standing on it | Generate enemy |
| coins | It is a collectable for the player and is randomly generated. It sits on the platforms. | Collisions with person to pick up | Generate coin  disappear |
| Power ups | There are two types of power ups -- 5 second invincibility and double jumps. They are randomly generated | Collisions with person to pick up | Generate double jump  Generate invincibility  disappear |
| multiplayer | Have a second person on the screen who uses arrow keys and shift button instead of WASD and space. | Creating overlay for second person to keep track of lives, coins, etc | none |

1. I don’t think this game requires any special libraries outside of Java script. Pre written code may include code for collisions between the player and coins, power ups, and ground. One design assumption that may be an entire feature is that the levels are supposed to be randomly generated.
2. The first thing to get done is the player controls since that is the basis of the game. The dynamics of the jumping might be a little tricky. After player controls are done, then coding the randomly generated platforms would be next. The maximum space between the platforms has to be set so that the player doesn’t face any impossible jumps. Next would be adding in the coins and power ups, and for power ups there has to be a function to let the player use them which would change some of the game mechanics. Lastly would be multiplayer support and game overlay.
3. Feasibility analysis
   1. The most difficult items to program is designing the jumping mechanics where holding the jump key makes the player jump higher and farther than just tapping the key. Also designing the randomly generated platforms in a way in which the game is actually possible seems difficult.
   2. I think code such as for collisions with power ups, coins, and the ground can be found online, so the jumping mechanics and platforms will definitely take the most time, and especially as they are related to each other. Programming definitely won’t be super fast but I don’t think it’s overly complicated either.
   3. The one issue I’m thinking of right now is how to ensure that every level is beatable since the levels are randomly generated.
4. Data storage and variables

|  |  |  |  |
| --- | --- | --- | --- |
| Variable name | Structure | Global or local | Description / functionality |
| coins | int | local | Counts number of coins |
| Double jump | int | local | Counts number of double jump power ups |
| lives | int | local | Counts number of player lives |
| invincibility | int | local | Counts number of invincibility power ups |
| Level speed | int | global | Stores speed of level, which increases with each level |
| Level number | int | global | Keeps track of what level the person is playing |

Data structures for this game: array

1. Functional code relational map
   1. Start - starts level, generates platforms, coins, power ups,

Collide - called when the person touches a coin, ground, enemy, or a power up to trigger a subsequent action

Die - all objects such as platforms, enemies, coins, power ups go away and the game level is reset to the beginning of the same level, person loses 1 life

Win level - game stops and is reset, level increases by 1, person gains 1 life

Lose - if lives go to 0, then the game resets to level 1 and person goes to 3 lives

Use power up - number of power up goes down by 1, person gets power up

boost according to which power up was used

Conclusion

I think that this game is definitely feasible. If there is one feature that I think may come into question, it is that the levels are fully randomly generated. This part of the design may need to be reviewed since I don’t know how to ensure that the game stays possible given random levels. However, maybe taking measures to significantly reduce the chance of the player encountering an impossible jump would be enough. Other than that, the other features of the game are pretty simple. There are not too many objects to keep track of, and the controls for the player are only movement. Also, for this game the overlay and design can be quite simple, so the game should look clean. I anticipate that the coins and power ups are not overly complicated either. I think that once the randomly generated platforms are done, the rest of the game is not as difficult as that. Overall, I think that the random feature may need to be tweaked, but the rest of the game is feasible.