The basic classes (Grid, Cell, FallingBlock, FallenBlock) will be done before Sunday. By Monday game level 1 without special command should run. By Tuesday special command should run. By Friday most feature should run. I'll be working alone.

Question: How could you design your system (or modify your existing design) to allow for some generated blocks to disappear from the screen if not cleared before 10 more blocks have fallen?
Could the generation of such blocks be easily connected to more advanced levels?

Each cell has a pointer to the block that gave birth to it. The block has information of time created. After each turn the grid checks and deletes any cell whose birth time is < now – 10, and delete the cell (deleting all member of a block will automatedly delete the block as well)

How could you design your program to accommodate the possibility of introducing

additional levels into the system, with minimum recompilation?

The Player can have one more child named “PlayerLevelN”. Also only the controller.h and no other .h file would need to include PlayerLevelN.h.

Question: How could you design your program to allow for multiple special effects to applied simultaneously? What if we invented more kinds of effects? Can you prevent your program from having one else-branch for every possible combination?

Firstly, changes to special effect should only affect the Player class because the effects are ‘special’ and basic classes shouldn’t check for them. Visitor Pattern could be used. Command can be made into virtual class. Players are already virtual. So different behaviors would happen when different level players are applying different special effects.

How could you design your system to accommodate the addition of new command
names, or changes to existing command names, with minimal changes to source and minimal
recompilation? (We acknowledge, of course, that adding a new command probably means adding
a new feature, which can mean adding a non-trivial amount of code.) How difficult would it be to
adapt your system to support a command whereby a user could rename existing commands (e.g.
something like rename counterclockwise cc)?

using a command interpreter class ‘controller.cc’, this class would be solely responsible for translating command and calling method for Player class(and its children perhaps) because it’s the actual player who is giving the commands.