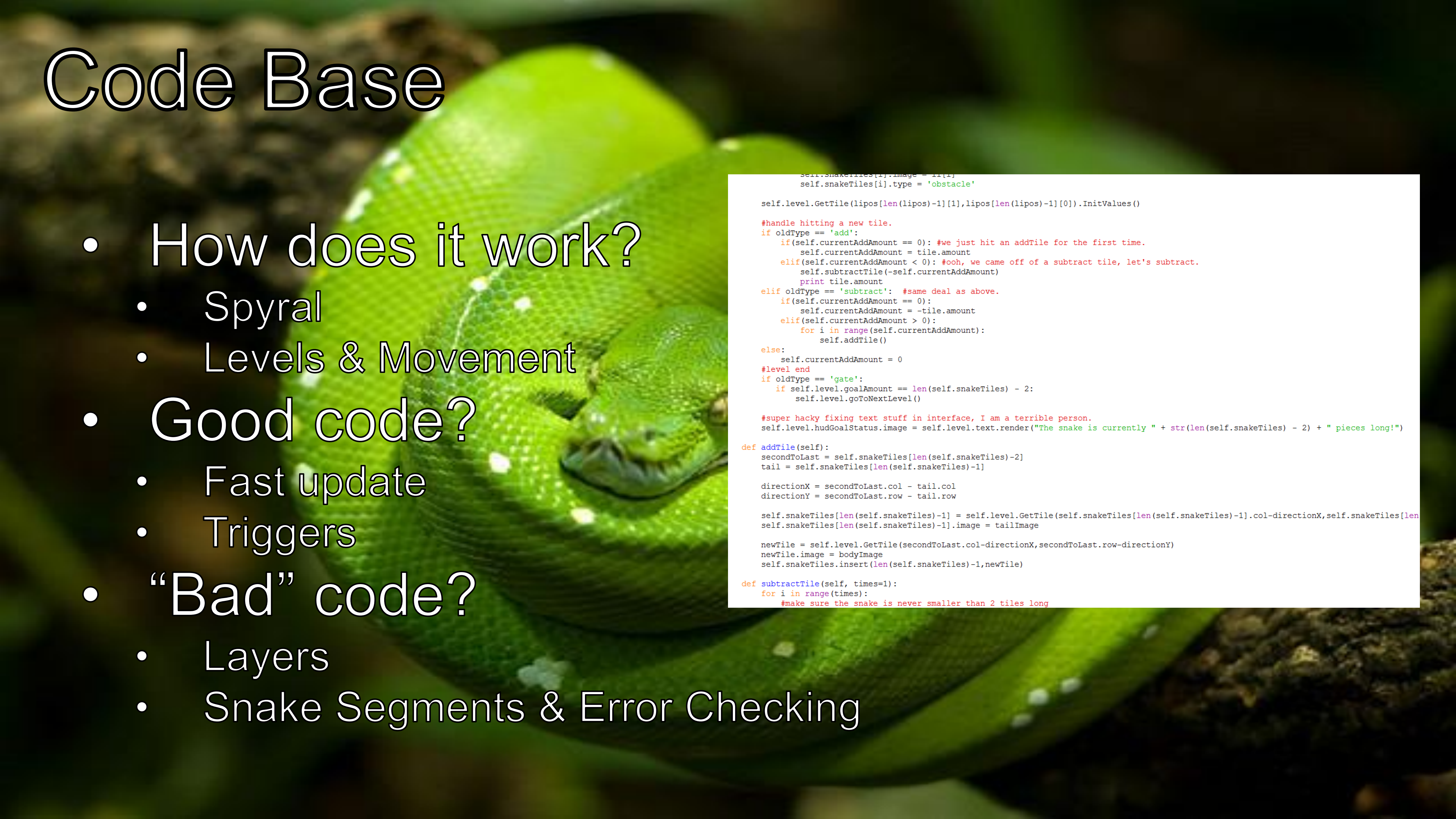
The snake needs to be 10 pieces long to go to the next level!
The snake is currently 2 pieces long!

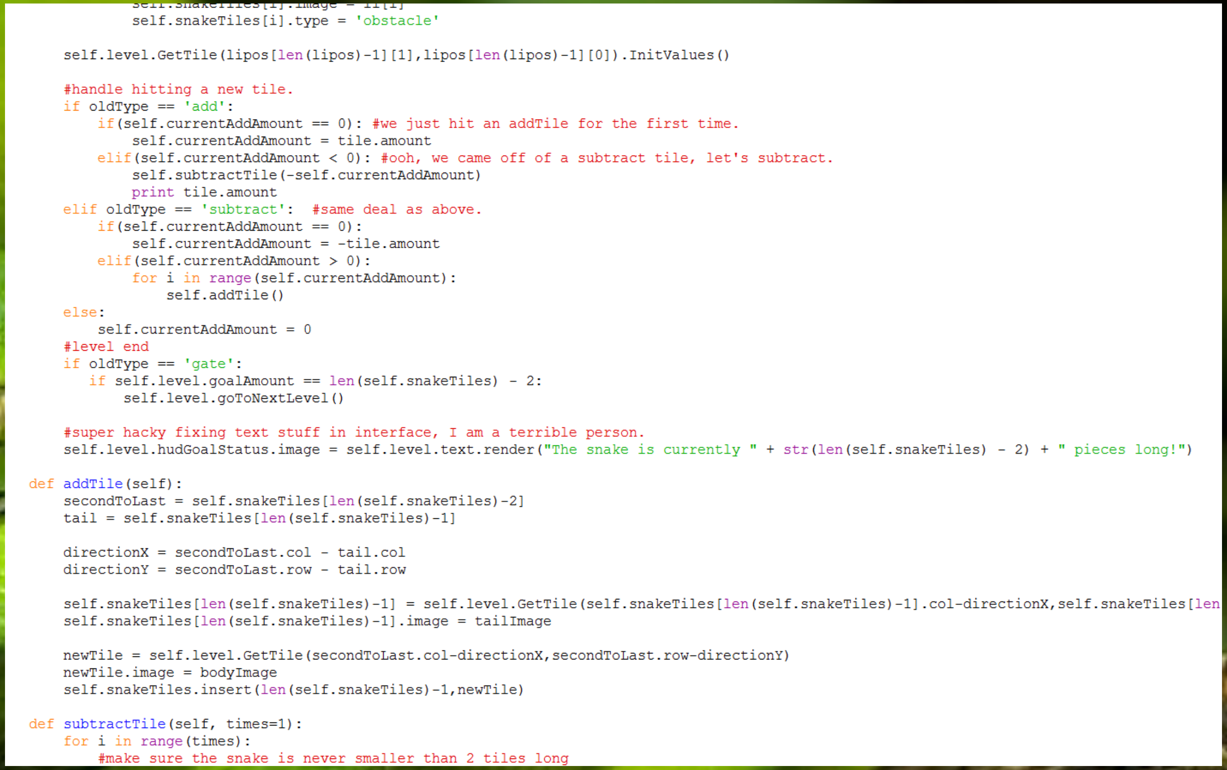
Python Math
Adder's Garden Adventure

danShumway
Pharas
mstubinis
DJ_Mark



Code Base

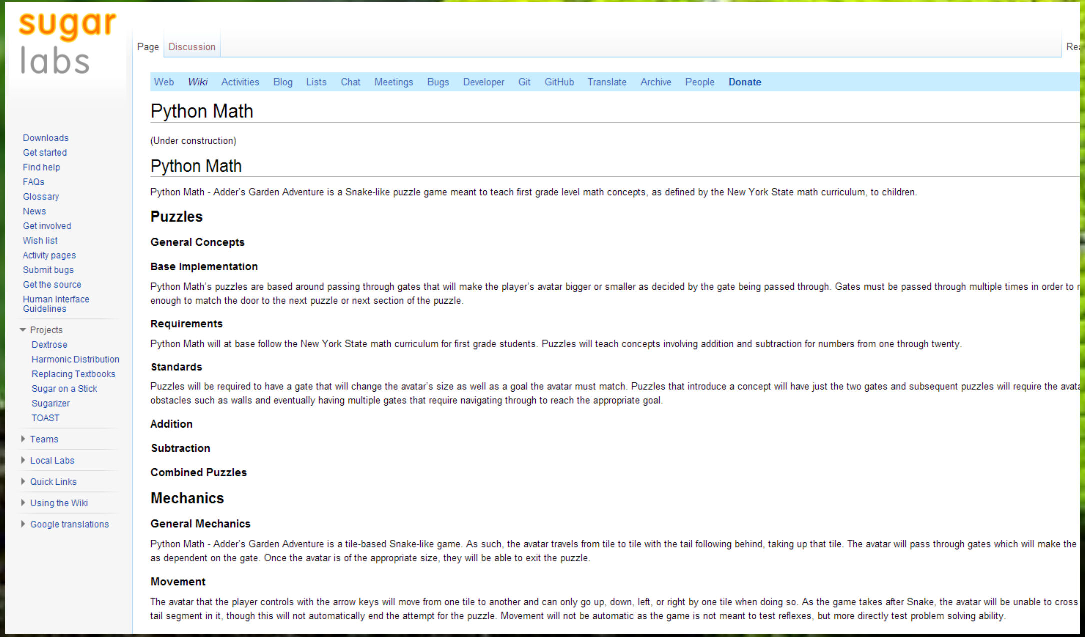
- How does it work?
  - Spyral
  - Levels & Movement
- Good code?
  - Fast update
  - Triggers
- “Bad” code?
  - Layers
  - Snake Segments & Error Checking



```
            self.snakeTiles[i].image = li[i]
            self.snakeTiles[i].type = 'obstacle'

    self.level.GetTile(lipos[len(lipos)-1][1],lipos[len(lipos)-1][0]).InitValues()

    #handle hitting a new tile.
    if oldType == 'add':
        if(self.currentAddAmount == 0): #we just hit an addTile for the first time.
            self.currentAddAmount = tile.amount
        elif(self.currentAddAmount < 0): #ooh, we came off of a subtract tile, let's subtract.
            self.subtractTile(-self.currentAddAmount)
            print tile.amount
    elif oldType == 'subtract':  #same deal as above.
        if(self.currentAddAmount == 0):
            self.currentAddAmount = -tile.amount
        elif(self.currentAddAmount > 0):
            for i in range(self.currentAddAmount):
                self.addTile()
    else:
        self.currentAddAmount = 0
    #level end
    if oldType == 'gate':
       if self.level.goalAmount == len(self.snakeTiles) - 2:
           self.level.goToNextLevel()

    #super hacky fixing text stuff in interface, I am a terrible person.
    self.level.hudGoalStatus.image = self.level.text.render("The snake is currently " + str(len(self.snakeTiles) - 2) + " pieces long!")

def addTile(self):
    secondToLast = self.snakeTiles[len(self.snakeTiles)-2]
    tail = self.snakeTiles[len(self.snakeTiles)-1]

    directionX = secondToLast.col - tail.col
    directionY = secondToLast.row - tail.row

    self.snakeTiles[len(self.snakeTiles)-1] = self.level.GetTile(self.snakeTiles[len(self.snakeTiles)-1].col-directionX,self.snakeTiles[len
    self.snakeTiles[len(self.snakeTiles)-1].image = tailImage

    newTile = self.level.GetTile(secondToLast.col-directionX,secondToLast.row-directionY)
    newTile.image = bodyImage
    self.snakeTiles.insert(len(self.snakeTiles)-1,newTile)

def subtractTile(self, times=1):
    for i in range(times):
        #make sure the snake is never smaller than 2 tiles long
```

Documentation

- SugarLabs Wiki
- Design Docs & Process



Python Math

(Under construction)

Python Math

Python Math - Adder's Garden Adventure is a Snake-like puzzle game meant to teach first grade level math concepts, as defined by the New York State math curriculum, to children.

Puzzles

General Concepts

Base Implementation

Python Math's puzzles are based around passing through gates that will make the player's avatar bigger or smaller as decided by the gate being passed through. Gates must be passed through multiple times in order to r enough to match the door to the next puzzle or next section of the puzzle.

Requirements

Python Math will at base follow the New York State math curriculum for first grade students. Puzzles will teach concepts involving addition and subtraction for numbers from one through twenty.

Standards

Puzzles will be required to have a gate that will change the avatar's size as well as a goal the avatar must match. Puzzles that introduce a concept will have just the two gates and subsequent puzzles will require the avata obstacles such as walls and eventually having multiple gates that require navigating through to reach the appropriate goal.

Addition

Subtraction

Combined Puzzles

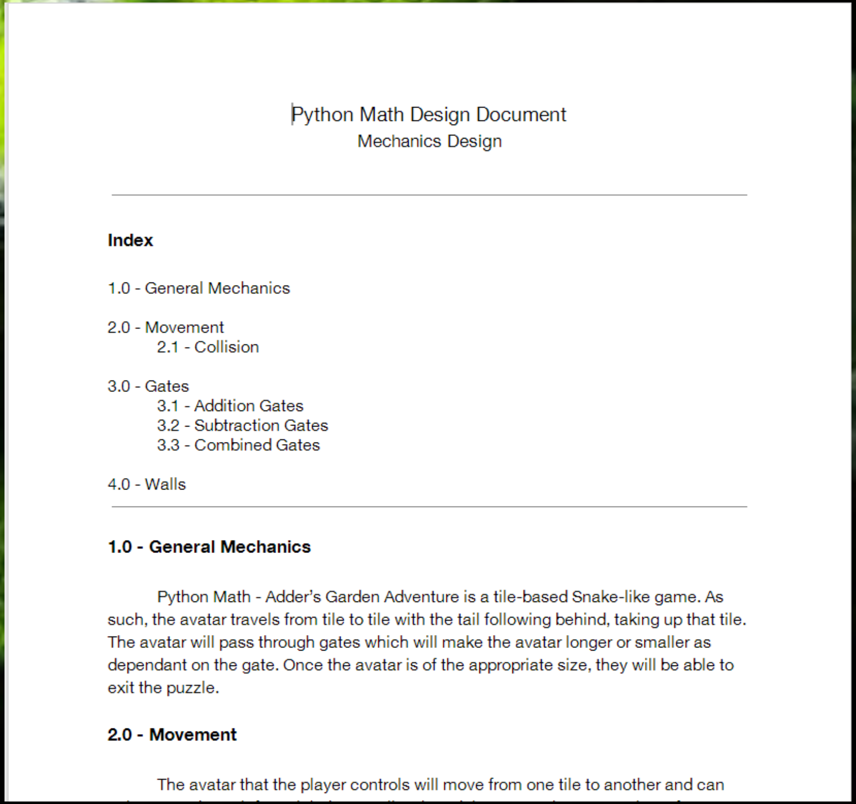
Mechanics

General Mechanics

Python Math - Adder's Garden Adventure is a tile-based Snake-like game. As such, the avatar travels from tile to tile with the tail following behind, taking up that tile. The avatar will pass through gates which will make the as dependent on the gate. Once the avatar is of the appropriate size, they will be able to exit the puzzle.

Movement

The avatar that the player controls with the arrow keys will move from one tile to another and can only go up, down, left, or right by one tile when doing so. As the game takes after Snake, the avatar will be unable to cross tail segment in it, though this will not automatically end the attempt for the puzzle. Movement will not be automatic as the game is not meant to test reflexes, but more directly test problem solving ability.



Python Math Design Document
Mechanics Design

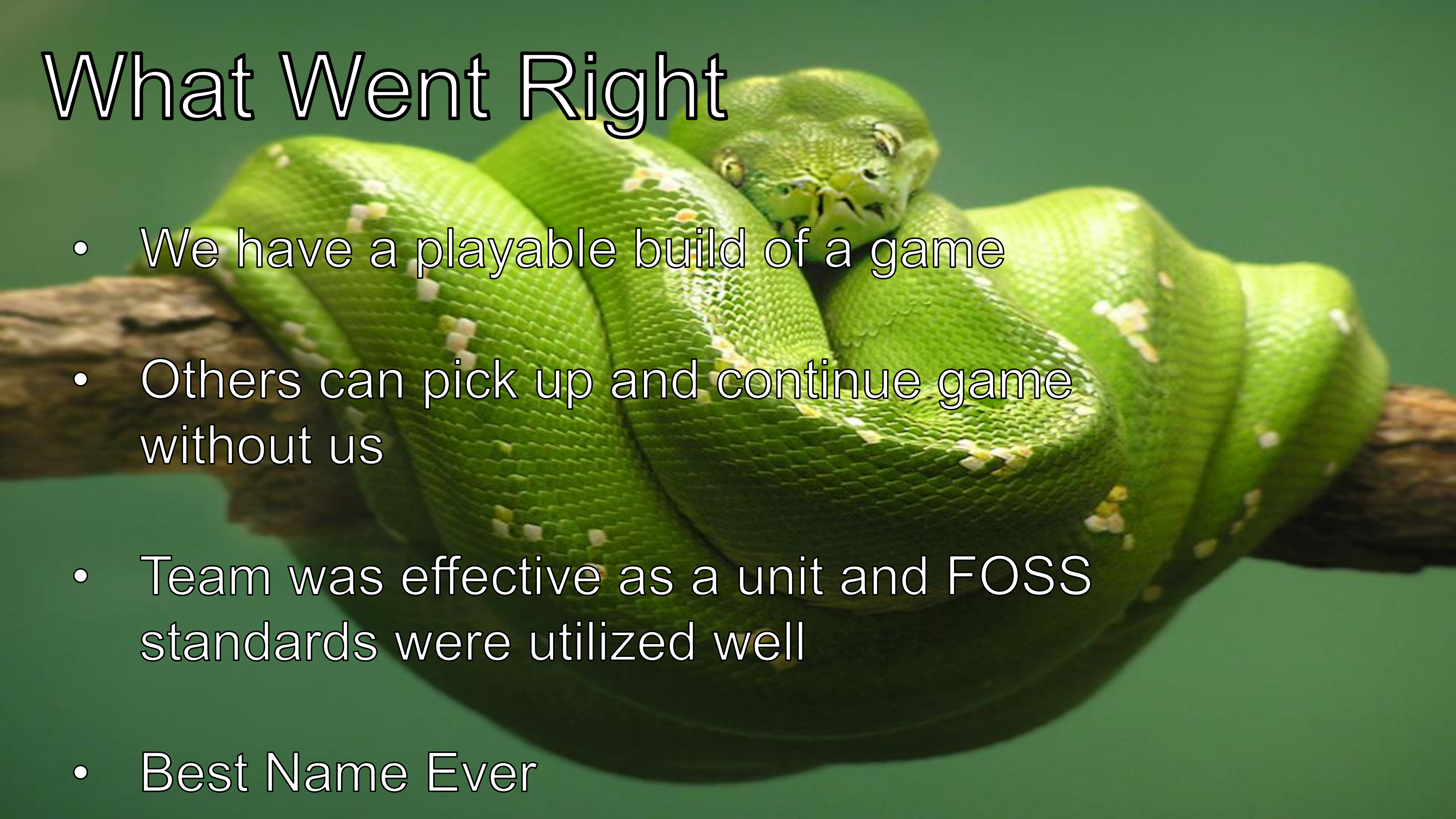
Index

1.0 - General Mechanics

Python Math - Adder's Garden Adventure is a tile-based Snake-like game. As such, the avatar travels from tile to tile with the tail following behind, taking up that tile. The avatar will pass through gates which will make the avatar longer or smaller as dependant on the gate. Once the avatar is of the appropriate size, they will be able to exit the puzzle.

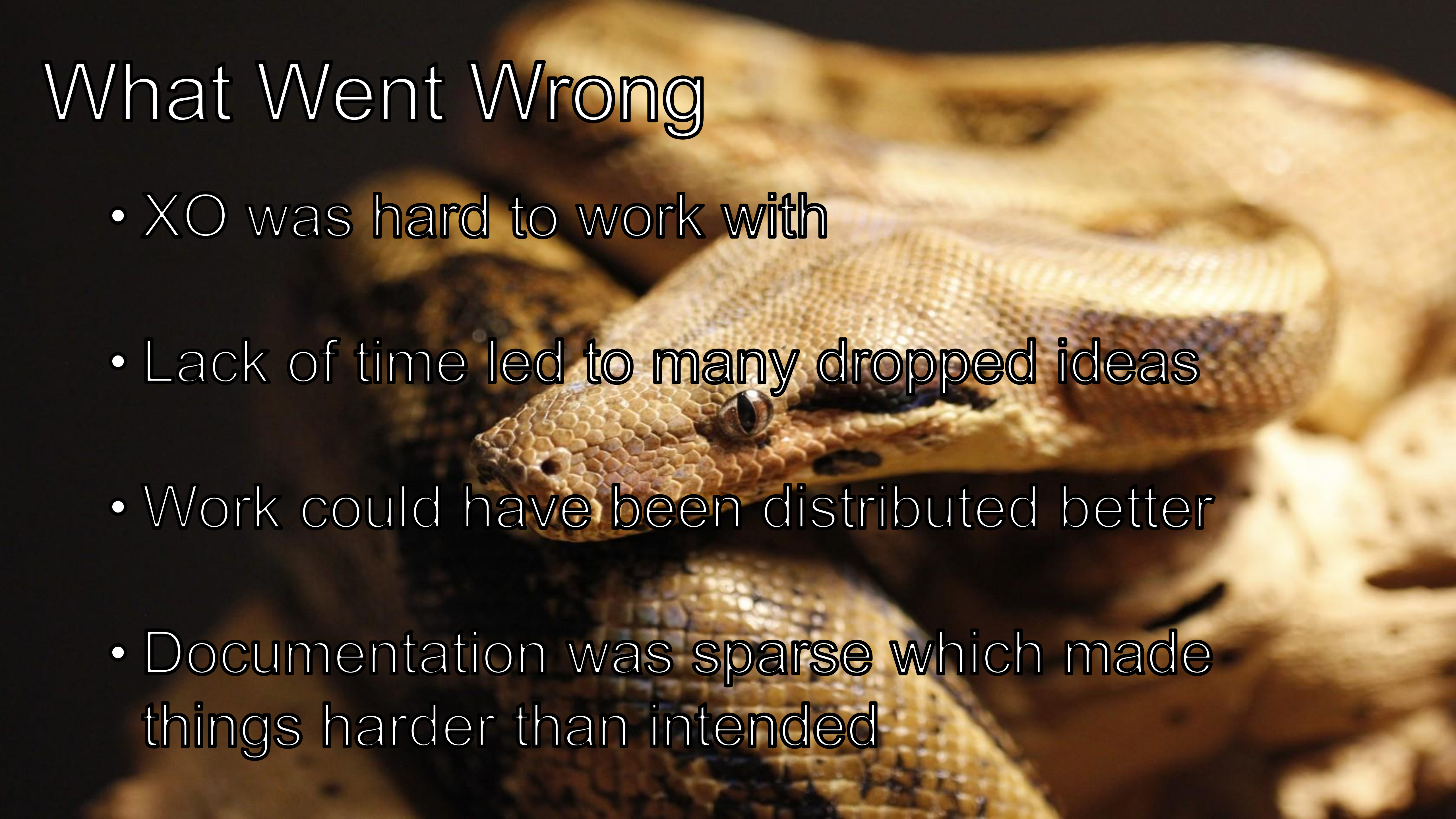
2.0 - Movement

The avatar that the player controls will move from one tile to another and can



What Went Right

- We have a playable build of a game
- Others can pick up and continue game without us
- Team was effective as a unit and FOSS standards were utilized well
- Best Name Ever



What Went Wrong

- XO was hard to work with
- Lack of time led to many dropped ideas
- Work could have been distributed better
- Documentation was sparse which made things harder than intended

What Didn't Make it In

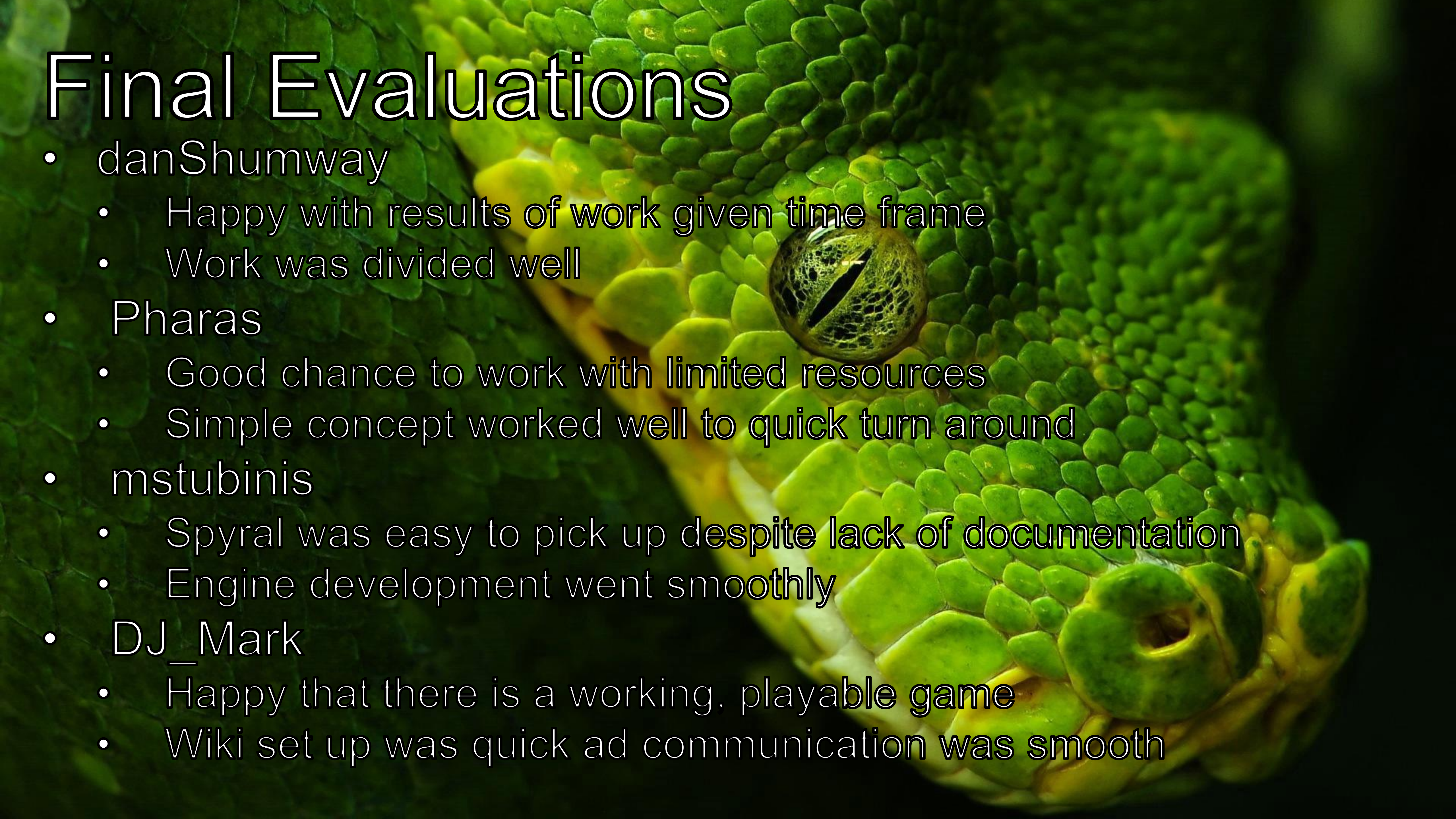
- External Art Assets

  🟩🟩🟩

- Reaching out for external design Elements

- No proper external tools

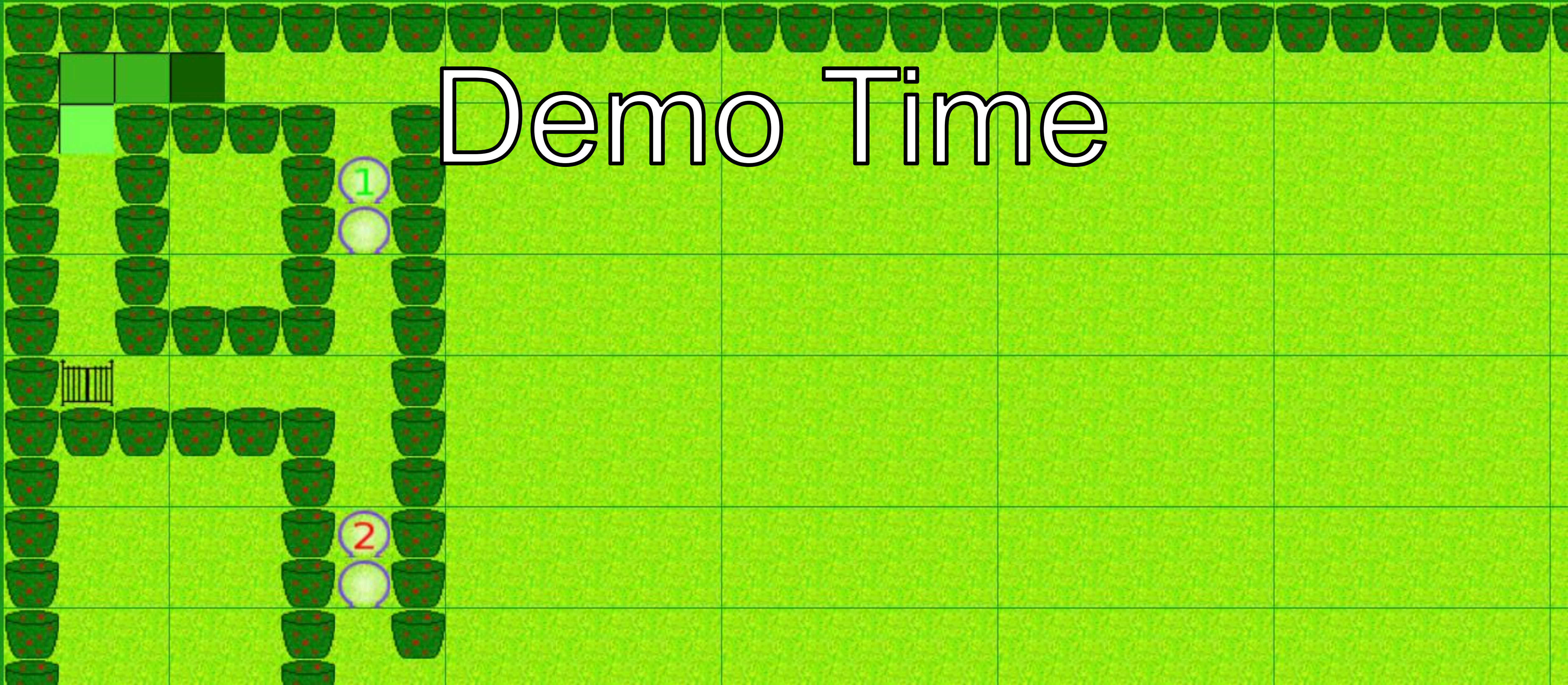
- Badges



Final Evaluations

- danShumway
  - Happy with results of work given time frame
  - Work was divided well
- Pharas
  - Good chance to work with limited resources
  - Simple concept worked well to quick turn around
- mstubinis
  - Spyral was easy to pick up despite lack of documentation
  - Engine development went smoothly
- DJ_Mark
  - Happy that there is a working. playable game
  - Wiki set up was quick ad communication was smooth

The snake needs to be 3 pieces long to go to the next level!
The snake is currently 2 pieces long!



Demo Time