**Alex Kidd2: Project Proposal**

**Description**

Alex Kidd is a comic book character that appeared in a lot of video games and arcades in the 1980s. The first was Alex Kidd in Miracle World and more kept coming. However, this era soon died, and development for versions compatible to new technology ceased. Thus, as the tech industry grew, Alex Kidd was no longer playable and other games took its place. None were as good, though, I would add.

Alex Kidd2 is a new, improved version of Alex Kidd. With multiplayer features and an addition of exciting board games in the larger game, Alex Kidd2 is going to bring back memories for the older generations and an entertaining web-based game for the younger generations.

**Modules**

PyGame

Sockets

Pyjsdl (If time, for web integration)

**Tasks**

Mini Games: Rock Paper Scissors, Checkers, Chinese Checkers

Player Abilities: Walking, shooting, jumping, flying for level 2

TASK ONE

Write Environment and Player Class. Add background surface and ground to the environment class. Allow scrolling using arrow keys. Allow player to move using arrow keys. Add lives for player. Add level and score function to player. (init, player.level, player.score)

GRAPHICS

Create Player images: standing, walking (right leg, left leg), punching

TASK TWO

Write Enemy Class. Enemy image depends on level. Enemies should already be placed on board when game begins. Allow enemy to walk back and forth on a small distance. If enemy touches player and player.attack is not called, player loses a life.

Write attack function in player class. Call player.attack if space key is pressed. This function should change player’s image to punching player, and destroy enemy that is touched (collisions) by player’s punch (shoots out a boomerang time thing). (Add +5 to player score)

GRAPHICS

Create Enemy images: standing, walking (right leg, left leg)

TASK THREE

Make sure all collisions work between enemy and player. Add jumping ability to player.

Create Display: Home screen (Just a start game option initially), Game Over Screen, Level one screen, Level two screen. Allow keypresses/mouse presses to start game/move on from the display screen, and home button to go back to home screen (or esc if game over screen). (If any button is pressed from game over screen, call init( ) again to reset game values.)

If player.lives = 0, display game over screen. If player reaches end of level, display level 2 screen for 5 seconds. If player clicks start game from home screen or restarts game from game over screen, show level 1 screen for 5 seconds before showing game play screen.

Display level, score, and lives (hearts) on game play screen.

GRAPHICS

Page Display backgrounds, icons

TASK FOUR

Place platforms in sky for player to jump on. Ground can be split in two, under ground and upper, and enemies can come from either one.

Place rocks and obstacles on way. Player has to jump over them. Cannot go through.

GRAPHICS

Create platform designs, rocks, obstacles, and such.

TASK FIVE

Write Coins class. Coins should randomly appear throughout the game. Player needs to click or walk on the coin to collect it. If player collects a coin, add +1 to score. If coin is collected, remove coin from screen. If coin is not collected for 20 seconds, remove coin from screen.

Place treasure chests (subclass of coins) on game screen. These must be punched (attacked) by player to open. If opened, they give 5-10 coins. Show animation and add coins to score.

GRAPHICS

Create coins, create treasure chests (closed, open, and coins falling out, flying up to score)

TASK SIX

Place castles on game screen. Create indoor castle screen, with board game grid.

If player presses enter, they enter indoor castle. Click exit button or press esc to exit indoor castle. Player uses arrow keys to choose board game move. Player cannot walk inside castle.

GRAPHICS

Create indoor castle background assets. (Rock paper scissor hands, etc.)

TASK SEVEN

Write algorithms for Tic Tac Toe and Rock Paper Scissor. If player wins the game, player wins coins or super powers, randomly chosen by the computer. Create a list of awards: coins, bombs power (set a variable to true and create a bomb class which allows player to place bombs instead of punching), a gun for shooting. Exit player out of the castle and into normal game screen at the end of game. Keep track of levels for castle games. Games get harder as levels go up.

GRAPHICS

Create bomb and gun and coin animations.

TASK EIGHT

Develop Home Screen. Add a Play Single Player button and Instructions button. Create Instructions page which shows image of a portion of the keyboard with player one and player two instructions. Add a Play Multiplayer button but it should do nothing right now. Play Single Player should pull up game play screen, and start level one.

Instructions:

Use left and right arrow keys to move player 1

Use up arrow to jump

Press spacebar to punch

Press enter to enter castles

GRAPHICS

Create keyboard and instructions graphics.

TASK NINE

Add sounds for coin collections. Add background music for game. Add sounds for when player dies and when enemy dies. Add win sound when level is passed and dynamic sound when game is over. Add animations: when player dies, make a flying ghost for player. If enemy dies, show a ‘poof’. Make sure player’s walking images show up when left/right keys are pressed.

GRAPHICS

Create Ghost.

TASK TEN

Create the multiplayer game play screen. Set some variable to true, and add another player. Player2 can move using different keys (A and D for left and right. W for jumping. F for shooting and G to enter castles.) Cannot kill player 1. Set timer: game ends when time ends and compare scores. Add points of time left to score for player when finished. Level 2 starts when both players have finished level 1. Player with highest score wins.

1. Develop game environment-
   1. Environment (class)
      1. Background surface
      2. Ground
   2. Player (class)
   3. Scrolling as player moves, back and forth
2. Bullets for player to shoot forward (or punching ability)-
3. Enemy (class)-
   1. Cars
   2. Porcupine
   3. Changes as levels go up
   4. May be able to shoot for higher levels
4. Score (class?)
5. Collisions-
   1. If enemy and player collide, take out one life
   2. If player’s bullet and enemy collide, enemy destroyed
      1. Add +5 to score
6. Player can jump, up and forward on ground-
7. Finish line-
   1. If player reaches end of level, show Level 2 screen and start level two upon keypress or mouse press
      1. With refilled lives
   2. If player runs out out of lives, game over screen and option to start over
8. Coins (class)-
   1. Place coins on screen for player to collect. Add to score.
   2. Disappear in 20 seconds if not collected
   3. Click or walk on it to collect
   4. Treasure Chests (class): punch to get more coins
9. Castles (class) -
   1. Place castles on route for player to enter. Play board games.
      1. If won: randomly get coins, or super powers
         1. Jumping shoes for higher/longer jumps
         2. Gun for shooting. . .(if punches are default)
         3. Bombs to place, burst on timer.
      2. Rock paper scissors and tic tac toe for level 1
         1. Games get harder as levels go up
10. Place platforms in sky for player to jump on. Coins can be up there too. Ground can be split in two, under ground and upper, and enemies can come from either one. -
11. Sounds-
    1. Coin collecting sound
    2. Background music
    3. Sound when enemy dies
    4. Sound when player dies
12. Animations-
    1. Ghost flying up when player or enemy dies
13. Add home screen:
    1. Play Single Player button-
       1. Does all the above.
    2. Play Multiplayer button
       1. Set some variable to true, and add another player.
       2. Player2 can move using different keys
       3. Cannot kill player 1
       4. Set timer: game ends when time ends.
          1. Add points of time left to score for player when finished.
          2. Level2 starts when both players have finished level1
       5. Player with highest score wins
    3. How to Play: instructions-
       1. Image of keyboard
       2. Player1 and Player2 instructions