Cave Age

Made by Justin Kephart

Music by Kevin MacLeod

Version. 1.0.0

Game Design Technologies

Spring 2018

Tim Samoff

Abstract

Cave Age is a side-scrolling platform game set in prehistoric times. You play as Dave, the caveman, as he journeys through a treacherous cave system. Dave can jump, climb walls, and throw spears at enemies. The goal is to reach the exit, gaining as many points as possible along the way, by killing enemies and collecting bones.

Enemies are smart but predictable, if you keep your eye on them. Puzzles are relatively easy, but the enemies won't allow you to progress so easily. There are secret areas that may require backtracking for a chance to increase your total score. Players have a chance to submit a high score at the end.

Design Statement

As the game is supposed to be set in prehistoric times, I had to design enemies and environments that would make sense for that time period. No magical or high tech moving platforms or spikes. No, Dave has to instead contend with boulders and tar pits. Instead of a key, I had Dave burn away leaves with a torch. This mechanic allowed for several opportunities, like a level that burns away around you.

Audience and Context

Cave Age is an action-platforming game. Some skill is required to beat the game, as it is mostly focused on combat, and the enemies are smart. I recommend this to players who like challenges, and overcoming tough enemies through prediction and skill.

Core Gameplay

The gameplay is focused around platforming and combat.

You can pause at any time using escape, and you can switch to fullscreen with F11.

Platforming

Dave is not only able to jump high, but he is able to jump off of walls in a wall jump. Wall jumps do not allow scaling a single wall on its own. You must keep kicks off opposite walls to climb, or jump on a platform.

In addition to wall climbing, there is platforming using one-way platforms. You can drop through them if you choose, and they do not hold up enemies.

One of the "enemies" is simultaneously dangerous for Dave and useful for platforming. If you find a giant boulder, you can ride on top of it safely across otherwise lethal tar pits. Just don't fall in front of the boulder, or Dave will be killed.

If you have a torch, you burn away the first leaves you touch, regardless of intent. This fire will spread to nearby leaves until the chain of leaves is burnt through. This is used in a level as a time limiting mechanic, and is also used to hide secrets.

Bones are like coins in that they add to your score when collected. They can be hidden away by tough enemies and/or secret areas.

Skulls are checkpoints and will save your progress in case of death. They actually create a save state using a buffer, so you don't have to worry about losing anything on death.

The end goal of a level is the cave exit. This will transition you to the next room in the list, or the credits menu.

Combat

Dave can kill bugs and bats by simply jumping on them, but that can be harder than you think, as bugs can crawl on walls, and bats can fly.

The best way to kill bugs and bats is to find a spear. A spear can be thrown and retrieved again afterwards. If the spear is destroyed, either by falling in tar or off the map, it respawns at its starting point.

Killing enemies earns you 500 points. That is 5x as much as a bone. As such, killing as many enemies as possible is advisable. It also stops them from sneaking up on you further on.

Boulders cannot be killed, but can be ridden.

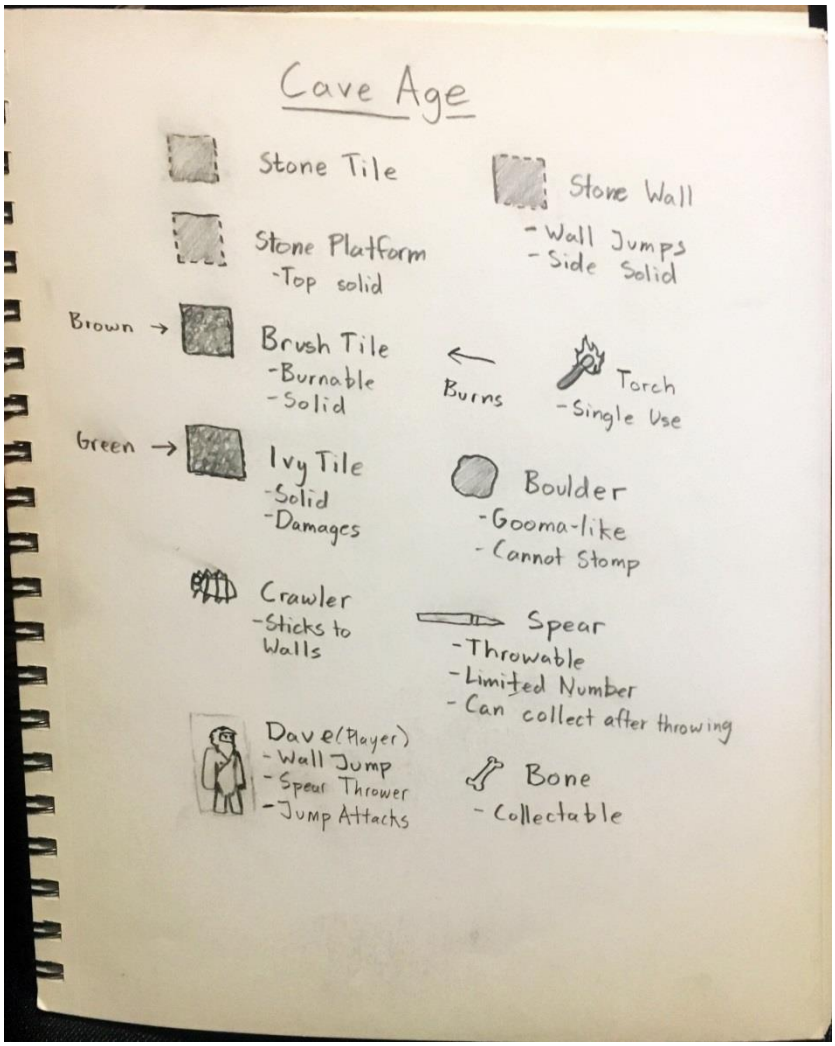
Warning: Dying resets all progress back to the last checkpoint or level start, and subtracts 500 points from your score.

Combat requires high precision as spears are lost when thrown. Enemies are also smart. Bugs will drop on you if they are above you on the ceiling. Bats will stop and charge directly at you.

Look and Feel

Cave Age intends to reflect prehistoric times by using alternatives to typical platforming tropes.

Original Design



I deviated a lot from the original design after finding out what game maker could do.

The stone tiles are the same object as the wall now. Instead of a brush and ivy, I just have leaves and tar pits, as I have an instant death system instead of a health based one.

When designing this, I was thinking of alternatives to the typical spikes, keys, and jumping mechanics.

Player Sprite

Here's Dave. I custom made his textures and animations in Game Maker, using the built in editor. Honestly I much prefer animating in flash compared to this, but it's usable. He wearing very basic caveman garb, and has a bushy ungroomed beard. The change in design was mostly due to pixel constraints (most of the blocks are 32 x 32 pixels), but I like the new design.

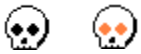
He has a sprite or animation for just about every action: jumping, wall jumping, falling, dying in tar, etc. And he has a separate set of sprites for when he has a spear as well.

Enemies

Bug ***Bat*** **Boulder**

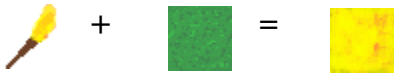
I tried to pick enemies that made sense for a cave based game. When searching for "Cave Creatures" on google, I really only found bugs and bats. I added the boulder for the "goomba-style" enemy, but I wanted something useful for the game as well.

Checkpoints (Skulls)



So I didn't think a magical floating orb fit the theme of my game very well, so I reskinned the checkpoint to a skull, since skulls are found in caves a lot (lots of rituals took place in caves)

Key and Door (Torches and Leaves)



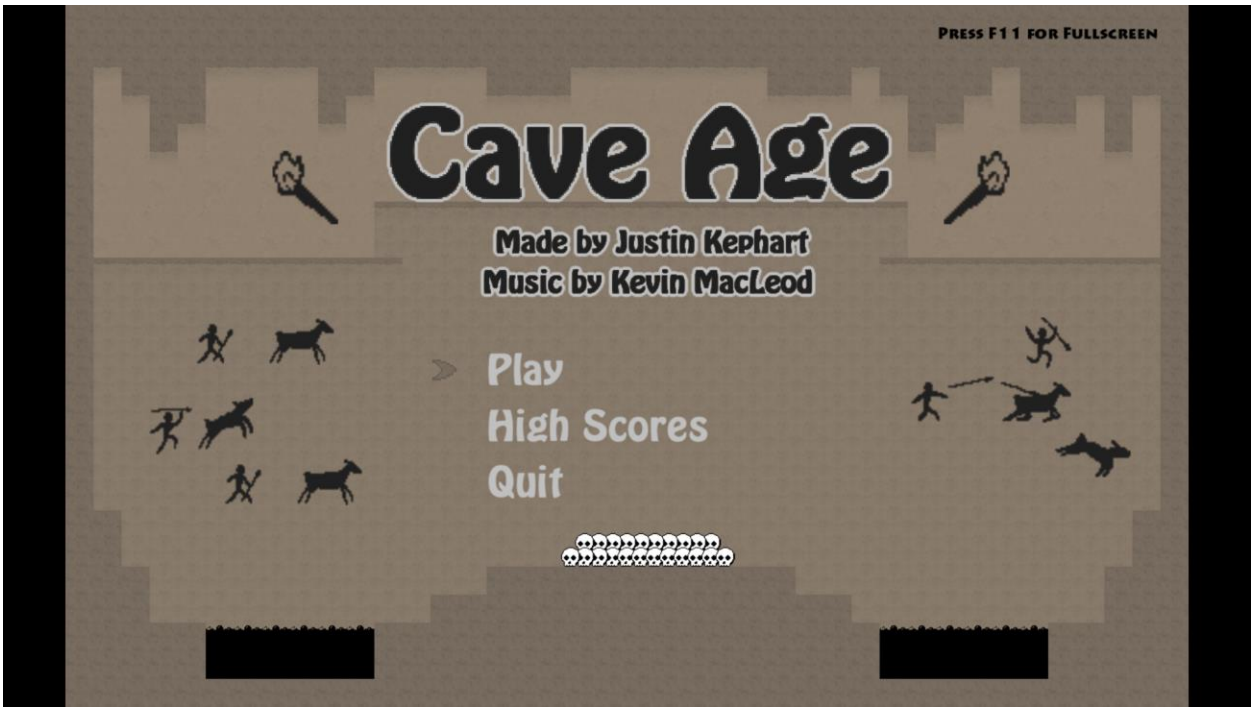
Same as above. Cavemen didn't have locks and keys, but they were famous for fire.

Powerups (Spears and Bones)

When I think of cavemen, I think of spears. The reason why I implemented one was because I thought bugs would be too difficult to jump on if they were on walls.

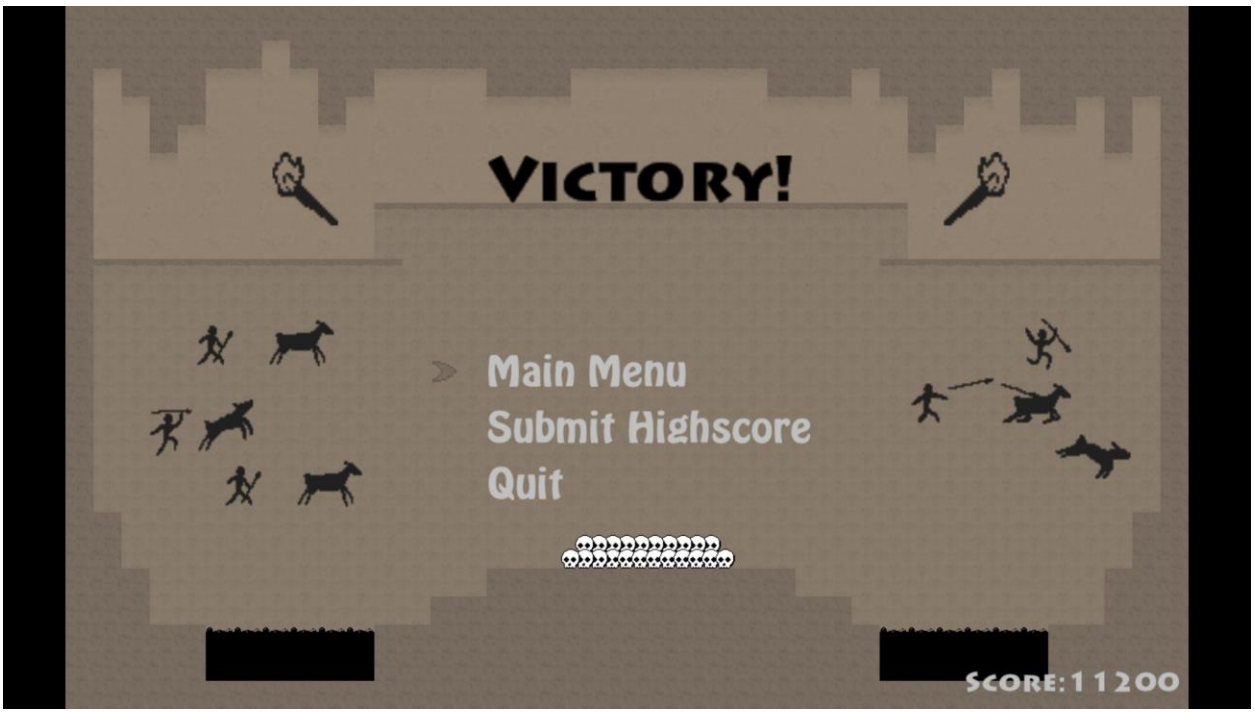
Bones made more sense than a coin, at least to me. Bones had a lot of utility back then. You could make tools from them, and eat them if you were hungry enough.

Main Menu

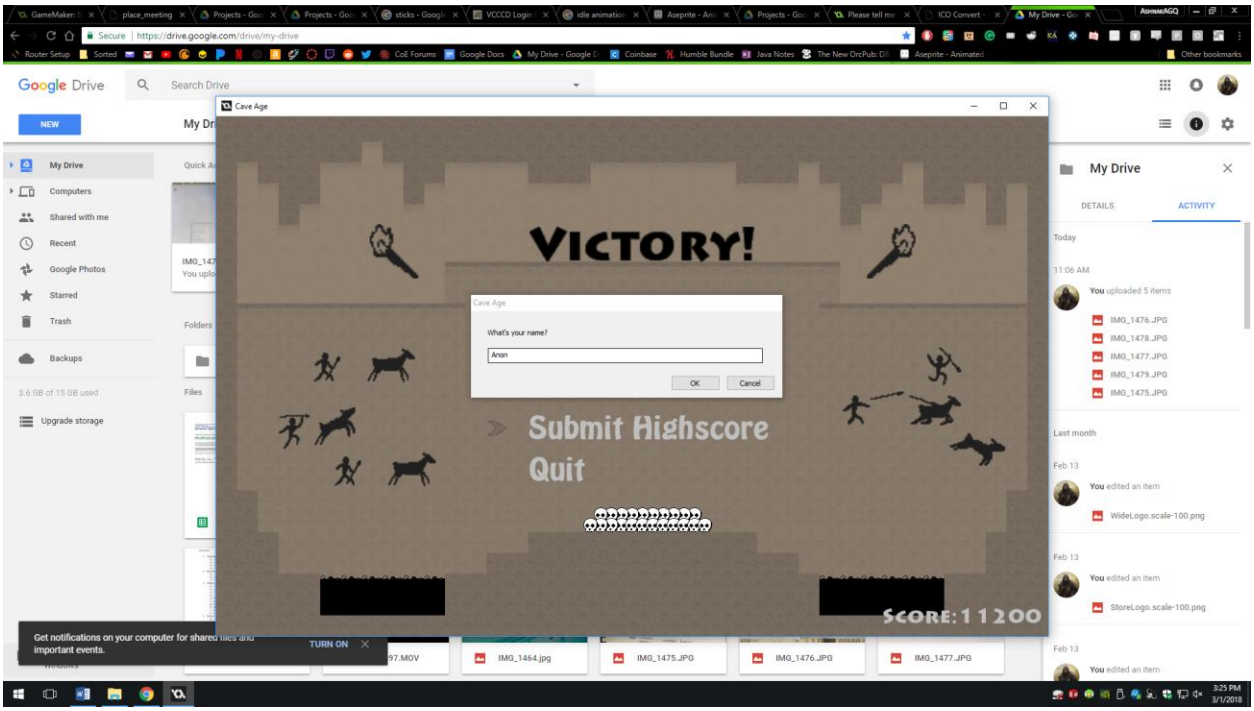


I wanted to show off the art I worked hard on to make, so I have enlarged versions of my cave paintings in the background, and a pile of skulls to set the mood for the game. The Game Complete menu is basically the same background.

Game Complete Menu



High Scores



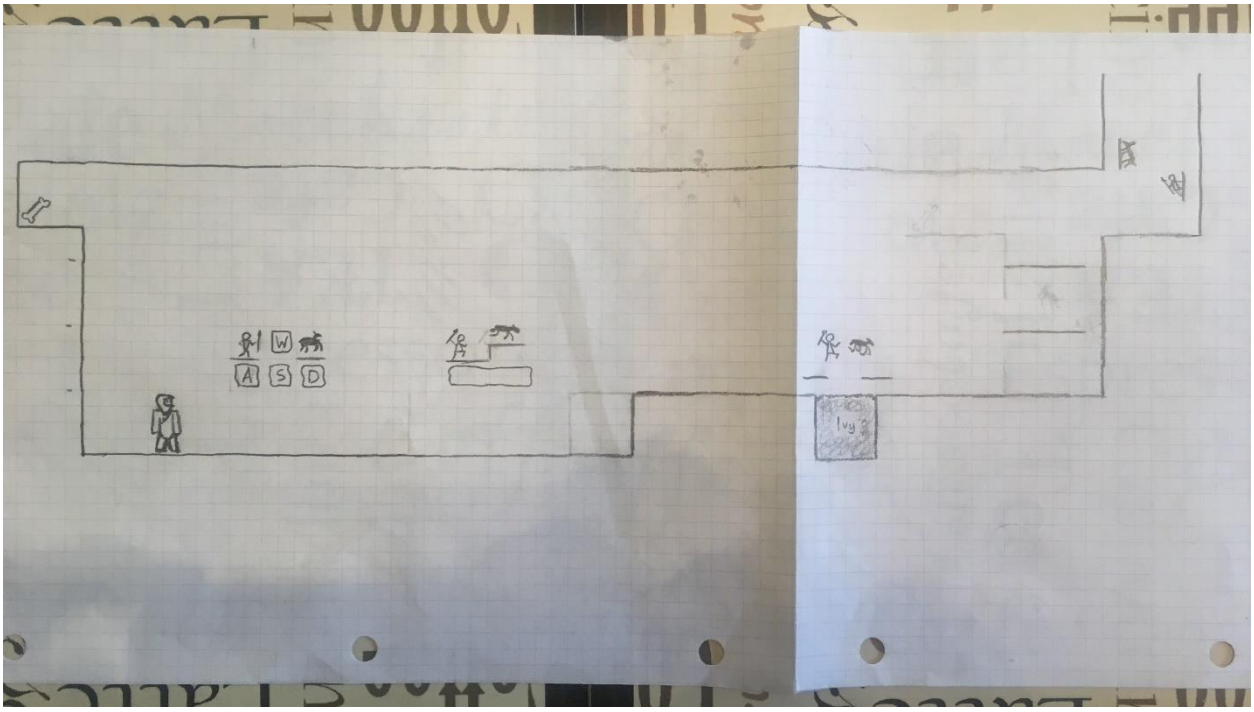
I feel like every game needs a goal or trophy. For a game like this, a high score screen is perfect.

Blocks

I made all the textures in this game from scratch in either Photoshop or Game Maker itself. It took sooo many iterations to make rocky textures that tiled well, were not too noisy, and could be differentiated from each other, but still looks like good stone. They really set the color pallet of the game. I tried using google images to find a good color for them, but it always looked nasty in pixelated form. Eventually I got to these colors by messing around with the color selector. I used the Fade tool in Game Maker to quiet the noisiness of my blocks, and colorize them.

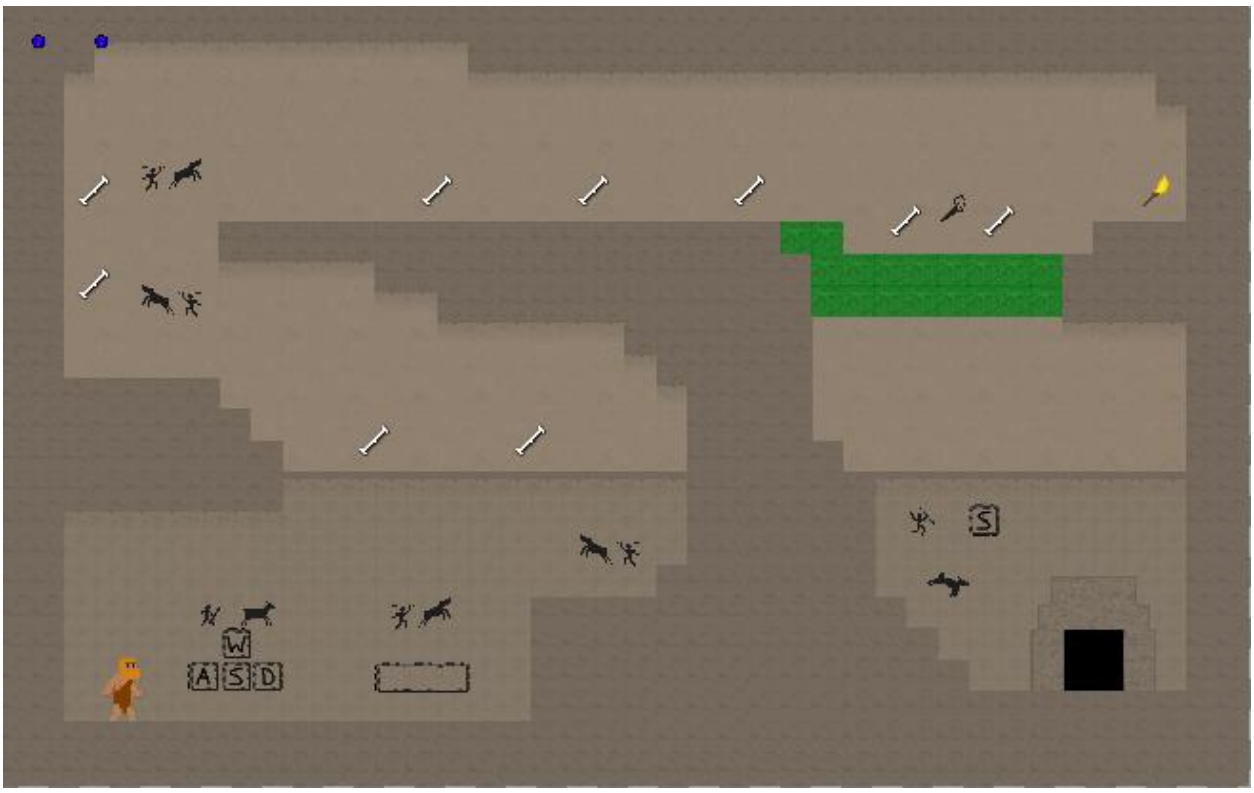
Maps and Game Flows

Level 1 Draft



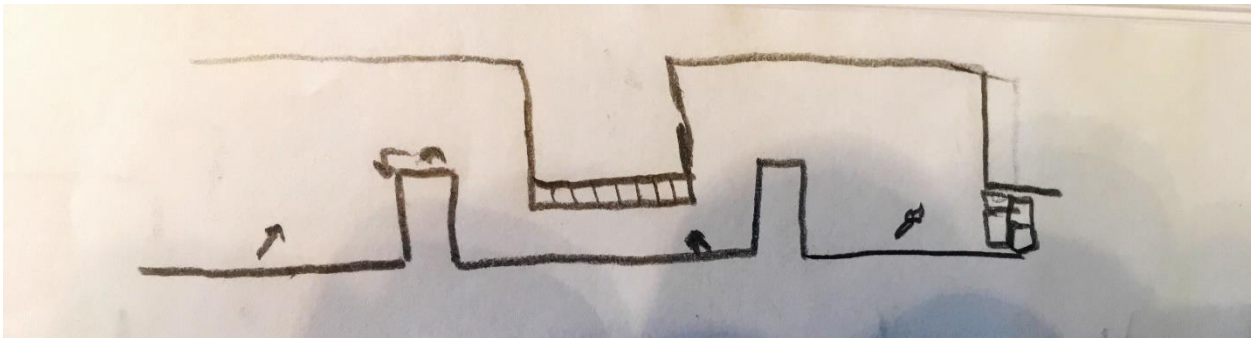
This is the only original design I fairly kept too. The idea was to slowly introduce the character to mechanics of the game through example and cave paintings. I immediately scrapped the ivy idea after making this. I decided to implement all my mechanics before working on level design, as I was not sure how much and what was possible to make. I may have gotten a little carried away though.

Level 1 Final Changes



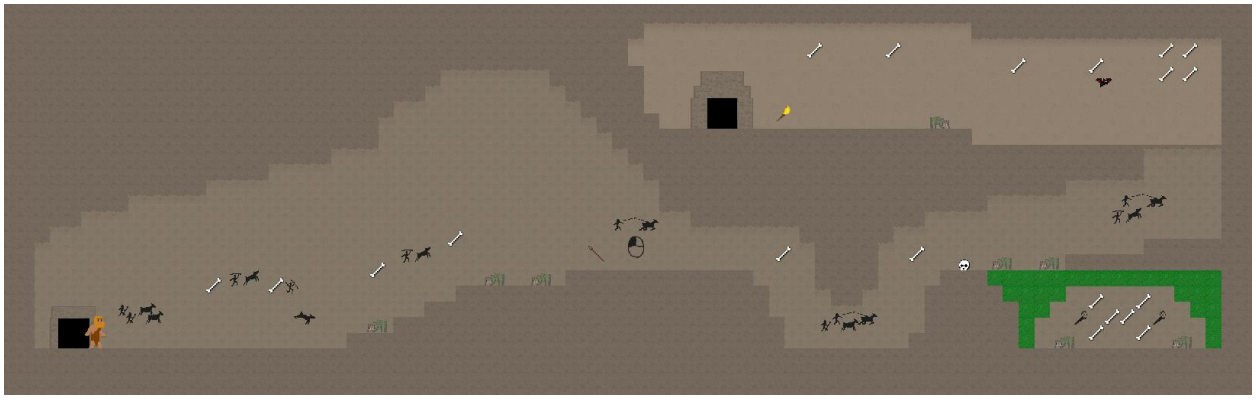
So like I said, I made a lot of changes to the level design while implementing it, as I had a lot more mechanics to draw from. I wanted to introduce the player slowly to the game. What better than a room you can die in, that teaches you the basic mechanics?

Level 2 Draft



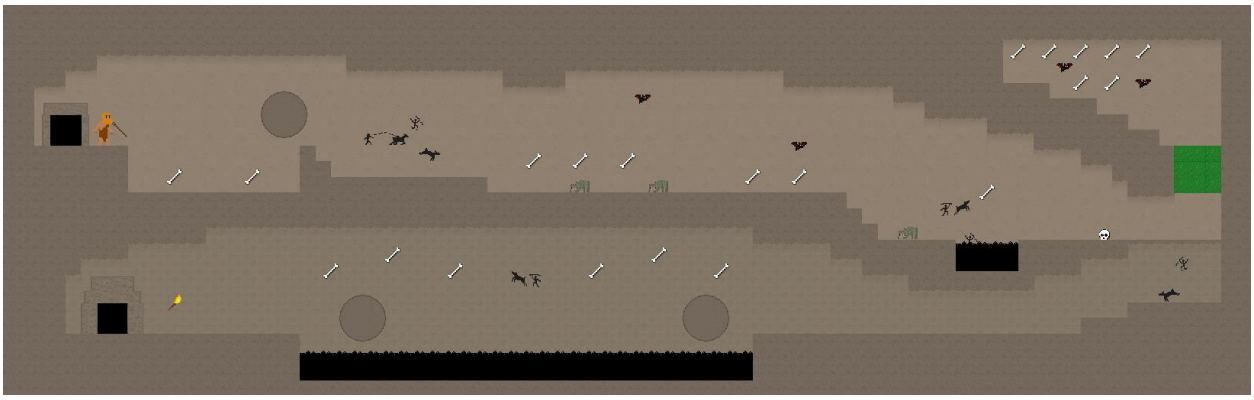
I made this draft (most of them actually) in my political science class. It's where I get my best work done. The goal here was to force the place to use the spear by trapping them in a narrow corridor.

Level 2 Final Changes



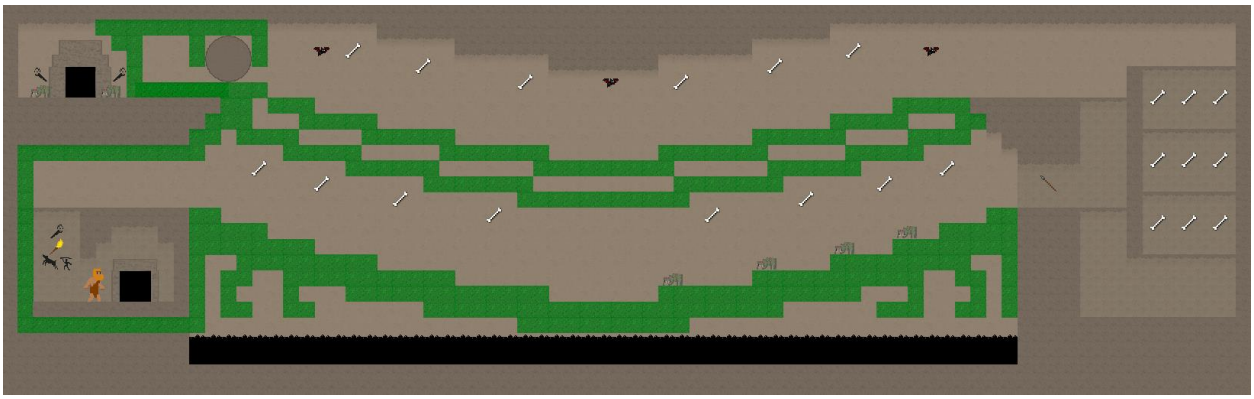
I am introducing the enemies now, so I thought I'd include some enemies to jump on first. I added a secret using the torch I though the player how use in the previous level.

Level 3 Final Changes



I made this level because I came up with the boulder and tar mechanics. I thought a clever way to combine them was to ride the boulder over the tar pits. I placed a boulder in the beginning so the player could figure out its mechanics. Also, players must cross the bottom tar bit three times to get the secret.

Level 4 Final Changes



I made this because of the ivy mechanic. The timing was too hard to get on paper. Basically, every time I made a mechanic, I made a level around it, compounding from the previous levels. As you can see there is also a boulder and tar pit in this.

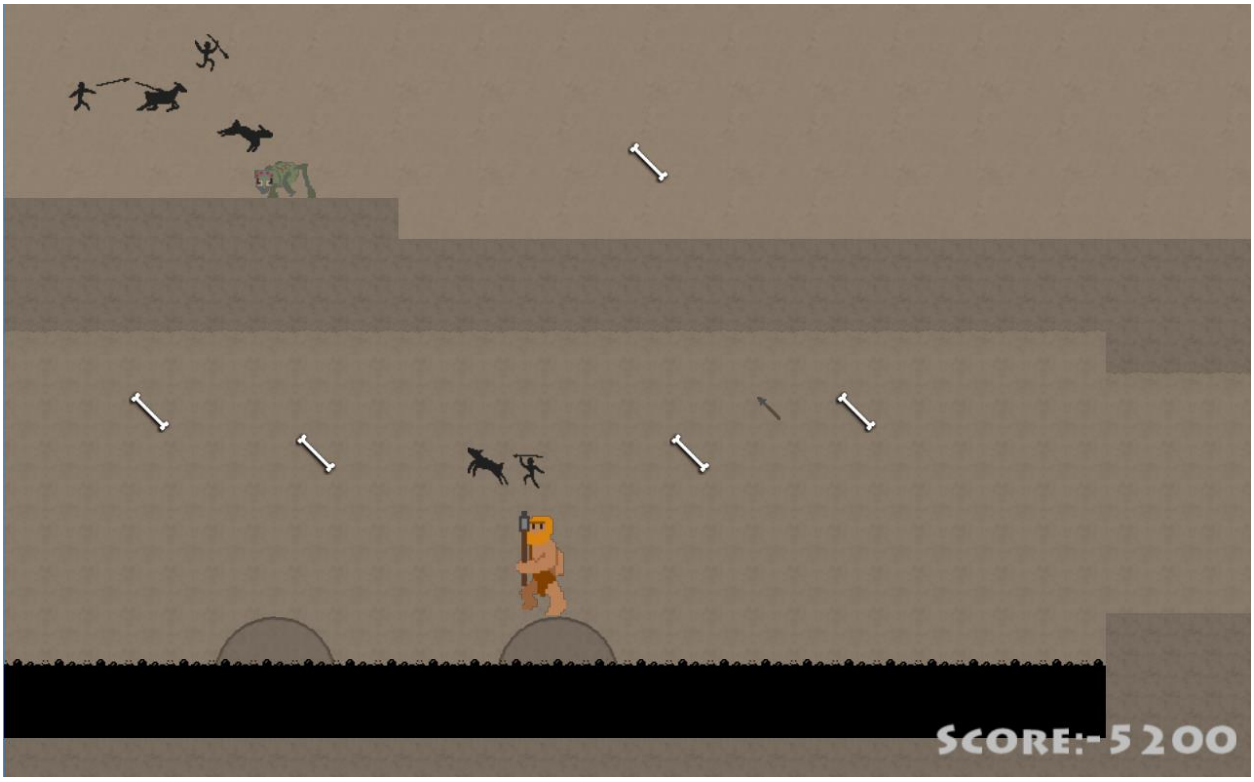
Gameplay Scenes

Teaching the player to wall jump (level 1):

Holding down the left mouse button draws the path of the spear (level 2):

Levels get harder as the amount of enemies increases (level 3):

Floating bones provide extra points for the brave… but at great risk (level 3):



This level burns around you as you try to beat it, effectively (level 4):

Conclusion/ Future

This game has been a blast to make. I love making mechanics and seeing them finally work. It took a lot of debugging and trial and error to get this game going.

The first challenge of this game was the platforming. I have implemented a lot of stuff that wasn't taught to us in class. First I made a system so you can't just hold jump to keep jumping. Then I made used that system to allow jumping on walls. At first I made my own code for my one way platforms, and even shared it with other students in class. But it was very buggy, so I followed a tutorial that helped me refactor it. Of course that just introduced a bunch of other problems. The other guy's solution, instead of checking the type of platform in the collision code, was just to remove the mask of the platform when the player was below it. This of course mean that enemies would fall through them as well, and could even get stuck, if they were halfway when I was jumping, so I had to do a lot of work with parenting. I decided to make abstract objects to specify what range of objects an object interacts with. That also allowed me to add hard exceptions for certain types of platforms.

The second challenge was getting the movement right. I had a multitude of factors affecting the rate of change of my speed, so I introduced acceleration to my object, which required refactoring a lot of stuff. Actually, probably a third of my time programming was probably done refactoring stuff to allow for more flexibility and new features. Once there were a ton of new variables affecting my character's speed that increased the complexity of tweaking the movement to something appropriate for my game. I made a test world with increasing platforms and walls to test out my movement until I settled on to what it is today.

The third challenge was implementing all the features that I planned in my initial design, like boulders, torches, and fire. That was relatively easy, but not painless. A lot of time was spent in the debugger as objects would teleport around or set fire to the wrong objects due to bad collision coding.

Level design was mostly straightforward, other than the fact that I didn't know what damn resolution to set the game window to. My textures would either get stretched or the game was too zoomed in. I eventually found a good balance.

Speaking of textures, making all of those myself was a pain. I am not an artist, and it took a long while to find a color palette that worked. The fade too helped *immensely*.

The biggest challenge for me was the save system. I need to use buffer type saving if I wanted to implement a score system, which introduced a lot of complications. With the buffer based system, I lost all changes to variables after the save, including global variables. The fake mouse I made, for instance, would jump across the screen after dying, which is bad IMO. I came up with a system to write to an ini file and check it for changes periodically (It means the game reset). In order for that to work though, I had to have it activate a timer, cause the game loads a few frames after reading and writing to the file, meaning it would revert the file, then load the save, losing my changes from the file.

I still have a bunch of features that I would have liked to have implemented in my game, but wisely decided to only add them if I had time. (I barely made it by 1:00 PM *without* sleeping).

For example, I was thinking instead of the spear gaining velocity based on the distance to the mouse, I could hold it town to charge it.

Also I would like to add special effects to my character movement, like dust particles and such, but I know such things take time and a lot of effort.

I would also like more variety in my enemies and other hazards, but the amount for now is sufficient for a vertical slice. I loved coding the bat's homing effect (not so much the spider's wall climbing. That took me weeks to figure out, as I wanted it to work with a non-square spider.)

This game currently is good enough to put on my portfolio, which is fantastic. Thanks again for the opportunity. I would otherwise not have time or the motivation to deal the Game Maker.