*\*\* houd in dat dit een nieuwere aantekening is, gemaakt bij de meest recente les.*

* *🡨 Houd in dat een punt gelukt, gedaan of behaald is*
* *Zorg dat je de versie beheer okee hebt voor de les van de volgende dag!*

Project demo notes:

* Presenteer naar de klant, vraag om mening/feedback, speel in op de klant

Vragen voor de C# check up

* Alléén de coördinaten gebruiken voor initialisatie?! (in array zetten voor objecten herkenning heb je een locatie nodig) (buren – attribuut in de tekst?)
* Btn’s aparte class voor aanmaken of in de main form aanhouden
* Realtime > timer?

Vragen voor de UML check up

* Unaire verbinding, box die een box kan duwen?

Notitie UML

* hoofdklasse staat boven aan, form is daar de papa van

form

main

* bezit is een zwart ruitje (interface heeft een speelveld = <>, zonder interface kan speelveld niet bestaan)
* Collectie van vakjes voor het burencollectief vanwege een unaire verbinding naar zich zelf
* Vang de volger klasse hernoemen naar iets beters: gamewindow
* Zorg dat je ook de klasse references hebt die terug te vinden zijn n het diagram
* om bidirectionele pijltjes aan te geven moet je twee pijltjes over elkaar heen plaatsen
* \*\*\*Dump de path classe en de interface Iview
* \*\*\* voor clicks(buttons) als het maar 1 regel is mag je het weglaten op je uml anders dump je het

Notitie C#

* Box bewegen 🡪 if player\_block is adjacent to [box] && player move [tegengestelde richting van box(rechts : links||up : down)]
* Walk method: generale methode van beweging, enemy & player hebben het, player geeft toets-input door in de parameters, enemy random.
* Enemy zet elke seconde(of sneller) een stap
* Pauze knop, blokkeert alle invoer van beide kanten en zet de game dus “stil”
* Speler kan ook hele rij dozen duwen: if (box2 is adjacent to [box1] && [box2] is adjacent to player block && player move > box) {

Move box 1 [tegengestelde richting]}

* 2d array moet een vakje zijn waarop een spelobject staat(met buren)
* Maak een vakje class (locatie maak een picture box hier aan)hier steeds nieuw plaatje
* vakje is een speciaal soort picture box(overerving van speciale objecten)
* Bij lvl opbouw vervang int[10,10] naar Tile[10,10], maak ook class Tile aan, en zorg dat in de class Tile de objecten aan de juiste random nummer worden gekoppeld en geplaatst(OBJECT GEORIENTEERT!)
* Static is om attributen door te kunnen geven uit abstracte klassen
* zorg dat je alles begrijpt wat op je scherm staat
* ipv voor lopen coördinaten geeft van waar hij staat, is er een variabele die door geeft op welk vakje hij staat
* look up a star/Dijkstra algorithm(pathfinding)
* ~ houdt tegen over gesteld in
* Public Dictionary<direction>, tile) = new public dictionary….. 🡪 voor opslaan buurman
* Tests toevoegen voor de moeilijke methodes
* Gameobject bepaald of hij een tile verplaatst niet vanuit tile
* \*\*\*Neighbor direction != null && object neighbor == null 🡪 move
* \*\*\*when writing a check line in an if statement use ? to find out if it is existing
* \*\*\*help box aanmaken using messageBox.show (using windows forms)
* \*\*\*percentage using a constant variable rnd (1,101) percentage <wallPercentage

speelveld

overerving

Picture box

Vakje

\_\_\_\_\_\_tekene

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Extra:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Power-up extra/ debuff blokje wat als je aan raakt 2 blokjes opzij gaat ipv 1
* Enemy spawnt X sec na initialisatie?
* Soundbytes voor enemy spawn, item pickup, BGM, game over?
* Settings: kies je vak grootte : 11- makkelijk , 25 – gemiddeld, 50 – moeilijk?
* enemy kan 8 richtingen(diagonal)?