|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tarea** | **Fecha de inicio** | **Fecha estimada** | **Fecha de fin** | **Retrasos** | **Resumen** | **Dificultades** | **Realizada por** |
| Buscar información sobre el juego. | 15/02 | 16/02 | 16/02 |  | Se han encontrado varios ejemplos del juego 2048 realizados en C (sobre CPU) y en JavaScipt. La IA que se utiliza por lo general consiste en un árbol de decisión basado en MIN MAX. | Se ha encontrado poca información sobre las librerías gráficas. | Juan  Casandra |
| Elección del formato de representación de datos dentro de las matrices. | 15/02 | 16/02 | 16/02 | (modificación el 22/02) | Se ha elegido representación float, por rapidez en el cómputo |  | Juan  Casandra |
| Algoritmo de desplazamiento de los números en la matriz. | 16/02 | 20/02 | 16/02 |  | Se realiza el desplazamiento correctamente tal como en 2048.  El algoritmo se hace sobre un vector y se aplicará a la matriz fila a fila o columna a columna en paralelo según el moviento que se realice. | Hacer el algoritmo de forma que sea fácil extraer metadatos de los movimientos que se han realizado en caso de que fuera necesario. | Juan |
| Gestión de los datos introducidos por el usuario. | 16/02 | 20/02 | 17/02 |  | Se le piden datos al usuario si no los ha introducido al iniciar el programa. |  | Casandra |
| Generación de la semilla inicial de forma aleatoria | 16/02 | 20/02 | 17/02 |  | Se genera una semilla desde la GPU de forma aleatoria. |  |  |
| Transformación de las matrices internas a la representación en número a postrar por pantalla. | 22/02 | 03/03 | 01/03 |  | En un principio la representación se iba a hacer en chars empezando por la ‘A’ y luego realizar una conversión a número potencia de dos. Finalmente se ha decidido utilizarlos números directamente por lo que no ha sido necesario hacer esta parte. |  | Juan |
| Calcular la puntuación que hay en un tablero. | 22/02 | 03/03 |  |  | En progreso. Hemos decidido hacerlo por reducción sumando pares contiguos. |  | Juan |
| Averiguar cuando en un tablero ya no se pueden realizar más movimientos. | 22/02 | 03/03 | 28/02 |  | Recorrer por filas el tablero, comprobando a izq, dcha, arriba y abajo, si están vacíos, son iguales o distintos |  | Casandra |
| Gestión de las vidas. | 22/02 | 03/03 |  | Todavía no comenzado pero ya que parece sencillo puede que no acabe en retraso. |  |  | Casandra |
| Calcular el tamaño de bloques e hilos a utilizar en función de las características hw del equipo. | 03/03 | 06/03 |  |  | Creemos que se hará mirando maxThreads |  | Juan |
| Añadir una “pieza” nueva a cada iteración del juego en una posición vacía | 06/03 | 10/03 |  |  |  |  |  |
| Implementar los kernell en memoria global. Máximo 32x32 | 07/03 | 10/03 |  | No se ha implementado. |  |  | Juan |
| Implementar los kernell en memoria global con bloques. | 07/03 | 10/03 |  |  |  |  | Casandra |
| Implementar los kernell en memoria compartida. | 07/03 | 10/03 |  |  |  |  | Juan |
| Interfaz ASCII del juego. | 03/03 | 06/03 |  |  |  |  | Casandra  Juan |
| Guardar el estado de una partida. | 03/03 | 06/03 |  |  |  |  | Casandra |
| IA para jugar de forma automática BASADA EN ALEATORIEDAD. | 07/03 | 10/03 |  |  |  |  | Casandra |
| Interfaz OpenGL del juego. | 11/03 | 14/03 | NO |  |  |  | Juan |
| Elección de Heurísticas para evaluar es estado del juego. | 11/03 | 14/03 | 10/03 |  |  |  | Casandra  Juan |
| Evaluador de matriz, dada unas heurísticas califica una matriz. | 11/03 | 14/03 | 10/03 |  |  |  | Casandra |
| IA para jugar de forma automática BASADA EN HEURISTICA SIMPLE. | 11/03 | 14/03 | 10/03 |  |  |  | Juan |
| IA para jugar de forma automática BASADA EN HEURISTICA SOBRE MIN MAX. | 11/03 | 14/03 | NO |  |  |  | Casandra  Juan |